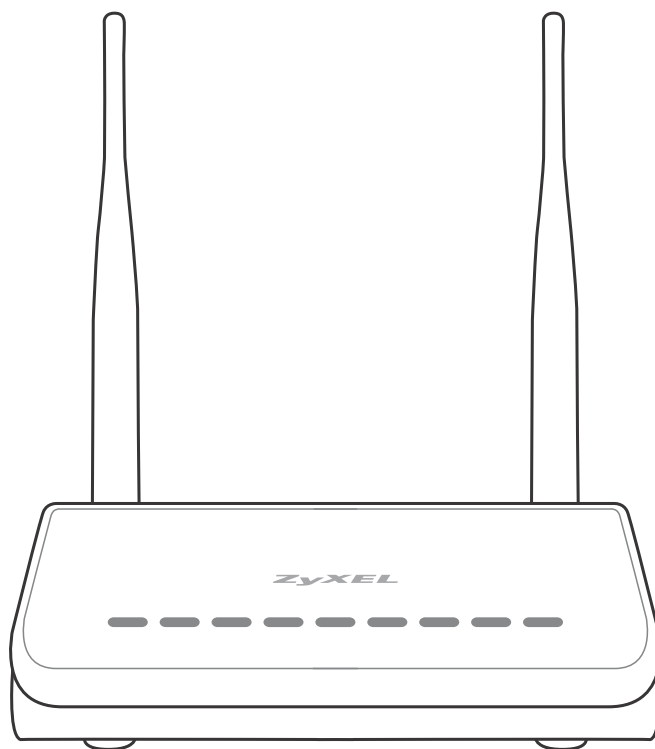


Keenetic

*Wireless 802.11n Broadband Router
(300 Mbps) with multi-purpose USB port*

User's guide



Firmware Version 2.00(3)B2
Edition 0.6 11.04.2012

www.zyxel.com

ZyXEL

Contents Overview

Command line operation 11

General purpose commands 17

Basic IP configuration 21

Firewall and NAT address translation 35

Switch and VLAN Interfaces 43

Bridges 49

PPP 53

Wireless network 802.11 63

Configuring DNS 69

DHCP 71

IGMP 77

Device access control 79

Diagnostics 83

Examples of settings 87

Glossary 97

Table of Contents

Table of Contents	3
Chapter 1	
Command line operation	11
1.1 Login	11
1.2 Entering commands	11
1.3 Entering a group	12
1.4 Auto-completion and help	12
1.5 Prefix no	13
1.6 Commands and settings	14
1.7 Commands with multiple input	14
1.8 Saving settings	15
1.9 Delayed restart	15
Chapter 2	
General purpose commands	17
2.1 Command reference	17
2.1.1 system	17
2.1.2 system reboot	17
2.1.3 system set	18
2.1.4 system hostname	18
2.1.5 copy	19
2.1.6 more	19
2.1.7 show running-config	20
Chapter 3	
Basic IP configuration	21
3.1 Network Interfaces	21
3.2 Classes of the interfaces	21
3.3 Static and dynamic network interfaces	21
3.4 Basic functions of the interface	22
3.5 MAC functions	22
3.6 IP functions	22
3.7 Ethernet Functions	23
3.8 Command description	23
3.8.1 interface	23
3.8.2 interface name	23
3.8.3 interface description	24
3.8.4 interface up	24

3.8.5 interface down	24
3.8.6 interface mac address	25
3.8.7 interface ip address	25
3.8.8 interface ip alias	26
3.8.9 interface ip dhcp	26
3.8.10 interface ip mtu	27
3.8.11 interface ip tcp adjust-mss	27
3.8.12 ip route	28
3.8.13 interface ip global	29
3.8.14 IPv6 settings	29
Chapter 4	
Firewall and NAT address translation	35
4.1 Command description	35
4.1.1 ip nat	35
4.1.2 ip static	35
4.1.3 interface security-level	37
4.1.4 isolate-private	37
4.1.5 access-list	37
4.1.6 interface ip access-group	39
4.2 Examples	40
Chapter 5	
Switch and VLAN Interfaces	43
5.1 Switch Interface	43
5.2 VLAN interface	43
5.3 Command description	44
5.3.1 interface port	44
5.3.2 interface port speed	44
5.3.3 interface port duplex	45
5.3.4 interface port mode	46
5.3.5 interface port access	46
5.3.6 interface port trunk	47
Chapter 6	
Bridges	49
6.1 Bridge interface	49
6.2 Command description	50
6.2.1 interface include	50
6.2.2 interface inherit	51
Chapter 7	
PPP	53
7.1 PPP functions	53
7.2 Functions Secure	53
7.3 PPPoE interface	53

7.4 PPTP Interface	53
7.5 L2TP Interface	53
7.6 Configuration sequence	53
7.7 Additional options	54
7.7.1 LCP Echo	54
7.7.2 CCP	55
7.7.3 IPCP	55
7.8 Command description	55
7.8.1 interface peer	55
7.8.2 interface connect	56
7.8.3 interface authentication pap	56
7.8.4 interface authentication chap	56
7.8.5 interface authentication mschap	57
7.8.6 interface authentication mschap-v2	57
7.8.7 interface authentication identity	57
7.8.8 interface authentication password	58
7.8.9 interface encryption mppe	58
7.8.10 interface lcp echo	59
7.8.11 interface ccp	59
7.8.12 interface ipcp default-route	59
7.8.13 interface ipcp name-servers	60
7.8.14 interface debug	60
7.8.15 interface ip mru	60
Chapter 8	
Wireless network 802.11	63
8.1 Access points	63
8.2 Wireless stations	63
8.3 Command description	63
8.3.1 interface ssid	63
8.3.2 interface channel	64
8.3.3 interface compatibility	64
8.3.4 interface power	65
8.3.5 interface authentication wpa-psk	65
8.3.6 interface authentication shared	66
8.3.7 interface encryption enable	66
8.3.8 interface encryption key	66
8.3.9 interface encryption wpa	67
8.3.10 interface encryption wpa2	67
Chapter 9	
Configuring DNS	69
9.1 Command description	69
9.1.1 ip name-server	69

9.1.2 service dns-proxy	70
Chapter 10	
DHCP	71
10.1 DHCP server	71
10.2 DHCP relay	71
10.3 Commands description	71
10.3.1 ip dhcp pool	71
10.3.2 ip dhcp pool range	72
10.3.3 ip dhcp pool default-router	72
10.3.4 ip dhcp pool dns-server	73
10.3.5 ip dhcp pool lease	73
10.3.6 ip dhcp host	74
10.3.7 service dhcp	74
10.3.8 ip dhcp relay lan	75
10.3.9 ip dhcp relay wan	75
10.3.10 ip dhcp relay server	76
10.3.11 service dhcp-relay	76
Chapter 11	
IGMP	77
11.1 Command description	77
11.1.1 interface igmp upstream	77
11.1.2 interface igmp downstream	77
11.1.3 interface igmp fork	78
11.1.4 service igmp-proxy	78
Chapter 12	
Device access control	79
12.1 Command description	79
12.1.1 user	79
12.1.2 user password	79
12.1.3 user tag	80
12.1.4 service http	81
12.1.5 service ftp	81
12.1.6 service telnet	81
Chapter 13	
Diagnostics	83
13.1 Command description	83
13.1.1 show system	83
13.1.2 show interface	83
13.1.3 show interface mac	84
13.1.4 show ip route	85
13.1.5 show ip arp	85
13.1.6 show ip name-server	85

13.1.7 **show log** 85

13.1.8 **tools ping** 86

Chapter 14

Examples of settings 87

14.1 Testing of throughput 87

14.2 Routing with NAT enabled 88

14.3 DHCP-server and DHCP-client 89

14.4 Wi-Fi access point in bridge mode 90

14.5 PPP connection 92

Glossary 97

Command line operation

The primary tool for managing the Keenetic router is *the command line interface* (CLI). System settings can be defined as a sequence of commands, which can be executed to bring the device to the specified condition.

Keenetic has three types of settings:

Current settings	<i>running config</i> is a set of commands describing the current status of the system. Current settings are stored in RAM and reflect every change of the system settings. However, the content of RAM is lost when the device is turned off. To restore the settings after reboot, they must be saved in non-volatile memory.
Startup configuration	<i>startup config</i> is a sequence of commands, which is stored in a specific partition of the non-volatile memory. It is used to initialize the system immediately after startup.
Default settings	<i>default config</i> contains factory default settings of Keenetic. RESET button is used to reset startup configuration to the factory default.

1.1 Login

When you connect to Keenetic via telnet, it prompts for the administrator password first. The password is set to 1234 by default.

```
$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to 192.168.1.1
Escape character is '^]'.

Password: ****
(config)>
```

1.2 Entering commands

Command line interpreter of Keenetic is designed for beginners as well as experts. All command names and options are clear and easy to remember.

Commands are divided into groups and arranged in a hierarchy. Thus, to do a setting, the operator needs to enter a sequence of nested command group names (*node* commands), and then enter the final command with parameters.

For example, IP-address of the Switch0/VLAN2 network interface is set using the *address* command, which is located in the **interface** → **ip** group:

```
(config)> interface Switch0/VLAN2 ip address 192.168.15.43/24
Network address saved.
```

1.3 Entering a group

Some of the node commands (containing a group of child commands) can be “entered” to allow direct executing of the child commands without typing the node name as prefix. In this case the prompt is changed to indicate the entered group.

The **exit** command or [Ctrl]+[D] key combination can be used to exit a group.

For example, after entering the interface group the command line prompt is changed to (config-if):

```
(config)> interface Switch0/VLAN2
(config-if)> ip address 192.168.15.43/24
Network address saved.
(config-if)> [Ctrl]+[D]
(config)>
```

1.4 Auto-completion and help

To make the configuring process as comfortable as possible, the command line interface provides auto-completion of commands and parameters, hinting the operator, which commands are available at the current level of nesting. Auto-completion works by pressing [Tab]. Example:

```
(config)> in[Tab]

interface - network interface configuration

(config)> interface Sw[Tab]

Usage template:
interface {name}

Variants:
Switch0
Switch0/VLAN1
Switch0/VLAN2

(config)> interface Switch0[Tab]

Usage template:
interface {name}

Variants:
Switch0/VLAN1
Switch0/VLAN2

(config)> interface Switch0/VLAN2[Enter]
(config-if)> ip[Tab]
```

```

        address - set interface IP address
        alias - add interface IP alias
        dhcp - enable dhcp client
        mtu - set Maximum Transmit Unit size
        mru - set Maximum Receive Unit size
    access-group - bind access-control rules
        apn - set 3G access point name

(config-if)> ip ad[Tab]

        address - set interface IP address

(config-if)> ip address[Tab]

Usage template:
address {address} {mask}

(config-if)> ip address 192.168.15.43[Enter]
Configurator error[852002]: address: argument parse error.
(config-if)> ip address 192.168.15.43/24[Enter]
Network address saved.
(config-if)>

```

Hint for the current command can always be displayed by pressing [?]. Example:

```

(config)> interface Switch0/VLAN2 [?]

        description - set interface description
        alias - add interface name alias
        mac-address - set interface MAC address
        dyndns - DynDns updates
        security-level - assign security level
        authentication - configure authentication
            ip - set interface IP parameters
            igmp - set interface IGMP parameters
            up - enable interface
            down - disable interface

(config)> interface Switch0/VLAN2

```

1.5 Prefix no

Prefix **no** is used to negate a command.

For example, the command **interface** is responsible for creating a network interface with the given name. When used with this command, prefix **no** causes the opposite action — removing of the interface:

```

(config)> no interface PPPoE0

```

If the command is composite, **no** can be placed in front of any member. For example, **service dhcp** turns on the DHCP service. It consists of two parts: **service** — the group name in the

hierarchy of commands, and **dhcp** — the final command. Prefix **no** can be placed either at the beginning, or in the middle. The action is the same in both cases: stopping of the service.

```
(config)> no service dhcp
(config)> service no dhcp
```

1.6 Commands and settings

Most of the commands change the system state and store the changes in the current settings, which can be displayed using **show running-config**. For example, if you type **service http**, the HTTP management service is started and the “service http” record appears in the settings:

```
(config)> service http
HTTP server enabled.
(config)> show running-config
...
service http
```

When you enter a command with the prefix **no**, the “service http” string is removed from running-config. The HTTP service is turned off by default. This behavior allows to restore the service status at system startup, if such settings are saved in the startup-config.

Almost all commands are added to the settings using the principle “have a command — have a function”. So, when the operator is looking at the running-config, he can see what is turned on in the system.

1.7 Commands with multiple input

Many commands have the property of *idempotence*, which means that multiple input of a command has the same effect as the single input. For example, entering **service http** adds a single line “service http” to the current settings, and re-entering does not change anything.

However, some of the commands allow you to add not a single, but multiple records, if they are entered with different arguments. For example, static routing table entries **ip route** or filters **access-list** are added sequentially and appear in the settings as a list:

Example 1.1. Using a command with multiple input

```
(config)> ip route 1.1.1.0/24 PPTP0
Route added.
(config)> ip route 1.1.2.0/24 PPTP0
Route added.
(config)> ip route 1.1.3.0/24 PPTP1
Route added.
(config)> show running-config
...
ip route 1.1.1.0 255.255.255.0 PPTP0
ip route 1.1.2.0 255.255.255.0 PPTP0
ip route 1.1.3.0 255.255.255.0 PPTP0
...
```

Records from such tables can be removed one by one, using prefix **no** and arguments to identify the record you want to remove:

```
(config)> no ip route 1.1.2.0/24
Route deleted.
(config)> show running-config
...
ip route 1.1.1.0 255.255.255.0 PTP0
ip route 1.1.3.0 255.255.255.0 PTP0
...
```

1.8 Saving settings

Current and startup settings are stored in the files `running-config` and `startup-config`, respectively. To save the current settings in the non-volatile memory, copy them as shown below:

```
(config)> copy running-config startup-config
Copied: running-config -> startup-config
```

1.9 Delayed restart

If Keenetic is located away from the operator and is managed remotely, there is a risk to lose control over it because of a misoperation. In this case it will be difficult to reboot and return to the saved settings.

The **system reboot** command lets you set a delayed restart timer, perform “risky” settings, then turn off the timer and save the changes. If connection to the device is lost during configuration, the operator will be enough to wait for automatic reboot and connect to the device again.

General purpose commands

2.1 Command reference

2.1.1 system

Entering the group of system functions, including commands to configure global parameters.

Properties	
Prefix no	no
Change settings	no
Multiple input	no
Enter group	(system)

2.1.2 system reboot

Reboot the system. If the parameter is set, reboot is executed after a timeout, in seconds. Prefix **no** cancels a scheduled reboot. If the timer is already set, using of the command replaces the old value of the timer to the new one.

Using a scheduled reboot is convenient in the case when the device is under remote control, and the user doesn't understand the effect of the commands he/she is trying. The user can turn on a scheduled reboot for fear of losing control over the device. After reboot the system will return to its original state and become available.

Properties	
Prefix no	yes
Change settings	no
Multiple input	no

```
(system)>  reboot [interval]
```

```
(system)> no reboot
```

Argument	Type	Description
<i>interval</i>	Integer	Timeout for scheduled reboot. If not specified, the reboot will be executed immediately.

2.1.3 system set

Sets the value of the specified system parameter and saves it in the current settings. When used with the prefix **no**, it returns the value, which was set by default (before the first change).

Example:

```
(config)> system
(system)> set net.ipv4.ip_forward 1
(system)> set net.ipv4.tcp_fin_timeout 30
(system)> set net.ipv4.tcp_keepalive_time 120
(system)> set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
(system)> set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
(system)> set net.ipv4.netfilter.ip_conntrack_max 4096
(system)> exit
(config)> show running-config
system
set net.ipv4.ip_forward 1
  set net.ipv4.tcp_fin_timeout 30
  set net.ipv4.tcp_keepalive_time 120
  set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
  set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
  set net.ipv4.netfilter.ip_conntrack_max 4096
!
...
(config)>
```

Properties	
Prefix no	yes
Change settings	yes
Multiple input	yes

```
(system)> set <name> <value>
```

```
(system)> no set <name>
```

Argument	Type	Description
<i>name</i>	String	Identifier of the system parameter
<i>value</i>	String	Value of the system parameter

2.1.4 system hostname

Set the host name. Prefix **no** sets the default value, which depends on the model name.

Host name used to identify a node in the network. It is required to enable some of the built-in services, such as [CIFS](#) server.

Properties	
Prefix no	yes

Properties	
Change settings	yes
Multiple input	no

```
(system)> hostname <hostname>
```

```
(system)> no hostname
```

Argument	Type	Description
<i>hostname</i>	String	Name of the host

2.1.5 copy

Copies the contents of one file to another. Used to update the firmware, save the current settings, reset to factory defaults etc.

For example, the current settings can be saved as follows:

```
(config)> copy running-config startup-config
```

File names in this example are aliases. Full names of the configuration files are `system:running-config` and `flash:startup-config`, respectively.

Properties	
Prefix no	no
Change settings	no
Multiple input	no

```
(config)> copy <source> <destination>
```

Argument	Type	Description
<i>source</i>	Filename	From where to copy, full name of a file or an alias.
<i>destination</i>	Filename	Copy destination, full name of a file or an alias.

2.1.6 more

Displays the contents of a text file line by line.

Properties	
Prefix no	no
Change settings	no

```
(config)> more <filename>
```

Parameter	Type	Description
<i>filename</i>	File name	Full filename or alias.

2.1.7 show running-config

Displays current settings, that is file system:running-config{sp} content, just as command **more** does.

Properties	
Prefix no	no
Changes the settings	no

Basic IP configuration

3.1 Network Interfaces

Network interface Keenetic is a physical or a virtual network interface — an abstraction that describes a connection of a device to LAN. Physical interface corresponds to a network adapter with physical (wired or wireless) connection. Virtual interface corresponds to a virtual network, such as VLAN IEEE 802.1Q or Multiple ESSID in the wireless network IEEE 802.11.

In general, a device has several network interfaces, allowing one to transmit packets from one network to another in the router or bridge mode.

Each interface has a defined type, for example, FastEthernet or PPTP. Interfaces of each type are enumerated. Moreover, the interfaces are *nested* such that an inner interface is only meaningful in the context of another, *parent interface*. Thus, in Keenetic each network interface has a unique name consisting of parent interface name, type and index, for example:

```
FastEthernet0/VLAN100
```

Here FastEthernet0 — parent interface, VLAN — type, and 100 — unique index within the parent interface. In turn, the parent interface FastEthernet0 has type FastEthernet and index 0.

3.2 Classes of the interfaces

Each network interface has some set of functions and allows the operator to change the settings according to the interface objective. Settings are divided into groups. One can isolate the basic settings, IP settings, PPP settings, etc.

If the interface supports settings a certain group, they say that the interface belongs to *the class of the interfaces* with such settings. For example, interfaces such as FastEthernet, PPTP, Bridge and VLAN allow one to configure IP-address in spite of the significant differences, so they belong to the one base class — IP.

An interface can refer to several classes at the same time. In this case, they say that the interface *aggregates* functions of several classes.

3.3 Static and dynamic network interfaces

Static interfaces correspond to the hardware configuration of device. They are determined at system startup and exist till shutdown. The interfaces of this type can not be created or removed while the system is running.

Dynamic interfaces usually correspond to the virtual network connections. They can be created or removed while the system is running. Dynamic configuration of interfaces is displayed in the

current settings and can be stored in nonvolatile memory. Interfaces of the following classes can be called dynamic: VLAN, PPP, USB, Bridge and others.

3.4 Basic functions of the interface

Basic functions are applicable to all interfaces. The following functions are treated as basic:

- turning interface on/off using commands **interface up** and **interface down**;
- renaming interface using command **interface name**;
- assigning arbitrary description string using command **interface description**.

3.5 MAC functions

MAC class interfaces operate in networks of IEEE EUI-48 addressing schema. MAC functions are supported by the Ethernet class interfaces, which include VLAN, SSID, Bridge and WiMAX. MAC class interfaces have a default MAC address and allow one to set it manually using command **interface mac address**.

3.6 IP functions

IP address and connection to IP subnet can be configured via IP interface. Such network is called *directly (immediately)* connected. Directly connected networks are automatically added to the routing table.

Routing table IP knows about available IP-subnets, and consists of routes. Each route has a destination network specified by IP-address and mask. In the route specified identifier of an interface through which you can get to the destination network. If the interface is connected to a network with multiple entries, then transit IP address of the neighboring transit node or a **router** is to be specified on top of interface id. Packets transmission to the destination address is carried out using information contained in the IP routing table.

The routing table can have only one *default route* with the destination address 0.0.0.0/0. Such address is used to transmit packets for which more suitable specialized route could not be found.

There are three ways to add entries to the routing table:

- *Automatically*, when configuring interface IP-address using command **interface ip address** or by the service responsible for the connection via DHCP or PPP protocol. Automatic routes point to the networks that are connected directly.
- *Statically*, with the command **ip route**. Static routes are displayed in the current settings, and exist in the system through the lifetime of the network interface with which they work.
- *Dynamically* by the service responsible for the connection via DHCP or PPP protocol. Dynamic routes are typically transmitted by provider along with the IP address settings.

3.7 Ethernet Functions

Ethernet class interfaces aggregate IP and MAC base classes. Ethernet interfaces have a series of special properties, which will be discussed in following sections, namely:

- starting DHCP-client;
- creating VLAN hild interfaces;
- tapping into virtual bridges as ports;
- PPPoE protocol connections support;
- support of authentication 802.1x.

3.8 Command description

3.8.1 interface

Access to a group of commands to configure the selected interface. If the interface is not found, the command tries to create it. Prefix **no** deletes the interface.

Each of the commands of this group is applicable to the particular type of hierarchy interface. The base type of interface to which the command applies is specified in the command property “Interface Type”.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Access to Group	(config-if)

```
(config)> interface <name>
```

```
(config)> no interface <name>
```

Argument	Type	Description
<i>name</i>	String	Full interface name or an alias.

3.8.2 interface name

Assigns arbitrary name to the specified network interface. The interface can be referred to by the new name just like by ID. Prefix **no** deletes the configuration.

Properties	
Prefix no	yes
Changes the settings	yes

Properties	
Multiple entry	no

```
(config-if)> name <new>
```

```
(config-if)> no name
```

Argument	Type	Description
<i>new</i>	String	New interface name.

3.8.3 interface description

Assigns arbitrary description to the specified network interface. Prefix **no** deletes the description.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-if)> description <description>
```

```
(config-if)> no description
```

Argument	Type	Description
<i>description</i>	String	Arbitrary description of the interface.

3.8.4 interface up

Turns the network interface on and persists the state “up” to the settings. Prefix **no** turns the the network interface off and and deletes “up” from settings.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

3.8.5 interface down

Turns the network interface on and persists the state “down” to the settings. Prefix **no** turns the network interface off and deletes “down” from settings.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

3.8.6 interface mac address

Sets the MAC-address to the specified network interface. Address is specified in hexadecimal format 00:00:00:00:00:00. The command allows one to assign arbitrary address, but warns the user if the new address “multicast” bit is set or “OUI enforced” bit is cleared.

Command with prefix **no** resets the original MAC-addresses on the interface.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	MAC

```
(config-if)> mac address <address>
```

```
(config-if)> no mac address
```

Argument	Type	Description
<i>address</i>	MAC-address	New MAC-address of the interface

3.8.7 interface ip address

Changes the IP-address and the mask of the network interface. If the interface is running an automated address setup service such as, for instance, DHCP-client, (see [interface ip dhcp](#)), then the manually set address can be overwritten. Prefix **no** resets the address at 0.0.0.0.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	IP

```
(config-if)> ip address <address> <mask>
```

```
(config-if)> no ip address
```

Argument	Type	Description
<i>address</i>	IP-address	Network interface address.
<i>mask</i>	IP-mask	Network interface mask. There are two ways to specify the mask: the canonical form (for example, 255.255.255.0) and the prefix with bit length (for example, /24).

Example 3.1. The two ways of specifying IP-address and mask

The network address, defined by the IP-address and mask, can be specified in either of the two ways: specify a mask in the canonical form, or set the prefix bit length.

```
(config)> interface Switch0/VLAN43
Created interface Switch0/VLAN43.
(config-if)> ip address 172.17.24.9 255.255.255.0
Network address saved.
(config-if)> ip address 172.17.24.9/24
Network address saved.
(config-if)> [Ctrl]+[D]
(config)> show interface Switch0/VLAN43

        mac: 00:23:f8:5b:d3:f4
        index: 43
        type: VLAN
description:
        state: up
        link: down
        address: 172.17.24.9
        mask: 255.255.255.0
        mtu: 1500
        global: no

(config)>
```

3.8.8 interface ip alias

Sets additional IP-address and mask of the network interface (alias). Prefix **no** resets the specified alias to 0.0.0.0, effectively removing it. If no arguments are specified when using prefix **no**, the command removes all aliases.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	IP

```
(config-if)> ip alias <address>
```

```
(config-if)> no ip alias [address]
```

Argument	Type	Description
<i>address</i>	IP-address	Additional address of the network interface.

3.8.9 interface ip dhcp

Starting the DHCP-client to automatically configure the network parameters: IP-address and mask of the interface, DNS servers and [default gateway](#). Prefix **no** stops the DHCP-client,

removes the dynamically configured settings and restores the previous settings of IP-address and mask.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Ethernet

```
(config-if)> ip dhcp
```

```
(config-if)> no ip dhcp
```

3.8.10 interface ip mtu

Changes the MTU value on the network interface. Prefix **no** resets the MTU value to that which was before the first use of the command. When establishing a connection via PPP (IPCP), packets with defined MTU size will be sent to the remote host, even if the host requested a lower MTU value.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	IP

```
(config-if)> ip mtu <mtu>
```

```
(config-if)> no ip mtu
```

Argument	Type	Description
<i>mtu</i>	Integer	MTU value. Range of permissible values — from 64 to 65535.

3.8.11 interface ip tcp adjust-mss

Sets the limit on the maximum segment size of outgoing TCP sessions, changing the value in the header of SYN-packets if the MSS value transmitted in them exceeds the specified limit. This command is applied to interface and affects all outgoing TCP SYN packets.

Prefix **no** rolls the command effect back.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

Properties	
Interface Type	IP

```
(config-if)> ip tcp adjust-mss (pmtu | <mss>)
```

```
(config-if)> no ip tcp adjust-mss
```

Argument	Type	Description
pmtu	Keyword	Set the upper limit of MSS, equal to the minimum MTU along the path to the remote node.
mss	Integer	MSS upper limit.

3.8.12 ip route

Configuring IP Static Routes. The command adds a static route to the routing table to describe a rule of IP-packets transmission through a particular gateway or network interface.

As the destination network, one can specify default keyword. In this case, a default route will be created.

Prefix **no** removes the route with the specified parameters.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip route (<address> <mask> | <host> | default)
(<gateway> [interface] | <interface>) [metric]
```

```
(config)> no ip route (<address> <mask> | <host> | default)
[<gateway> | <interface>] [metric]
```

Argument	Type	Description
address	IP-address	IP-address of the destination network.
mask	IP-mask	Mask of the destination network. There are two ways to enter the mask: in the canonical form (for example, 255.255.255.0) and in the form of prefix bit length (for example, /24).
host	IP-address	IP-address of the destination node.
default	Keyword	Helps specify default routes.
interface	Interface Name	Interface full name or an alias. Specified as the direction of the packet transferring, if the interface has a point-to-point channel connected that requires no additional addressing within the channel.

Argument	Type	Description
		If priority interface ip global is set on the interface, the route is added to the system table only if there is no other higher priority route with the same address.
<i>gateway</i>	IP-address	IP-address of the router in a directly connected network. Can be specified along with the interface name, if it is required to specify interface ip global priority. If no interface is specified, the systemd determines it automatically based on the current IP settings.
<i>metric</i>	Integer	Route metrics. Ignored in the In the current implementation.

3.8.13 interface ip global

Sets property “global” with a parameter to the interface. This property is necessary for setting the default route, DynDNS-Client and NAT functioning. Can represent global-interfaces as facing the global network (the Internet).

Property “global” affects the interface priority in setting the default route. The higher the priority the more desirable it is for the user to access the global network through the specified interface. Internet access backup (WAN backup) functionality is using priority “global”.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	IP

```
(config-if)> ip global <priority>
```

```
(config-if)> no ip global
```

Argument	Type	Description
<i>priority</i>	Integer	Interface priority when setting the default route.

3.8.14 IPv6 settings

This section describes the commands required to configure IPv6.

IPv6 is a new version of Internet Protocol, designed to solve the problems faced by the previous version (IPv4), by using 128-bit address length instead of 32. Both stacks of protocols - IPv6 and IPv4 - will be used simultaneously.

Simplified processing by routers:

- IPv6 routers do not perform fragmentation. IPv6 hosts are required to either perform path MTU discovery, perform end-to-end fragmentation, or to send packets no larger than the IPv6 default minimum MTU size of 1280 octets.

- Disappeared the checksum. IPv6 routers do not need to recompute a checksum when header fields (such as the time to live (TTL) or hop count) change.
- The packet header in IPv6 is simpler than that used in IPv4, with many rarely used fields moved to separate optional header extensions.
- The TTL field of IPv4 has been renamed to Hop Limit, reflecting the fact that routers are no longer expected to compute the time a packet has spent in a queue.
- IPv4 limits packets to 65535 (216–1) octets of payload. An IPv6 node can optionally handle packets over this limit, referred to as jumbograms, which can be as large as 4294967295 (232–1) octets. The use of jumbograms may improve performance over high-MTU links.

Despite the huge size of IPv6 address, thanks to these improvements, the packet header is only lengthened by half: from 20 to 40 bytes.

3.8.14.1 interface ipv6 address

Configuring an IPv6 address on the interface. If the argument is **auto**, address is autoconfigured. Likewise, with **dhcp**, an address is requested via DHCPv6-NA. Passing a literal address as an argument will assign it statically.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config-if)> ipv6 address (<address> | auto | dhcp)
```

```
(config-if)> no ipv6 address [<address> | auto | dhcp]
```

Argument	Type	Description
<i>address</i>	IPv6-address	Name server address.
<i>auto</i>	Keyword	Enable stateless autoconfiguration.
<i>dhcp</i>	Keyword	Enable stateful autoconfiguration, start DHCP client.

3.8.14.2 interface ipv6 prefix

Configuring prefix delegation. When **dhcp** is set, prefix is requested via DHCPv6-PD.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-if)> ipv6 prefix (dhcp)
```

```
(config-if)> no ipv6 prefix [dhcp]
```

Argument	Type	Description
<i>dhcp</i>	Keyword	Enable prefix delegation.

3.8.14.3 interface ipv6 name-servers

Configuring retrieval of DNS information. When **dhcp** is set, enables DHCPv6 name-server requests.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-if)> ipv6 name-servers (dhcp)
```

```
(config-if)> no ipv6 name-servers [dhcp]
```

Argument	Type	Description
<i>dhcp</i>	Keyword	Enable name-server autoconfiguration.

3.8.14.4 ipv6 local-prefix

Configuring a local (ULA) prefix. Argument can be a literal prefix or **default**, which generates a persistent unique prefix automatically.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> ipv6 local-prefix (default | <prefix>)
```

```
(config)> no ipv6 local-prefix [default | <prefix>]
```

Argument	Type	Description
<i>default</i>	Keyword	Generate persistent unique prefix.
<i>prefix</i>		Local ULA prefix. Must be a valid prefix in the block fd00::/8 with a prefix length no longer than 48.

3.8.14.5 ipv6 name-server

Configuring DNS server IPv6-addresses. Addresses saved in this fashion are called *static* as opposite to *dynamic* — as registered by PPP or DHCP services.

ipv6 name-server command can be entered multiple times if several DNS-server addresses need to be setup.

Prefix **no** removes the specified DNS server address from the static and the active lists if the command is furnished with arguments, or clears the list of static addresses if the command has no arguments.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ipv6 name-server <address>
```

```
(config)> no ipv6 name-server [<address>]
```

Argument	Type	Description
<i>address</i>	IPv6-address	Name server address.

3.8.14.6 ipv6 subnet

Creates or configures a LAN IPv6 segment.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Access to group	(config-subnet)

```
(config)> ipv6 subnet <name>
```

```
(config)> no ipv6 subnet [<name>]
```

Argument	Type	Description
<i>name</i>	String	Subnet name or an alias.

3.8.14.7 ipv6 subnet bind

Binds the subnet to an interface.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-subnet)> bind <interface>
```

```
(config-subnet)> no bind [<interface>]
```

Argument	Type	Description
<i>interface</i>	String	Full interface name or an alias.

3.8.14.8 ipv6 subnet mode

Chooses the address configuration mode for hosts in the subnet. Exclusive options are **dhcp** and **slaac**. The former will enable a local DHCPv6 server for the purposes of address assignment, and the latter will enable SLAAC (Stateless Address Autoconfiguration)

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-subnet)> mode (slaac | dhcp)
```

```
(config-subnet)> no mode (slaac | dhcp)
```

Argument	Type	Description
<i>slaac</i>	Keyword	Enable SLAAC (stateless autoconfiguration).
<i>dhcp</i>	Keyword	Enable DHCPv6 server (stateful autoconfiguration).

3.8.14.9 ipv6 subnet number

Configuring the Subnet ID, which will determine the advertised prefix for the segment. Must be unique across subnets.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-subnet)> number <n>
```

```
(config-subnet)> no number [<n>]
```

Argument	Type	Description
<i>n</i>	ID	Unique subnet ID.

3.8.14.10 ipv6 subnet stateless-dhcp

Enables a local DHCPv6 server for the purposes of network information (i.e. DNS servers) delivery.

Properties	
Prefix no	yes

Properties	
Changes the settings	yes
Multiple entry	no

```
(config-subnet)> stateless-dhcp
```

3.8.14.11 tools ping6

Sends Echo-Request requests of ICMPv6 protocol to specified network node and registers received Echo-Reply responses. The time between sending request and receiving the response Round Trip Time (RTT) allows you to define double ended delays on the route and frequency of packet losses, that is, indirectly determine loading on the channels of data transmission and intermediate devices.

Total absence of ICMP-replies can also mean that the remote node (or any of the intermediate routers) blocks ICMP Echo-Reply or ignores ICMP Echo-Request.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> tools ping6 <host> [count <count>] [size <packetsize>]
```

Argument	Type	Description
<i>host</i>	String	Domain name or host IP-address.
<i>count</i>	Integer	Number of ICMPv6 Echo requests. If not specified, the command will run until interrupted by the user.
<i>packetsize</i>	Integer	Size of the ICMPv6 Echo-Request data field in bytes. By default — 56, which together with the 8-byte header specifies the size of the ICMPv6-packet — 64 bytes.

Firewall and NAT address translation

4.1 Command description

4.1.1 ip nat

Turns on translation of “local” addresses of network *network* or network behind the interface *interface*. For example, command `ip nat Home` means that all packets from the network Home, passing through the router will undergo IP spoofing.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip nat (<interface> | <address> <mask>)
```

```
(config)> no ip nat (<interface> | <address> <mask>)
```

Argument	Type	Description
<i>interface</i>	Interface Name	Source interface name (full name or an alias)
<i>address</i>	IP-address	Together with mask <i>mask</i> sets the range of source IP-addresses to be translated.
<i>mask</i>	IP-mask	Mask of a translation range. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).

4.1.2 ip static

Specifies static linking of local IP-addresses to the global ones. If *interface* or *network* corresponds to the global interface, then the source address translation (SNAT) will run. If *to-address* corresponds to the global interface, then destination address translation (DNAT) will run. TCP/UDP port number is always treated as the destination port.

If *network* corresponds to a single address and this address is equal *to-address*, then this rule will prohibit the translation of the specified address, which could have been done based on the specified rules **ip nat**.

ip static rules have higher priority than the **ip nat** rules.

Example 4.1. Redirection of the incoming requests, command **ip static**

Let there be a router between the “local” network 172.16.1.0/24 and “global” network 10.0.0.0/16. It is required that all requests coming to the “global” interface of this router on port 80 to be broadcast to the “local” server with the address 172.16.1.33. The sequence of commands to implement the required schema might look like this:

```
interface Home
    ip address 172.16.1.1/24
!
interface Internet
    ip address 10.0.0.1/16
    ip global 1
!
ip nat Home
ip static tcp Internet 80 172.16.1.33 80
```

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip static [protocol] (<interface> | <address> <mask>) [port]
<to-address> [to-port]
```

```
(config)> no ip static [[protocol] (<interface> | <address> <mask>) [port]
<to-address> [to-port]]
```

Argument	Type	Description
<i>protocol</i>	String	IP-protocol: tcp or udp. Used in conjunction with parameter <i>port</i> . If not specified, the translation will be performed for all protocols.
<i>interface</i>	String	Input interface name (full name or alias).
<i>address</i>	IP-address	Along with mask <i>mask</i> sets the range of destination IP-addresses that are to be translated.
<i>mask</i>	IP-mask	Translation range mask. There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).
<i>port</i>	Integer	TCP/UDP port number for which a translation request comes. If one is not specified, all incoming requests will be translated.
<i>to-address</i>	IP-address	The destination address after translation.
<i>to-port</i>	Integer	TCP/UDP port number after translation. If one is not specified, the destination port remains the same.

4.1.3 interface security-level

Specifies the interface security level. The security levels define the firewall logic:

- Allow establishing `private` → `public` connections.
- Prohibit establishing connections coming to the interface `public`, i. e. in the direction `public` → `private` and `public` → `public`.
- The device itself accepts network connections (allows control) only from `private` interfaces.
- Data transfer between `private` interfaces can be allowed or disallowed depending on the global parameter `isolate-private` setting.

Note: By default, all newly created interfaces are set with `public` security level.

Note: Access lists `access-list` have higher priority than the security levels, so they can be used to set additional rules of packet filtering.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no

```
(config-if)> security-level (public | private)
```

4.1.4 isolate-private

Command prohibits data transfer between any interfaces with `security level` `private`. Prefix **no** cancels the command, allowing data transfer between `private` interfaces.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> isolate-private
```

```
(config)> no isolate-private
```

4.1.5 access-list

Packet filtering rules setup. The command creates a list of rules with the specified name, to which you can add allowing and denying rules by using commands `permit` and `deny` respectively. Such a list can be assigned to a network interface using command `interface ip access-group`.

Prefix **no** removes the list of rules.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Group entry	(config-acl)

```
(config)> access-list <acl>
```

```
(config)> no access-list <acl>
```

Argument	Type	Description
<i>acl</i>	String	Filtering rules list name(Access Control List , ACL).

4.1.5.1 access-list deny

The command adds a packet filtering deny rule into a specified [ACL](#). Prefix **no** removes the rule.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config-acl)> deny (tcp | udp) <source> <source-mask> [port [lt | eq | gt] <source-port>]
<destination> <destination-mask> [port [lt | eq | gt] <destination-port>]
```

```
(config-acl)> deny icmp <source> <source-mask> <destination> <destination-mask>
```

```
(config-acl)> no deny (tcp | udp) <source> <source-mask> [port [lt | eq | gt] <source-port>]
<destination> <destination-mask> [port [lt | eq | gt] <destination-port>]
```

```
(config-acl)> no deny icmp <source> <source-mask> <destination> <destination-mask>
```

4.1.5.2 access-list permit

The command adds a packet filtering allowing rule into a specified [ACL](#). Prefix **no** removes the rule.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config-acl)> permit (tcp | udp) <source> <source-mask> [port [lt | eq | gt] <source-port>]
<destination> <destination-mask> [port [lt | eq | gt] <destination-port>]
```

```
(config-acl)> permit icmp <source> <source-mask> <destination> <destination-mask>
```

```
(config-acl)> no permit (tcp | udp) <source> <source-mask> [port [lt | eq | gt]
<source-port>]
<destination> <destination-mask> [port [lt | eq | gt] <destination-port>]

(config-acl)> no permit icmp <source> <source-mask> <destination> <destination-mask>
```

4.1.5.3 permit and deny command arguments

Commands **permit** and **deny** use the same set of arguments shown in the table below.

Argument	Type	Description
tcp	Keyword	TCP protocol
udp	Keyword	UDP protocol
icmp	Keyword	ICMP protocol
source	IP-address	The source address in the header of IP-packet.
source-mask	IP-mask	Mask to be applied to the source address in the header of IP-packet before comparison with <i>source</i> . There are two ways to enter the mask: the canonical form (for example, 255.255.255.0) and the form of prefix bit length (for example, /24).
source-port	Integer	Source port in the TCP or UDP header.
destination	IP-address	The destination address in the header of IP-packet.
destination-mask	IP-mask	Mask to be applied to the destination address in the header of IP-packet before comparison with <i>destination</i> . There are two ways to enter the mask: in the canonical form (for example, 255.255.255.0) and in the form of prefix with bit length (for example, /24).
destination-port	Integer	Destination port in the TCP or UDP header.
port	Keyword	
lt	Keyword	Operator "less" to compare the port with the specified value <i>source-port</i> or <i>destination-port</i> .
eq	Keyword	Operator equal to compare the port with the specified value <i>source-port</i> or <i>destination-port</i> .
gt	Keyword	Operator "greater" to compare the port with the specified value <i>source-port</i> or <i>destination-port</i> .

4.1.6 interface ip access-group

Assigns a named list of filtering rules (**ACL**, see **access-list**) to the interface. Parameter in or out indicates the traffic direction for which the **ACL** will be applied. Several ACL's can be assigned to a single interface.

Prefix **no** disables the **ACL** for the specified interface and traffic direction.

Properties	
Prefix no	yes

Properties	
Changes the settings	yes
Multiple entry	yes

```
(config-if)> ip access-group <acl> (in | out)
```

```
(config-if)> no ip access-group <acl> (in | out)
```

Argument	Type	Description
<i>acl</i>	ACL name	List of filtering rules as previously created using access-list command.
<i>in</i>	Keyword	Apply filtering to incoming packets.
<i>out</i>	Keyword	Apply filtering to outgoing packets.

4.2 Examples

Example 4.2. Access to the device control through the interface public

Suppose that Keenetic is connected to the internet via PPPoE0 interface with security level public, and that it should be allowed to log on to the device web-interface via PPPoE0. By default, access to the device control via public interfaces is prohibited, so it is necessary to create an allowing rule and assign it to the interface PPPoE0.

```
(config)> access-list REMOTE_ACCESS
(config-acl)> permit tcp 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 port eq 80
(config-acl)> exit
(config)> interface PPPoE0
(config-if)> ip access-group REMOTE_ACCESS in
```

Example 4.3. Disabling the firewall between two interfaces

Despite the fact that there is no functionality to turn off the firewall completely, it is possible to disable it for particular directions. Suppose that it is necessary to allow data transfer between the “home” network Home and global network PPPoE0. To accomplish that, both interfaces must be assigned security level private and function [isolate-private](#) must be disabled.

```
(config)> interface Home security-level private
(config)> interface PPPoE0 security-level private
(config)> no isolate-private
```

Note: Oftentimes, one does not realize that the firewall and the address translation — are the functions designed to address fundamentally different problems. Enabling NAT between Home and PPPoE0 interfaces in the configuration shown above, does not prohibit access to the network Home from the global network. Even as the address translation is enabled by command `ip nat Home`, the packets from PPPoE0 will get to Home network.

Example 4.4. Redirecting incoming connections for “home” network

Suppose that “home” network uses private IP-addresses 192.168.1.0/24. Connecting to the Internet is carried via the interface PPPoE0, which has an external IP-address assigned by provider. One of the most common tasks given this configuration —to redirect incoming connection from PPPoE0 interface to a specific node of “home” network.

To accomplish this goal the two rules could be created: **ip static** to redirect incoming connections and **access-list permit** — allowing rule for the firewall.

Supposet that the address and the port of the local server — 192.168.1.110:80, and that the connections come through PPPoE0 port 8080.

```
(config)> access-list VIRTUAL_SERVERS
(config-acl)> permit tcp 0.0.0.0 0.0.0.0 192.168.1.110 255.255.255.255 port ►
eq 80
(config-acl)> exit
(config)> interface PPPoE0
(config-if)> ip access-group VIRTUAL_SERVERS in
(config-if)> exit
(config)> ip static tcp PPPoE0 8080 192.168.1.110 80
```

Note: In the allowing rules **access-list** which are used to redirect incoming connections, it is necessary to specify the address and port of the server of “home” network, because the filters are applied after the address translation.

Note: To redirect incoming connections it is required that property **global** is set on input interface (in the example, PPPoE0) .

Switch and VLAN Interfaces

5.1 Switch Interface

Switch interface type corresponds to a IEEE 802.1D hardware switch, which has a certain set of ports — jacks on the physical device. Switch allows for plugging ports into various virtual local networks VLAN IEEE 802.1Q.

Each switch port is configured by the commands of group **interface port**, in which one can specify the connection parameters, access or trunk working mode as well as enumerate the identifiers of VLAN to which the port belongs.

In the access mode the port can belong to a single VLAN only, in which case the Ethernet frames transmitted to the network are not marked with tag 802.1Q. In the trunk mode the port can belong to several VLAN's, in which case the frames to be transmitted are tagged.

5.2 VLAN interface

VLAN interface type allows one to work with a virtual network IEEE 802.1Q over IP protocol. VLAN is created dynamically using the interface of **Ethernet** class (this includes Bridge, SSID and etc.) or **Switch** as a parent. For example:

```
(config)> interface Switch0/VLAN100
Created interface Switch0/VLAN100.
(config-if)>
```

The index of the VLAN interface (in this example — 100) sets the ID of VLAN 802.1Q. If VLAN is created over the Ethernet interface, then its outgoing frames are transmitted with a marker. If VLAN is created over the Switch interface, then marking of the outgoing frames depends on the switch settings (see commands of group **interface port**).

VLAN belongs to **Ethernet** class, ie it aggregates properties of the classes **IP** and **MAC**. Thus, on the VLAN interface one can configure IP-address, MAC-address, make it a part of a bridge and so on. By default VLAN inherits MAC-address of the parent interface.

Example 5.1. Configuring VLAN with routing

On the switch Switch0 it is necessary to configure two VLANs with IDs 11 and 202 in access mode. Create interface on the first VLAN Switch0/VLAN11 with IP-address 172.16.1.1/24, on the second one — interface Switch0/VLAN202 where DHCP-client is running. Turn on ports 1 and 2 on the 11th VLAN, and port 3 on the 202th.

```
(config)> interface Switch0
(config-if)> port 1
(config-if-port)> mode access
(config-if-port)> access vlan 11
```

```

(config-if-port)> exit
(config-if)> port 2
(config-if-port)> mode access
(config-if-port)> access vlan 11
(config-if-port)> exit
(config-if)> port 3
(config-if-port)> mode access
(config-if-port)> access vlan 202
(config-if-port)> exit
(config-if)< up
(config-if)< exit
(config)> interface Switch0/VLAN11
(config-if)> ip address 172.16.1.1/24
(config-if)> up
(config-if)> exit
(config)> interface Switch0/VLAN202
(config-if)> ip dhcp
(config-if)> up
(config-if)> exit
(config)>

```

5.3 Command description

5.3.1 interface port

The group of commands for configuring a hardware switch port. Port number is specified as the argument. Prefix **no** is not applicable.

Properties	
Prefix no	no
Changes the settings	no
Multiple entry	yes
Interface Type	Switch
Group entry	(config-if-port)

```
(config-if)> port <port>
```

Argument	Type	Description
<i>port</i>	Integer	Switch port number

5.3.2 interface port speed

Sets the connection speed in Mbit/sec. This command is used in cases when due to incompatibility or poor cable quality the connection is not determined or wrong speed is determined on some side.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no
Interface Type	Switch

```
(config-if-port)> speed (10 | 100 | auto)
```

Argument	Type	Description
10	Keyword	Manually set the speed to 10 Mbit/sec.
100	Keyword	Manually set the speed to 100 Mbit/sec.
auto	Keyword	Turn on automatic speed control.

Note: Due to hardware limitations the command has a feature: when the speed is set manually, the auto-detection of connectivity stops working. Even if the cable is not connected, the device will indicate its presence. Duplex autodetection stops working as well, so it is recommended to use the command together with the [interface port duplex](#).

5.3.3 interface port duplex

Specifies the bidirectional communication mode: duplex or half-duplex. This command is used in the cases when due to incompatibility or poor cable quality connection is not determined or wrong mode is determined on some side.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no
Interface Type	Switch

```
(config-if-port)> duplex (half | full | auto)
```

Argument	Type	Description
half	Keyword	Manually set the half-duplex transmission mode.
full	Keyword	Manually set the full-duplex transmission mode.
auto	Keyword	Enable duplex auto-detect mode.

Note: Due to hardware limitations the command has a feature: when the duplex is set manually, connection auto-detection stops working. Even if the cable is not connected, the device will indicate its presence. Speed auto-detection stops working as well, so it is recommended to use the command together with the [interface port speed](#).

5.3.4 interface port mode

Command to select the tag processing mode of the switch VLAN-port: either *access mode*, or *trunk mode*.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Switch

```
(config-if-port)> mode (access | trunk)
```

```
(config-if-port)> no mode (access | trunk)
```

Argument	Type	Description
access	Keyword	Turn on the access mode to a virtual local network VLAN, that is the mode when only the untagged frames pass through the port. The incoming frames get tagged with the PVID marker set using port access command. The port is an output one only for VLAN with PVID ID. Once a frame is transferred to the port, the VLAN marker gets removed.
trunk	Keyword	Turn on the VLAN multiplexing mode, that is the mode when frames belonging to several VLANs get transmitted through the port. In this case each frame gets tagged. The list of IDs of VLAN networks that include the port is set with port trunk command.

Note: Due to hardware limitations the port cannot be in the access and multiplexing mode at the same time. Therefore, the processing of tagged and untagged frames on a single port is impossible.

5.3.5 interface port access

Setting of port VLAN ID for access mode. Sets default VLAN ID of port (PVID), allows transferring of the frames of the specified VLAN to the port and turns on the removing of VLAN marker from the transferred frames. Prefix **no** removes the setting.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Switch

```
access vlan <vid>
```

no access vlan

Argument	Type	Description
<i>vid</i>	Integer	Access VLAN ID. Allowed values range — from 1 to 4094.

5.3.6 interface port trunk

Adding a port to the VLAN allows for receiving and transmitting the frames of the given VLAN to the port such that VLAN marker from the transmitted frames is not removed. In the `trunk` mode it is allowed to add a port to several VLANs.

Command with **no** prefix removes the port from the specified VLAN or from all the VLANs, if *vid* is not specified.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	Switch

trunk vlan *<vid>*

no trunk vlan [*vid*]

Argument	Type	Description
<i>vid</i>	Integer	VLAN ID. Allowed values range — from 1 to 4094.

Bridges

6.1 Bridge interface

Interface of Bridge type is a dynamic interface of virtual bridge IEEE 802.1D, to which you can add interfaces of [Ethernet](#) class (including VLAN, SSID, etc.). Bridge is implemented in software, by means of router operating system.

Bridge belongs to [Ethernet](#) class, i.e. it aggregates properties of the [IP](#) and [MAC](#) classes. Thus, on the Bridge interface one can be configure IP-address, MAC-address and so on. By default Bridge inherits the MAC-address from the first interface added.

The interface added to the bridge gets IP settings blocked. It also loses the ability to transmit IP packets directly because it gets connected with other interfaces added to the bridge on the channel level. There are two ways to add interface to the bridge:

- [interface include](#) — losing IP settings.
- [interface inherit](#) — transferring IP settings to the bridge interface.

Tip: The second method is convenient because when configuring the router over the network one can add to the bridge even the interface through which the device control is executed. In this case the control session is not disrupted.

Note: The bridge works much slower than the hardware switch [Switch](#). It is needed to unite some disparate Ethernet interfaces, for example, a wireless network and VLAN.

Example 6.1. Several wireless Wi-Fi networks and VLAN

Two VLANs with IDs 10 and 20 in trunk mode are configured on the switch Switch0. On the wireless adapter WifiMaster0 there configured two access points: WifiMaster0/AccessPoint0 and WifiMaster0/AccessPoint1. Using bridges Bridge0 and Bridge1 the packets from the wireless networks get transmitted to the port 5 of switch Switch0 with VLAN markers 10 and 20 respectively.

A fragment of the configuration file:

```
interface Switch0
    port 5
        mode trunk
        trunk vlan 10
        trunk vlan 20
    !
!
interface Switch0/VLAN10
    up
```

```

!
interface Switch0/VLAN20
    up
!
interface WifiMaster0/AccessPoint0
    ! здесь настраиваются параметры
    ! первой беспроводной сети
!
interface WifiMaster0/AccessPoint1
    ! здесь настраиваются параметры
    ! второй беспроводной сети
!
interface Bridge0
    include Switch0/VLAN10
    include WifiMaster0/AccessPoint0
    up
!
interface Bridge1
    include Switch0/VLAN20
    include WifiMaster0/AccessPoint1
    up
!

```

6.2 Command description

6.2.1 interface include

Specifies Ethernet-interface name which will be added to the software bridge as a port. Prefix **no** removes the interface from the bridge.

See also [inherit](#) command, which allows one to pass to the bridge some settings of the interface being added, such as IP-address, mask and IP-aliases. This allows for adding an interface to the bridge, with device control being conducted via that interface without losing the control.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	Bridge

```
(config-if)> include <interface>
```

```
(config-if)> no include <interface>
```

Argument	Type	Description
<i>interface</i>	Interface Name	Name or alias of Ethernet-interface, which should be hooked to the bridge.

6.2.2 interface inherit

Specifies the name of the Ethernet-interface that will be added to the program bridge as a port. In contrast with the **include** command, **inherit** command transfers some settings of the interface being added to the bridge, such as IP-address, mask and IP-aliases. On removing either the bridge itself or the bridge interface, these settings, even if they have been changed will be copied back to the vacant interface.

Command with **no** prefix removes the interface from the bridge, returning the settings that have earlier been inherited by the bridge, back to the interface, and resets these settings on the bridge.

The command allows one to add the device control interface to the bridge such that control is not lost.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	Bridge

```
(config-if)> inherit <interface>
```

```
(config-if)> no inherit <interface>
```

Argument	Type	Description
<i>interface</i>	Interface Name	Name or alias of the Ethernet-interface that should be plugged into the bridge.

7.1 PPP functions

Interfaces of PPP class aggregate [Secure](#) and [IP](#) functions and add their own set of settings for the functions to support of PPP protocol. PPP is used to establish a direct connection between two network nodes, such that it can provide for connection authentication, data encryption and compression. In particular, PPP is used to connect to the Internet.

7.2 Functions Secure

Interfaces of Secure class provide for basic authentication settings that are used in the interfaces of [PPP](#) class to connect to some remote node.

7.3 PPPoE interface

PPPoE — channel layer network protocol for transferring PPP frames over Ethernet.

7.4 PPTP Interface

PPTP (Point-to-Point Tunneling Protocol) — tunneling protocol that works in IP networks using PPP mechanisms to implement the authentication, compression and encryption.

PPTP interfaces support MPPE encryption, which can be turned on by command [interface encryption mppe](#).

7.5 L2TP Interface

L2TP (Layer 2 Tunneling Protocol) — tunneling protocol that uses PPP to establish a virtual connection between two network nodes such that packets are switched similarly to PPTP.

7.6 Configuration sequence

Keenetic supports several simultaneous PPP-connections via PPPoE, PPTP and L2TP. Each connection has a corresponding network interface with index, for example: PPPoE0. Configuring of basic connection settings is done as follows.

1. Creating network interface using interface command:

```
(config)> interface PPPoE0
(config-if)>
```

2. Entering the name of a remote access hub using `peer` command:

```
(config-if)> peer tp.example.net
Using peer tp.example.net.
```

3. Entering the supported authentication protocols (PAP, CHAP, MS-CHAP, MS-CHAPv2):

```
(config-if)> authentication chap
CHAP authentication enabled.
(config-if)> authentication mschap
MSCHAP authentication enabled.
(config-if)> authentication mschap-v2
MSCHAPv2 authentication enabled.
```

4. Entering user name and password to authenticate with a remote access hub:

```
(config-if)> authentication identity 00889@example.net
Identity saved.
(config-if)> authentication password P@sSW0Rd
Password saved.
```

5. To connect via PPPoE protocol, enter the name of the Ethernet interface through which the connection takes place. To connect via PPTP and L2TP protocols — enter the name of the IP interface, through which there will be created a host route to the PPP hub using the tunnel as default route¹:

```
(config-if)> connect via Switch0/VLAN2
```

Connect command starts the connection process. Keenetic will maintain a connection to a remote node, trying to restore it each time the remote host breaks it.

7.7 Additional options

7.7.1 LCP Echo

By default, the PPP interface has enabled the connection control function using the LCP Echo-Request and Echo-Reply packages. **lcp echo** command allows one to set interval of sending Echo-Request requests as well as the threshold on the number of missed Echo-Reply responses at which point the connection is considered broken.

The LCP Echo function can be turned off using the **lcp echo** command with **no** prefix:

```
(config-if)> no lcp echo
LCP echo disabled.
```

¹Without creating such a host route the functioning of tunnel will be impossible. If interface *via* is not specified, the route to the hub PPTP or L2TP will be determined automatically, using the most appropriate gateway. If there is no such gateway, the tunnel will not be used as the default route.

7.7.2 CCP

Compression Control Protocol (CCP) can be used during connection process. By default it is turned off, but the remote node may request it. In this case, one can turn it on using the **ccp** command:

```
(config-if)> ccp
CCP enabled.
```

Tip: When connecting via PPP, a large number of parameters is getting agreed upon and this process can fail. To debug such problems, use command **interface debug**. System log will display detailed information about the failure cause.

7.7.3 IPCP

In the process of agreeing on parameters, the IP router Keenetic receives from provider some **default gateway** address as well as DNS-servers addresses. If the PPP connection is used to connect to the Internet, the obtained addresses are stored in the system. However, in some cases, they should be ignored. To this end, during the connection setup one has to invoke the commands **interface ipcp default-route** and **interface ipcp name-servers** with **no** prefix during configuring connection:

```
(config-if)> no ipcp default-route
Not using peer as a default gateway.
(config-if)> no ipcp name-servers
Not using remote name servers.
```

7.8 Command description

7.8.1 interface peer

Specifies ID of the remote node to which the PPP connection will be used. A more precise meaning of configuration depends on interface type. For example, for PPPoE the **interface peer** command specifies the name of access hub, and for PPTP — remote host name or IP-address.

Prefix **no** cancels the setting.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> peer <peer>
```

```
(config-if)> no peer
```

Argument	Type	Description
<i>peer</i>	String	Remote connection point ID

7.8.2 interface connect

Starts the process of connecting to a remote node. The *via* parameter specifies the connection interface.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> connect [via <via>]
```

```
(config-if)> no connect
```

Argument	Type	Description
<i>via</i>	Interface Name	Interface through which remote node is accessed. For PPPoE this option is mandatory.

7.8.3 interface authentication pap

Turns on PAP authentication support. Prefix **no** turns PAP off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Secure

```
(config-if)> authentication pap
```

```
(config-if)> no authentication pap
```

7.8.4 interface authentication chap

Turns on CHAP authentication support. Prefix **no** turns CHAP off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

Properties	
Interface Type	Secure

```
(config-if)> authentication chap
```

```
(config-if)> no authentication chap
```

7.8.5 interface authentication mschap

Turns MS-CHAP authentication on. Prefix **no** turns MS-CHAP off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Secure

```
(config-if)> authentication mschap
```

```
(config-if)> no authentication mschap
```

7.8.6 interface authentication mschap-v2

Turns MS-CHAPv2 authentication support on. Prefix **no** turns MS-CHAPv2 off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	Secure

```
(config-if)> authentication mschap-v2
```

```
(config-if)> no authentication mschap-v2
```

7.8.7 interface authentication identity

Specifies user name for device authentication on the remote system. Equally often used on PPTP, PPPoE and L2TP connections.

Command with **no** prefix deletes the previously specified user name.

Properties	
Prefix no	yes
Changes the settings	yes

Properties	
Multiple entry	no
Interface Type	Secure

```
(config-if)> authentication identity <identity>
```

```
(config-if)> no authentication identity
```

Argument	Type	Description
<i>identity</i>	String	User name for authentication

7.8.8 interface authentication password

Specifies password for device authentication on the remote system. Equally often used on PPTP, PPPoE and L2TP connections.

Command with **no** prefix deletes the password.

Properties	
Prefix no	
Changes the settings	
Multiple entry	
Entrance to a group	

```
(config-if)> authentication password <password>
```

```
(config-if)> no authentication password
```

Argument	Type	Description
<i>password</i>	String	Password for authentication

7.8.9 interface encryption mppe

Turns on MPPE encryption support. Prefix **no** turns MPPE encryption off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPTP

```
(config-if)> encryption mppe
```

```
(config-if)> no encryption mppe
```

7.8.10 interface lcp echo

Specifies the testing rules of the PPP connection by LCP echo tools. Prefix **no** turns LCP echo off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> lcp echo <interval> <count>
```

```
(config-if)> no lcp echo
```

Argument	Type	Description
<i>interval</i>	Integer	Interval between sending LCP echo, in seconds. If within the specified time interval there is no LCP echo request from the remote location, the same request will be sent there asking for response LCP reply.
<i>count</i>	Integer	The number of consecutive requests LCP echo sent, for which no response LCP reply was received. If count of LCP echo requests goes unanswered, the connection is terminated.

7.8.11 interface ccp

Turn CCP (Compression Control Protocol) support on during establishing connection. Prefix **no** turns CCP off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> ccp
```

```
(config-if)> no ccp
```

7.8.12 interface ipcp default-route

Use the remote node address as *default gateway*. Prefix **no** prohibits changing *default gateway* while connecting to remote node.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> ipcp default-route
```

```
(config-if)> no ipcp default-route
```

7.8.13 interface ipcp name-servers

Use DNS servers addresses as obtained via IPCP. Prefix **no** prohibits changing DNS settings while connecting to remote node.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> ipcp name-servers
```

```
(config-if)> no ipcp name-servers
```

7.8.14 interface debug

Turn on debug mode of PPP connection. In debug mode detailed info about connection progress is saved to the system log. Prefix **no** turns debug mode off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> debug
```

```
(config-if)> no debug
```

7.8.15 interface ip mru

Sets the value of MRU (Maximum Receive Unit) to be transmitted to a remote node during establishing the the PPP (IPCP) connection. Prefix **no** cancels the command. If the MRU value is not specified, the default value of 1460 is used.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	PPP

```
(config-if)> ip mru <mr>
```

```
(config-if)> no ip mru
```

Argument	Type	Description
<i>mr</i>	Integer	MRU value

Wireless network 802.11

8.1 Access points

Base stations for connecting clients to a wired network via a radio channel are described in Keenetic as wireless access points (**AccessPoint**) interfaces. This functionality requires that the router have some relevant hardware — wireless adapter. Every physical adapter capable of accepting client connections has a corresponding interface **WifiMaster** in Keenetic, that has as many child interfaces **AccessPoint**, as wireless networks that it can serve simultaneously.

WifiMaster interface belongs to a class **Radio**, that is that is has a physical layer settings: broadcast frequency, transmitter power, and so on. Interfaces **AccessPoint** aggregate **Ethernet** and **SSID** classes functions. That is, all access points of a single adapter will have the same "physical" settings, but may different network name (SSID) and security algorithms. Access points interfaces are routable.

8.2 Wireless stations

Client connection to the network of another wireless access point corresponds to the **WifiStation** interface. For wireless adapters that support the station mode, the interfaces of this type get created automatically in Keenetic. Depending on the capabilities of the adapter hardware, there can be several **WifiStation** interfaces as well as wfunctioning in the wireless access point mode and a station, simultaneously. If a given adapter does not support two-mode functioning simultaneously, a relevant warning will be issued on the attempt to turn on another interface.

Interfaces **WifiStation** aggregate functions of **Ethernet**, **Radio** and **SSID** classes. Interfaces of this type are also routable.

8.3 Command description

8.3.1 interface ssid

Specifies the wireless network name (SSID) for interfaces "wireless station" (**WifiStation**) and "access point" (**AccessPoint**). Depending on the type of interface, SSID value is processed differently.

- For the access point SSID there needs to be a setup without which the it would not accept incoming connections.
- For station, SSID defines to which access point it will connect. Without a specified SSID, the station can connect to any available wireless network at its own discretion.

Prefix **no** deletes the configuration.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	SSID

```
(config-if)> ssid <ssid>
```

```
(config-if)> no ssid
```

Argument	Type	Description
<i>ssid</i>	String	Wireless Network Name (SSID)

8.3.2 interface channel

Sets the radio channel (broadcasting frequency band) for wireless interfaces. Wi-Fi interfaces take integers from 1 to 14 (frequency range from 2.412 GHz to 2.484 GHz) and from 36 to 165 (frequency range from 5.180 GHz to 5.825 GHz) as channel numbers.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no
Interface Type	Radio

```
(config-if)> channel <channel>
```

Argument	Type	Description
<i>channel</i>	Integer	Number of radio channel

8.3.3 interface compatibility

Sets the standard for wireless communications, with which a given wireless adapter (the interface) must be compatible. For Wi-Fi interfaces, the compatibility is set by string of Latin letters A, B, G, N, that denote extensions to the standard IEEE 802.11. For example, the presence 'A' in the compability line will imply that the given adapter will be able to deal with the 802.11a-compatible devices via radio channel. The set of admissible compatibility lines is defined by the hardware capabilities of a particular adapter and provisions of the relevant additions to the IEEE 802.11 standard.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no
Interface Type	Radio

```
(config-if)> compatibility <compatibility>
```

Argument	Type	Description
<i>compatibility</i>	String	The list of letter codes A, B, G, N.

8.3.4 interface power

Sets the transmitter power for the radio interface. Transmitter power is limited by the hardware capabilities and state laws applicable to radio broadcast. This command allows one to only reduce the power of the transmitter relative to its maximum power, such as to decrease potential interference with other devices in this range/band.

Properties	
Prefix no	no
Changes the settings	yes
Multiple entry	no
Interface Type	Radio

```
(config-if)> power <power>
```

Argument	Type	Description
<i>power</i>	Integer	The transmitter power as the percentage of the maximum power (from 1 to 100).

8.3.5 interface authentication wpa-psk

Specifies the pre-agreed key for authentication via WPA-PSK protocol. It is possible to specify the key as a 256-bit hexadecimal number or as a string of ASCII-characters. In the second case, the string is used as a code phrase to generate the key (*passphrase*).

Command with **no** prefix turns the setting off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	SSID

```
(config-if)> authentication wpa-psk <psk>
```

```
(config-if)> no authentication wpa-psk
```

Argument	Type	Description
<i>psk</i>	String	Pre-agreed key in the form of a 256-bit hexadecimal number, which consists of 64 hexadecimal digits, or in the form of ASCII string of 8 to 63 characters length.

8.3.6 interface authentication shared

Turns authentication with a shared key on. This mode is used only in conjunction with WEP encryption. Shared keys are specified by [interface encryption key](#) command.

Prefix **no** cancels the command, i. e. turns authentication to open mode.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	SSID

```
(config-if)> authentication shared
```

```
(config-if)> no authentication shared
```

8.3.7 interface encryption enable

Turns encryption on the wireless interface on. By default, WEP encryption is used. [wpa](#) and [wpa2](#) commands allow one to turn a stronger encryption algorithm on.

Calling the command with the **no** prefix turns wireless interface encryption off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	SSID

```
(config-if)> encryption enable
```

```
(config-if)> no encryption enable
```

8.3.8 interface encryption key

Specifies WEP encryption keys. Depending on the bit, the key can be set of 10 hexadecimal digits (5 characters ASCII) — 40-bit key, or 26 hexadecimal digits (13 characters ASCII) — 104-bit key. Overall, there can be 1 to 4 encryption keys, with one of them to be assigned to be the default key.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Interface Type	SSID

```
(config-if)> encryption key <id> (<hex> [default] | default)
```

```
(config-if)> no encryption key <id>
```

Argument	Type	Description
<i>id</i>	Integer	The key number. Overall, up to 4 keys could be specified.
<i>hex</i>	String	The key value as a hexadecimal number, consisting of 10 or 26 digits.
default	Keyword	Indicates that this key will be used by default.

8.3.9 interface encryption wpa

Turns on WPA security algorithms on the wireless interface. Wireless interface can support the joint use of WPA and WPA2, but supporting WEP automatically turns off when any of the WPA is turned on.

Calling with the **no** prefix turns WPA off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	SSID

```
(config-if)> encryption wpa
```

```
(config-if)> no encryption wpa
```

8.3.10 interface encryption wpa2

Enables WPA2 (IEEE 802.11i, RSN) security algorithms on the wireless interface. Wireless interface can support the joint use of WPA and WPA2, but supporting WEP automatically turns off when any of the WPA is turned on.

Calling with the **no** prefix turns WPA2 off.

Properties	
Prefix no	yes
Changes the settings	yes

Properties	
Multiple entry	no
Interface Type	SSID

```
(config-if)> encryption wpa2
```

```
(config-if)> no encryption wpa2
```

Configuring DNS

DNS-server names is configured by **ip name-server** command. They can also be obtained dynamically upon connecting via PPP or DHCP.

DNS-servers addresses are needed to resolve host names to IP-address when you work in DNS-proxy mode, as well as the interaction of the router Keenetic with the remote hosts specified by name.

9.1 Command description

9.1.1 ip name-server

Configuring DNS server IP-addresses. Addresses saved in this fashion are called *static* as opposite to *dynamic* — as registered by PPP or DHCP services.

Active, that addressed being used are the ones that have been registered most recently as compared to the others. Usually, the system uses the addresses which were obtained by several recent successfully connected PPP or DHCP services. If none of the services registers DNS addresses, static settings will be active. However, if after registering dynamic addresses the static settings are changed by the user, they become active until the new dynamic addresses are registered.

ip name-server command can be entered multiple times if several DNS-server addresses need to be setup. Moreover, each entered address can be associated with one or more domain names for working with specific areas, such as local names in the corporate network.

Prefix **no** removes the specified DNS server address from the static and the active lists if the command is furnished with arguments, or clears the list of static addresses if the command has no arguments.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip name-server <address> [domain]
```

```
(config)> no ip name-server [ <address> [domain] ]
```

Argument	Type	Description
<i>address</i>	IP-address	Name server address.

Argument	Type	Description
<i>domain</i>	The domain name	Domain for which the server will be used. In resolving names the DNS-proxy first selects the address of the server with name best matching the requested domain. If the domain is not specified, the server will be used for all requests.

9.1.2 service dns-proxy

DNS-proxy server starting command. Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service dns-proxy
```

```
(config)> no service dns-proxy
```

10.1 DHCP server

On the router Keenetic one can run *the DHCP server* which can provide the addresses from the specified pools on one or more network interfaces. The server is turned on by command **service dhcp**. The pools are configured using command **ip dhcp pool**.

It is possible to statically link IP-addresses to hosts' MAC-addresses of hosts using command **ip dhcp host**.

10.2 DHCP relay

If Keenetic is a part of a large network in which the issuance of addresses via DHCP is managed centrally, it may be necessary to forward DHCP-requests from the LAN to the corporate server. Such functionality is called *DHCP relay* and is turned on by command **service dhcp-relay**.

For relay to function, it is necessary to assign the roles to network interfaces using commands **ip dhcp relay lan** and **ip dhcp relay wan**. The system must have at least one interface of each type, then the relay will forward the requests from the network "lan" to "wan". One can also specify DHCP server address using command **ip dhcp relay server**.

Note: If the server address is not specified, the relay will broadcast to the "wan" network. Command **ip dhcp relay server** allows one to reduce the network load, because the requests are sent to a specific address.

10.3 Commands description

10.3.1 ip dhcp pool

Enter the group of commands for configuring DHCP-pool. If the pool is not found, the command will try to create it. Prefix **no** removes the pool. For a pool one sets a list of DNS-servers (**dns-server** command), *default gateway* (**default-router** command) and the lease time (**lease** command), as well as a range of dynamic IP-addresses (**range** command).

Note: In the current version of the system no more than one pool per interface is supported. For DHCP-server to function correctly it is required that the range of IP-addresses set by **range** command belong to the network that is configured on one of the device's Ethernet-interfaces.

Having configured the pool, it is necessary to turn the DHCP service on using the command **service dhcp**.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Entrance to a group	(config-dhcp)

```
(config)> ip dhcp pool <name>
```

```
(config)> no ip dhcp pool <name>
```

Argument	Type	Description
<i>name</i>	String	DHCP pool name

10.3.2 ip dhcp pool range

Configuring the range of dynamic addresses issued to DHCP-clients of a subnet. The range is set by start and end IP-addresses or the start address and size. The network interface to which the settings are applied is chosen automatically. Address of the chosen interface is used as the [default gateway](#) and DNS-server, if other addresses are not specified using commands [ip dhcp pool default-router](#) and [ip dhcp pool dns-server](#).

Prefix **no** removes the range.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-dhcp)> range <begin> (<end> | <size>)
```

```
(config-dhcp)> no range
```

Argument	Type	Description
<i>begin</i>	IP-address	Pool's start address
<i>end</i>	IP-address	Pool's end address
<i>size</i>	Integer	Pool size

10.3.3 ip dhcp pool default-router

Configuring [default gateway](#) IP-address. If not specified, the address of the Ethernet-interface determined automatically for a given range [range](#) will be used.

Prefix **no** cancels the setting.

Properties	
Prefix no	yes

Properties	
Changes the settings	yes
Multiple entry	no

```
(config-dhcp)> default-router <address>
```

```
(config-dhcp)> no default-router
```

Argument	Type	Description
<i>address</i>	IP-address	default gateway address.

10.3.4 ip dhcp pool dns-server

Configuring of IP-addresses of the DNS servers. If not specified, the address of the Ethernet-interface determined automatically for a given range [range](#) will be used.

Prefix **no** cancels the setting.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-dhcp)> dns-server <address1> [address2]
```

```
(config-dhcp)> no dns-server
```

Argument	Type	Description
<i>address1</i>	IP-address	Address of primary DNS-server
<i>address2</i>	IP-address	Address of secondary DNS-server

10.3.5 ip dhcp pool lease

Configuring the lease time of DHCP pool IP-address. Prefix **no** sets the default value, equal to 86400 seconds.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-dhcp)> lease <lease>
```

```
(config-dhcp)> no lease
```

Argument	Type	Description
<i>lease</i>	Integer	Lease time in seconds.

10.3.6 ip dhcp host

Configuring static linking of IP-address to MAC-address of the host. If the host with the specified name is not found, the command will try to create it. Prefix **no** removes the host. If the specified IP-address is not in range of any pool, the command will remain in the settings, but will not affect the DHCP-server functioning.

The command allows one to change the MAC-address, leaving the old value IP-address and vice versa — to change the IP-address, leaving the old MAC-address value intact.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip dhcp host <name> [mac] [ip]
```

```
(config)> no ip dhcp host <name>
```

Argument	Type	Description
<i>name</i>	String	arbitrary host name, used to identify a MAC-IP pair in the settings
<i>mac</i>	MAC-address	MAC-address of the host for static linking of IP-address. If not specified, the value is taken from the previous configuration.
<i>ip</i>		IP-address of the host. If not specified, the value is taken from the previous configuration.

10.3.7 service dhcp

Command to start DHCP-server. If there is not enough settings to start the service (see [ip dhcp pool](#)), the service will not respond to the network. As soon as there are enough settings, the service will turn on automatically.

Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service dhcp
```

```
(config)> no service dhcp
```

10.3.8 ip dhcp relay lan

Specifies which network interface the DHCP relay will use to handle client's requests. Several "lan" interfaces can be specified, to which end the command should be entered several times, enumerating all desired interfaces one by one.

Prefix **no** turns off the DHCP relay on the specified interface or on all interfaces, if the command is entered without arguments.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config)> ip dhcp relay lan <interface>
```

```
(config)> no ip dhcp relay lan [interface]
```

Argument	Type	Description
<i>interface</i>	Interface Name	Full name or an alias of Ethernet interface, through which DHCP relay will accept requests from clients.

10.3.9 ip dhcp relay wan

Specifies the network interface through which DHCP relay will interact with higher level DHCP server. There can be only one interface of such type in the in the system. If exact address of the server is not specified (see [ip dhcp relay server](#)), the requests will be broadcasted. It is recommended to specify server address.

Prefix **no** turns the setting off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> ip dhcp relay wan <interface>
```

```
(config)> no ip dhcp relay wan [interface]
```

Argument	Type	Description
<i>interface</i>		Full name or an alias of Ethernet interface, on which requests from the DHCP-clients will be sent.

10.3.10 ip dhcp relay server

Specifies the IP-address of the DHCP server, to which the relay will forward client requests from the LAN.

Prefix **no** turns the setting off.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> ip dhcp relay server <address>
```

```
(config)> no ip dhcp relay server [address]
```

Argument	Type	Description
<i>address</i>	IP-address	IP-address of the DHCP server.

10.3.11 service dhcp-relay

The command of starting DHCP-relay. If there are not enough settings to start the service, it will not respond within the network. As soon as there are enough settings, the service starts up automatically.

Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service dhcp-relay
```

```
(config)> no service dhcp-relay
```

IGMP

Router Keenetic can function in IGMP-Relay mode (proxy). To attain this goal, one needs to setup one network interface to receive IGMP-requests from the multicast receivers (downstream mode), and another interface — to the upstream mode for redirecting the received requests to the higher level network.

Moreover, sometimes it is necessary to mirror the IGMP packets going upstream into another interface. The working mode of such interface is called fork. Copies of IGMP-packets going to fork can be used by some IP-TV service providers in tuning the bandwidth.

11.1 Command description

11.1.1 interface igmp upstream

Turns IGMP mode on at the interface in the direction of the multicast source. The device must be running [service igmp-proxy](#) service. Only one upstream interface is allowed.

Prefix **no** cancels the command.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	IP

```
(config-if)> igmp upstream
```

```
(config-if)> no igmp upstream
```

11.1.2 interface igmp downstream

Turns IGMP mode on at the interface in the direction of the multicast recipients. The device must be running [service igmp-proxy](#) service. There can be several downstream interfaces.

Prefix **no** cancels the command.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

Properties	
Interface Type	IP

```
(config-if)> igmp downstream
```

```
(config-if)> no igmp downstream
```

11.1.3 interface igmp fork

Turns duplication of outgoing packets upstream IGMP into the specified interface. There can be only one fork interface.

Prefix **no** cancels the command.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no
Interface Type	IP

```
(config-if)> igmp fork
```

```
(config-if)> no igmp fork
```

11.1.4 service igmp-proxy

IGMP-proxy startup command. For service to function it is necessary to have one upstream interface and at least one downstream interface. If there are not enough settings to run the service, the service will not function. As soon as there are enough settings, the service will start automatically.

Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service igmp-proxy
```

```
(config)> no service igmp-proxy
```

Device access control

Control of settings Keenetic is executed via the command line (telnet), web-interface (HTTP) and FTP. On entry one will be prompted for the administrator password, which can be changed by using **user password** command. To control the device, there must be relevant services running and the user must have the access rights (see **user tag**)

By default Keenetic allows one to connect to the configuration services via network interfaces which have private security level set (see **interface security-level**) and does not allow one to connect to interfaces with public security level. Moreover, access can be controlled using additional allowing and denying rules (see commands **access-list** and **interface ip access-group**).

12.1 Command description

12.1.1 user

Entry to a group of settings of user account parameters. If the specified user does not exist, the command attempts to create one.

Prefix **no** removes the user.

Note: Account with reserved name admin can not be removed. In addition, the admin user can not lose the access right to command line.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes
Entrance to a group	(config-user)

```
(config)> user <name>
```

```
(config)> no user <name>
```

Argument	Type	Description
<i>name</i>	String	User name.

12.1.2 user password

Sets the user password. The password is stored as MD5-hash, computed from the line "user:ndm:password".

The command takes open string or hash-function value as argument. Saved password is used for user authentication.

Prefix **no** resets the password so that the user loses access to the device. For the admin user prefix **no** resets the password to the factory settings — 1234.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config-user)> password (md5 <hash> | <password>)
```

```
(config-user)> no password
```

Argument	Type	Description
<i>hash</i>	String	MD5-hash value.
<i>password</i>	String	Value of the password in open form, from which the hash value is calculated automatically.

12.1.3 user tag

Assigns a special tag to the account, which presence is checked at the time of user authorization as well as performing any action in the system. Set of permitted tag values depends on the system functionality. The full list is shown in the table below.

Several different tags can be assigned to one account by entering the command several times. Each tag can be viewed as granting or revoking certain permissions.

Entering the command with **no** prefix deletes the specified tag.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	yes

```
(config-user)> tag <tag>
```

```
(config-user)> no tag <tag>
```

Argument	Type	Description
<i>tag</i>	Tag	Tag which presence is required for user to perform certain actions.

Table 12.1. List of access permissions tags

Tag	Description
cli	Access to command line interface .

Tag	Description
http	Access to the Web-interface.
ftp	Connection to an integrated FTP-server.
cifs	Connection to the Windows files and printers service.
torrent	Entry to the BitTorrent file sharing client GUI.
readonly	Restrict commands that change the settings.

Note: admin account cannot be tagged readonly or untagged cli.

12.1.4 service http

Command of starting the HTTP-server that provides the user with Web-interface to configure the device. Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service http
```

```
(config)> no service http
```

12.1.5 service ftp

Command of starting the FTP-server that provides the user with access to connected USB-drives, configuration files and a file with firmware update. Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes
Multiple entry	no

```
(config)> service ftp
```

```
(config)> no service ftp
```

12.1.6 service telnet

Command of starting the telnet server that provides the user with command line interface to configure the device. Prefix **no** stops the service.

Properties	
Prefix no	yes
Changes the settings	yes

Properties	
Multiple entry	no

```
(config)>  service telnet
```

```
(config)> no service telnet
```

Diagnostics

13.1 Command description

13.1.1 show system

Displays the general state of the system:

```
(config)> show system

hostname: Undefined
domainname: WORKGROUP
cpuload: 0 ❶
memory: 13984/28976 ❷
swap: 0/0 ❸
uptime: 153787 ❹
```

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show system
```

System state general info

- ❶ CPU load, percentage.
- ❷ Occupied and available memory info, kilobytes.
- ❸ Swap file usage info, kilobytes.
- ❹ System uptime from the start, seconds.

13.1.2 show interface

Group of commands to display current settings of interfaces of various types. Each command of this group is applicable to a particular type of interface hierarchy.

Entering the command **show interface** itself, without specifying child commands will display a general table for all network interfaces or specific interface *name* information.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show interface <name>
```

Argument	Type	Description
<i>name</i>	Interface Name	Full name or an alias of the interface which information is to be displayed.

Example 13.1. Review the status of switch ports

The command **show interface** displays different information depending on the interface type. In particular, for **Switch** switch it shows current state of physical ports, speed and duplex, on top of general information.

```
config)> show interface Switch0
```

```

    index: 0
    type: Switch
  description:
    state: up
    link: up
    port, index = 1:
      link: up
      speed: 100M
      duplex: full
    port, index = 2:
      link: down
      speed:
      duplex:
    port, index = 3:
      link: down
      speed:
      duplex:
    port, index = 4:
      link: down
      speed:
      duplex:
    port, index = 5:
      link: up
      speed: 100M
      duplex: full
```

13.1.3 show interface mac

Displays the table of MAC-addresses of the switch.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show interface <name>mac
```

13.1.4 show ip route

Displays the current routing table.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show ip route
```

13.1.5 show ip arp

Displays the contents of the ARP cache.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show ip arp
```

13.1.6 show ip name-server

Displays a list of current addresses of DNS-servers in order of decreasing priority.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show ip name-server
```

13.1.7 show log

Displays system log contents (records that are present in a circular buffer), as well as new records as they come. The command executes in the background, that is, until forced to stop by the user pressing [Ctrl]+[C].

Properties	
Prefix no	no
Changes the settings	no

```
(config)> show log
```

13.1.8 tools ping

Sends Echo-Request requests of ICMP protocol to specified network node and registers received Echo-Reply responses. The time between sending request and receiving the response Round Trip Time (RTT) allows you to define double ended delays on the route and frequency of packet losses, that is, indirectly determine loading on the channels of data transmission and intermediate devices.

Total absence of ICMP-replies can also mean that the remote node (or any of the intermediate routers) blocks ICMP Echo-Reply or ignores ICMP Echo-Request.

Properties	
Prefix no	no
Changes the settings	no

```
(config)> tools ping <host> [count <count>] [size <packetsize>]
```

Argument	Type	Description
<i>host</i>	String	Domain name or host IP-address.
<i>count</i>	Integer	Number of ICMP Echo requests. If not specified, the command will run until interrupted by the user.
<i>packetsize</i>	Integer	Size of the ICMP Echo-Request data field in bytes. By default — 56, which together with the 8-byte header specifies the size of the ICMP-pack — 64 bytes.

Examples of settings

14.1 Testing of throughput

Two subnets 192.168.1.0/24 and 192.168.2.0/24 are directly connected to the router. The first network is connected through any port from 1 to 4, and the second one — through the port 5. Address translation is disabled, the default route is missing. This configuration is used to test the router throughput when packets are transmitted from one network to another.

Example 14.1. The simplest configuration without NAT

```
system
    set net.ipv4.ip_forward 1
!
interface Switch0
    port 1
        mode access
        access vlan 1
    !
    port 2
        mode access
        access vlan 1
    !
    port 3
        mode access
        access vlan 1
    !
    port 4
        mode access
        access vlan 1
    !
    port 5
        mode access
        access vlan 2
    !
    up
!
interface Switch0/VLAN1
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
!
interface Switch0/VLAN2
    security-level private
    ip address 192.168.2.1 255.255.255.0
    up
```

```
!
service telnet
```

14.2 Routing with NAT enabled

The easiest way to connect to the Internet. At the interface Switch0/VLAN2 there configured a global IP-address, default route through a gateway 203.0.113.241 is assigned and DNS-servers addresses are set. All hosts connected to the network Switch0/VLAN1, connect to the Internet through NAT and DNS-proxy.

Note: Interface Switch0/VLAN1 has security level `private`, and the interface Switch0/VLAN2 — `public`. Here is how the security policy gets set: built-in firewall allows one to initiate connections only from VLAN 1 to VLAN 2. Access from VLAN 2 to hosts of VLAN 1 and to the router is denied. Then, use the commands `access-list` and `interface ip access-group` to specify exceptions from the general rule.

Note: Interface Switch0/VLAN2 has an option `global`, without which address translation will not work.

Example 14.2. Static settings and NAT

```
system
  set net.ipv4.ip_forward 1
  set net.ipv4.netfilter.ip_conntrack_max 4096
  set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
  set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
  set net.ipv4.tcp_fin_timeout 30
  set net.ipv4.tcp_keepalive_time 120
!
interface Switch0
  port 1
    mode access
    access vlan 1
  !
  port 2
    mode access
    access vlan 1
  !
  port 3
    mode access
    access vlan 1
  !
  port 4
    mode access
    access vlan 1
  !
  port 5
    mode access
    access vlan 2
  !
up
!
```

```

interface Switch0/VLAN1
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
!
interface Switch0/VLAN2
    security-level public
    ip address 203.0.113.242 255.255.255.240
    ip global 1
    up
!
ip route default 203.0.113.241
ip name-server 8.8.8.8
ip name-server 8.8.4.4
ip nat Switch0/VLAN1
service telnet

```

14.3 DHCP-server and DHCP-client

Example with NAT is supplemented by obtaining a dynamic IP-address via DHCP on WAN interface. To this end DHCP-client is enabled using command **interface ip dhcp**.

Additionally, address from the range 172.16.1.33–172.16.1.53 are assigned to the network 172.16.1.0/24 via DHCP. To this end, DHCP-server is enabled using command **service dhcp** and pool _WEBADMIN is configured.

In the setting of nowhere explicitly specifies that the pool _WEBADMIN will work with Switch0/VLAN1 interface. Linking occurs automatically when the address range of the pool hits the network 172.16.1.0/24, configured on the interface.

Example 14.3. DHCP-server and DHCP-client

```

system
    set net.ipv4.ip_forward 1
    set net.ipv4.netfilter.ip_conntrack_max 4096
    set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
!
interface Switch0
    port 1
        mode access
        access vlan 1
    !
    port 2
        mode access
        access vlan 1
    !
    port 3
        mode access
        access vlan 1
    !

```

```

    port 4
        mode access
        access vlan 1
    !
    port 5
        mode access
        access vlan 2
    !
    up
!
interface Switch0/VLAN1
    description "LAN interface, DHCP server"
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
!
interface Switch0/VLAN2
    description "WAN interface, DHCP client"
    ip dhcp
    security-level public
    ip global 1
    up
!
ip dhcp pool _WEBADMIN
    range 172.16.1.33 20
!
ip nat Switch0/VLAN1
service telnet
service dhcp
service dns-proxy

```

14.4 Wi-Fi access point in bridge mode

Let us supplement [the example with the DHCP-client and server](#) by adding a wireless access point.

Suppose it is required to make a wireless connection to a local network (LAN), so that the hosts of the wireless and the wired parts of LAN “see” each other directly in one segment of the Ethernet. For this purpose one creates the Bridge0 interface, to which the interfaces Switch0/VLAN1 and WifiMaster0/AccessPoint0 are hooked up. Now the wired and the wireless networks are connected through the bridge. IP-address 172.16.1.1 is inherited by the bridge Bridge0 from the wired interface Switch0/VLAN1.

Example 14.4. Wi-Fi access point

```

system
    set net.ipv4.ip_forward 1
    set net.ipv4.netfilter.ip_conntrack_max 4096
    set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
!

```

```

interface Switch0
    port 1
        mode access
        access vlan 1
    !
    port 2
        mode access
        access vlan 1
    !
    port 3
        mode access
        access vlan 1
    !
    port 4
        mode access
        access vlan 1
    !
    port 5
        mode access
        access vlan 2
    !
    up
!
interface Switch0/VLAN1
    description "LAN interface, DHCP server"
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
!
interface Switch0/VLAN2
    description "WAN interface, DHCP client"
    ip dhcp
    security-level public
    ip global 1
    up
!
interface WifiMaster0
    country-code RU
    compatibility BGN
    up
!
interface WifiMaster0/AccessPoint0
    description "LAN, wireless Access Point"
    security-level private
    ssid "<productname/>"
    up
!
interface Bridge0
    description "LAN, IP address inherited from Switch0/VLAN1"
    security-level private
    include WifiMaster0/AccessPoint0
    inherit Switch0/VLAN1
!
ip dhcp pool _WEBADMIN

```

```

    range 172.16.1.33 20
!
ip nat Bridge0
service telnet
service dhcp
service dns-proxy

```

14.5 PPP connection

Example 14.5. PPPoE connection

```

system
    set net.ipv4.ip_forward 1
    set net.ipv4.netfilter.ip_conntrack_max 4096
    set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
!
interface Switch0
    port 1
        mode access
        access vlan 1
    !
    port 2
        mode access
        access vlan 1
    !
    port 3
        mode access
        access vlan 1
    !
    port 4
        mode access
        access vlan 1
    !
    port 5
        mode access
        access vlan 2
    !
    up
!
interface Switch0/VLAN1
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
!
interface Switch0/VLAN2
    security-level public
    up
!
interface PPPoE0

```

```

description "Internet connection"
lcp echo 30 3
ipcp default-route
ipcp name-servers
authentication identity andreyd
authentication password amd031181
authentication chap
ip tcp adjust-mss pmtu
ip global 1
connect via Switch/VLAN2
up
!
ip nat Switch0/VLAN1
service telnet

```

Example 14.6. PPTP connection

```

system
set net.ipv4.ip_forward 1
set net.ipv4.netfilter.ip_conntrack_max 4096
set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
set net.ipv4.tcp_fin_timeout 30
set net.ipv4.tcp_keepalive_time 120
!
interface Switch0
port 1
mode access
access vlan 1
!
port 2
mode access
access vlan 1
!
port 3
mode access
access vlan 1
!
port 4
mode access
access vlan 1
!
port 5
mode access
access vlan 2
!
up
!
interface Switch0/VLAN1
description "LAN interface, DHCP server"
security-level private
ip address 192.168.1.1 255.255.255.0
up
!

```

```

interface Switch0/VLAN2
    description "WAN interface, DHCP client"
    ip dhcp
    security-level public
    ip global 1
    up
!
interface PPTP0
    description "Internet connection"
    peer vpn.example.net
    lcp echo 30 3
    ipcp default-route
    ipcp name-servers
    authentication identity sergeymv
    authentication password smv050859
    authentication mschap-v2
    authentication chap
    connect via Switch0/VLAN2
    ip global 2
    up
!
ip dhcp pool _WEBADMIN
    range 172.16.1.33 20
!
ip nat Switch0/VLAN1
service telnet
service dhcp
service dns-proxy

```

Example 14.7. L2TP connection

```

system
    set net.ipv4.ip_forward 1
    set net.ipv4.netfilter.ip_conntrack_max 4096
    set net.ipv4.netfilter.ip_conntrack_tcp_timeout_established 1200
    set net.ipv4.netfilter.ip_conntrack_udp_timeout 60
    set net.ipv4.tcp_fin_timeout 30
    set net.ipv4.tcp_keepalive_time 120
!
interface Switch0
    port 1
        mode access
        access vlan 1
    !
    port 2
        mode access
        access vlan 1
    !
    port 3
        mode access
        access vlan 1
    !
    port 4
        mode access

```

```
        access vlan 1
        !
        port 5
        mode access
        access vlan 2
        !
        up
    !
interface Switch0/VLAN1
    description "LAN interface, DHCP server"
    security-level private
    ip address 192.168.1.1 255.255.255.0
    up
    !
interface Switch0/VLAN2
    description "WAN interface, DHCP client"
    ip dhcp
    security-level public
    ip global 1
    up
    !
interface L2TP0
    description "Internet connection"
    peer tp.example.net
    lcp echo 30 3
    ipcp default-route
    ipcp name-servers
    authentication identity sergeymv
    authentication password smv050859
    authentication chap
    connect via Switch0/VLAN2
    ip global 2
    up
    !
ip dhcp pool _WEBADMIN
    range 172.16.1.33 20
    !
ip nat Switch0/VLAN1
service telnet
service dhcp
service dns-proxy
```


Access Control List	is a table that tells a system which access rights each user has to a particular system object, such as a file directory or individual file. Each object has a security attribute that identifies its access control list. The list has an entry for each system user with access privileges. The most common privileges include the ability to read a file (or all the files in a directory), to write to the file or files, and to execute the file (if it is an executable file, or program).
Command Line Interface	is a user interface to a computer's operating system or an application in which the user responds to a visual prompt by typing in a command on a specified line, receives a response back from the system, and then enters another command, and so forth.
Common Internet File System	is a protocol that lets programs make requests for files and services on remote computers on the Internet. CIFS uses the client/server programming model. A client program makes a request of a server program (usually in another computer) for access to a file or to pass a message to a program that runs in the server computer. The server takes the requested action and returns a response.
Default gateway	is the node on the computer network that the network software uses when an IP address does not match any other routes in the routing table.
DHCP server	<p>The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, the name servers, other servers such as time servers, and so forth. On receiving a valid request, the server assigns the computer an IP address, a lease (length of time the allocation is valid), and other IP configuration parameters, such as the subnet mask and the default gateway. Depending on implementation, the DHCP server may have three methods of allocating IP-addresses:</p> <ul style="list-style-type: none"> • <i>dynamic allocation</i>: A network administrator assigns a range of IP addresses to DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. The request-and-grant process uses a lease concept with a controllable time period, allowing the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed. • <i>automatic allocation</i>: The DHCP server permanently a free IP address to a requesting client from the range defined by the administrator. This is like dynamic allocation, but the DHCP server keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.

- *static allocation*: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in (perhaps by a network administrator). Only requesting clients with a MAC address listed in this table will be allocated an IP address. This feature (which is not supported by all DHCP servers) is variously called Static DHCP Assignment (by DD-WRT), fixed-address (by the dhcpd documentation), Address Reservation (by Netgear), DHCP reservation or Static DHCP (by Cisco/Linksys), and IP reservation or MAC/IP binding (by various other router manufacturers).

Idempotence

is the property of certain operations in mathematics and computer science, that they can be applied multiple times without changing the result beyond the initial application. The concept of idempotence arises in a number of places in abstract algebra (in particular, in the theory of projectors and closure operators) and functional programming (in which it is connected to the property of referential transparency).

Network interface

is the point of interconnection between a computer and a private or public network. A network interface is generally a network interface card (NIC), but does not have to have a physical form. Instead, the network interface can be implemented in software.

Router

is a device that forwards data packets between computer networks, creating an overlay internetwork. A router is connected to two or more data lines from different networks. When a data packet comes in on one of the lines, the router reads the address information in the packet to determine its ultimate destination. Then, using information in its routing table or routing policy, it directs the packet to the next network on its journey. Routers perform the "traffic directing" functions on the Internet. A data packet is typically forwarded from one router to another through the networks that constitute the internetwork until it gets to its destination node.

Routing table

or Routing Information Base (RIB), is a data table stored in a router or a networked computer that lists the routes to particular network destinations, and in some cases, metrics (distances) associated with those routes. The routing table contains information about the topology of the network immediately around it. The construction of routing tables is the primary goal of routing protocols. Static routes are entries made in a routing table by non-automatic means and which are fixed rather than being the result of some network topology "discovery" procedure.