**Manual:Interface/PPPoE**

**Summary**

The PPPoE (Point to Point Protocol over Ethernet) protocol provides extensive user management, network management and accounting benefits to ISPs and network administrators. Currently PPPoE is used mainly by ISPs to control client connections for xDSL and cable modems as well as plain Ethernet networks. PPPoE is an extension of the standard Point to Point Protocol (PPP). The difference between them is expressed in transport method: PPPoE employs Ethernet instead of serial modem connection.

Generally speaking, PPPoE is used to hand out IP addresses to clients based on the username (and workstation, if desired) authentication as opposed to workstation only authentication, when static IP addresses or DHCP are used. It is advised not to use static IP addresses or DHCP on the same interfaces as PPPoE for obvious security reasons.

The PPPoE client and server work over any Ethernet level interface on the router - wireless 802.11 (Aironet, Cisco, WaveLan, Prism, Atheros), 10/100/1000 Mbit/s Ethernet, RadioLan and EoIP (Ethernet over IP tunnel).

**Feature list**

- PPPoE server and client support;
- Multilink PPP (MLPPP);
- MLPPP over single link (ability to transmit full-sized frames);
- BCP (Bridge Control Protocol) support - allows to send raw Ethernet frames over PPP links;
- MPPE 40bit and MPPE 128bit RSA encryption;
- pap, chap, mschap v1/v2 authentication;
- RADIUS support for client authentication and accounting.

Note that when RADIUS server is authenticating a user with CHAP, MS-CHAPv1 or MS-CHAPv2, the RADIUS protocol does not use shared secret, it is used only in authentication reply. So if you have a wrong shared secret, RADIUS server will accept the request. You can use /radius monitor command to see bad-replies parameter. This value should increase whenever a client tries to connect.

Supported connections:

- MikroTik RouterOS PPPoE client to any PPPoE server (access concentrator)
- MikroTik RouterOS server (access concentrator) to multiple PPPoE clients (clients are available for almost all operating systems and most routers)
Specifications

- Packages required: ppp
- License required: Level1 (limited to 1 interface), Level3 (limited to 200 interfaces), Level4 (limited to 200 interfaces), Level5 (limited to 500 interfaces), Level6 (unlimited)
- Submenu level: /interface pppoe-server, /interface pppoe-client
- Standards and Technologies: PPPoE (RFC 2516)
- Hardware usage: PPPoE server may require additional RAM (uses approx. 9KiB (plus extra 10KiB for packet queue, if data rate limitation is used) for each connection) and CPU power. Maximum of 65535 connections is supported.

Quick Setup Guide

To configure MikroTik RouterOS to be a PPPoE client, just add a pppoe-client:

```
/interface pppoe-client
  add name=pppoe-user-mike user=user password=passwd interface=wlan1 \
  service-name=internet disabled=no
```

To configure MikroTik RouterOS to be an Access Concentrator (PPPoE Server):

- add an address pool for the clients from 10.1.1.62 to 10.1.1.72;
- add ppp profile;
- add ppp secret (username/password);
- add pppoe server itself.

```
/ip pool
add name="pppoe-pool" ranges=10.1.1.62-10.1.1.72

/ppp profile
add name="pppoe-profile" local-address=10.1.1.1 remote-address=pppoe-pool

/ppp secret
add name=user password=passwd service=pppoe profile=pppoe-profile

/interface pppoe-server server
add service-name=internet interface=wlan1 default-profile=pppoe-profile
```
PPPoE Operation

Stages
PPPoE has two stages:

- **Discovery stage** - a client discovers all available access concentrators and selects one of them to establish PPPoE session. This stage has four steps: *initialization, offer, request and session confirmation*. PPPoE Discovery uses special Ethernet frames with their own Ethernet frame type 0x8863.

To initiate discovery, PPPoE client sends **PADI** frame to the broadcast Ethernet address (FF:FF:FF:FF:FF:FF) and may specify particular *service name*. When server receives **PADI** frame, it responds with **PADO** frame to Client's unicast Ethernet address. There can be more than one server in broadcast range of the client. In such case client collects PADO frames and picks one (in most cases it picks the server which responded first) to start session.

Client sends **PADR** frame to unicast Ethernet address of the server it chose. If server agrees to set up a session with this particular client, it allocates resources to set up PPP session and assigns *Session ID* number. This number is sent back to client in **PADS** frame. When client receives PADS frame, it knows server's mac address and *Session ID*, it allocates resources and session can begin.

- **Session** - When discovery stage is completed, both peers know **PPPoE Session ID** and other peer's *Ethernet (MAC) address* which together defines PPPoE session. PPP frames are encapsulated in PPPoE session frames, which have Ethernet frame type 0x8864.

When server sends confirmation and client receives it, **PPP Session stage** is started that consists of following steps:

  - **LCP negotiation**
  - **Authentication**
  - **IPCP negotiation** - client is assigned with an IP address.

PPPoE server sends Echo-Request packets to the client to determine the state of the session, otherwise server will not be able to determine that session is terminated in cases when client terminates session without sending
Terminate-Request packet.

More detailed description of PPPoE protocol can be found in RFC 2516

**Used Packet Types**

<table>
<thead>
<tr>
<th>Packet</th>
<th>Description</th>
</tr>
</thead>
</table>
| **PADI** | PPPoE Active Discovery Initialization  
The PPPoE client sends out a PADI packet to the broadcast address. This packet can also populate the "service-name" field if a service name has been entered on the dial-up networking properties of the PPPoE broadband connectoid. If a service name has not been entered, this field is not populated |
| **PADO** | PPPoE Active Discovery Offer  
The PPPoE server, or Access Concentrator, should respond to the PADI with a PADO if the Access Concentrator is able to service the "service-name" field that had been listed in the PADI packet. If no "service-name" field had been listed, the Access Concentrator will respond with a PADO packet that has the "service-name" field populated with the service names that the Access Concentrator can service. The PADO packet is sent to the unicast address of the PPPoE client |
| **PADR** | PPPoE Active Discovery Request  
When a PADO packet is received, the PPPoE client responds with a PADR packet. This packet is sent to the unicast address of the Access Concentrator. The client may receive multiple PADO packets, but the client responds to the first valid PADO that the client received. If the initial PADI packet had a blank "service-name" field filed, the client populates the "service-name" field of the PADR packet with the first service name that had been returned in the PADO packet |
| **PADS** | PPPoE Active Discovery Session confirmation  
When the PADR is received, the Access Concentrator generates a unique session identification (ID) for the Point-to-Point Protocol (PPP) session and returns this ID to the PPPoE client in the PADS packet. This packet is sent to the unicast address of the client |
| **PADT** | PPPoE Active Discovery Terminate  
might be sent anytime after a session is established to indicate that a PPPoE session terminated. It can be sent by either server or client |

### MTU

Typically largest Ethernet frame that can be transmitted without fragmentation is 1500 bytes. PPPoE adds another 6 bytes of overhead and PPP field adds two more bytes, leaving 1492 bytes for IP datagram. Therefore max PPPoE MRU and MTU values must not be larger than 1492.

TCP stacks try to avoid fragmentation, so they use an MSS (Maximum Segment Size). By default MSS is chosen as MTU of the outgoing interface minus the usual size of the TCP and IP headers (40 bytes), which results in 1460 bytes for an Ethernet interface. Unfortunately there may be intermediate links with lower MTU which will cause fragmentation. In such case TCP stack performs path MTU discovery. Routers which cannot forward the datagram without fragmentation are supposed to drop packet and send ICMP-Fragmentation-Required to originating host. When host receives such ICMP, it tries lower MTU. This should work in ideal world, however in real world many routers do not generate fragmentation-required datagrams, also many firewalls drop all ICMP datagrams. Workaround for this problem is to adjust MSS if it is too big. By default RouterOS adds mangle rules to intercept TCP SYN packets and silently adjust any advertised MSS option so they will be appropriate for the PPPoE link. Additional information on maximum supported MTUs for routerboards are listed here.
PPPoE Client

Sub-menu: /interface pppoe-client

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac-name (string)</td>
<td>Access Concentrator name, this may ne left blank and the client will connect to any access concentrator on the broadcast domain</td>
</tr>
<tr>
<td>add-default-route (yes/no; Default: no)</td>
<td>Enable/Disable whether to add default route automatically</td>
</tr>
<tr>
<td>allow (mschap2/mschap1/chap/pap; Default: )</td>
<td>allowed authentication methods, by default all methods are allowed</td>
</tr>
<tr>
<td>dial-on-demand (yes/no; Default: no)</td>
<td>connects to AC only when outbound traffic is generated</td>
</tr>
<tr>
<td>interface (string; Default: )</td>
<td>interface name on which client will run</td>
</tr>
<tr>
<td>max-mru (integer; Default: 1460)</td>
<td>Maximum Receive Unit</td>
</tr>
<tr>
<td>max-mtu (integer; Default: 1460)</td>
<td>Maximum Transmission Unit</td>
</tr>
<tr>
<td>mrru (integer: 512..65535</td>
<td>disabled; Default: disabled)</td>
</tr>
<tr>
<td>name (string; Default: pppoe-out[i])</td>
<td>name of the PPPoE interface, generated by ROuterOS if not specified</td>
</tr>
<tr>
<td>password (string; Default: )</td>
<td>password used to authenticate</td>
</tr>
<tr>
<td>profile (string; Default: )</td>
<td>default profile for the connection defined in /ppp profiles</td>
</tr>
<tr>
<td>service-name (string; Default: '')</td>
<td>specifies the service name set on the access concentrator, can be left blank to connect to any PPPoE server</td>
</tr>
<tr>
<td>use-peer-dns (yes/no; Default: no)</td>
<td>enable/disable getting DNS settings from the peer</td>
</tr>
<tr>
<td>user (string; Default: '')</td>
<td>username used for authentication</td>
</tr>
</tbody>
</table>

Status

Command /interface pppoe-client monitor will display current PPPoE status.

Available read only properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac-mac (MAC address)</td>
<td>MAC address of the access concentrator (AC) the client is connected to</td>
</tr>
<tr>
<td>ac-name (string)</td>
<td>name of the Access Concentrator</td>
</tr>
<tr>
<td>encoding (string)</td>
<td>encryption and encoding (if asymmetric, separated with '/') being used in this connection</td>
</tr>
<tr>
<td>mrru (integer)</td>
<td>effective MRU of the link</td>
</tr>
<tr>
<td>mtu (integer)</td>
<td>effective MTU of the link</td>
</tr>
<tr>
<td>service-name (string)</td>
<td>used service name</td>
</tr>
<tr>
<td>status (string)</td>
<td>current link status. Available values are:</td>
</tr>
<tr>
<td></td>
<td>• dialing,</td>
</tr>
<tr>
<td></td>
<td>• verifying password...,</td>
</tr>
<tr>
<td></td>
<td>• connected,</td>
</tr>
<tr>
<td></td>
<td>• disconnected,</td>
</tr>
<tr>
<td>uptime (time)</td>
<td>connection time displayed in days, hours, minutes and seconds</td>
</tr>
</tbody>
</table>
**Scanner**

Starting from v3.21 RouterOS has new tool - PPPoE Scanner. It allows you to scan all active PPPoE servers in broadcast domain. Command to run scanner is as follows:

```
/interface pppoe-client scan <interface>
```

Available read only properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>service</td>
<td>(string) Service name configured on server</td>
</tr>
<tr>
<td>mac-address</td>
<td>(MAC) Mac address of detected server</td>
</tr>
<tr>
<td>ac-name</td>
<td>(string) name of the Access Concentrator</td>
</tr>
</tbody>
</table>

**Notes**

Note for Windows. Some connection instructions may use the form where the "phone number", such as "MikroTik_ACmt1", is specified to indicate that "MikroTik_AC" is the access concentrator name and "mt1" is the service name.

Specifying MRRU means enabling MP (Multilink PPP) over single link. This protocol is used to split big packets into smaller ones. Under Windows it can be enabled in Networking tag, Settings button, "Negotiate multi-link for single link connections". Their MRRU is hardcoded to 1614. This setting is useful to overcome PathMTU discovery failures. The MP should be enabled on both peers.

**Example**

To add and enable PPPoE client on the ether1 interface connecting to the AC that provides testSN service using username with the password passwd:

```
[admin@RemoteOffice] interface pppoe-client> add interface=ether1 service-name=testSN user=user
password=passwd disabled=no
[admin@RemoteOffice] interface pppoe-client> print
Flags: X - disabled, R - running
  0  R  name="pppoe-out1" max-mtu=1480 max-mru=1480 mrru=disabled interface=ether1
     user="user" password="passwd" profile=default service-name="testSN"
     ac-name="" add-default-route=no dial-on-demand=no use-peer-dns=no
     allow=pap,chap,mschap1,mschap2

[admin@MikroTik] interface pppoe-client> monitor pppoe-out1
status: "connected"
  uptime: 6s
  idle-time: 6s
  encoding: "MPPE128 stateless"
  service-name: "testSN"
  ac-name: "MikroTik"
  ac-mac: 00:0C:42:04:00:73
     mtu: 1480
     mru: 1480
```
Additional Resources

PPPoE Clients:
- RASPPPoE\footnote{1} for Windows 95, 98, 98SE, ME, NT4, 2000, XP, .NET

**PPPoE Server Setup (Access Concentrator)**

**Sub-menu: /interface pppoe-server server**

The PPPoE server (access concentrator) supports multiple servers for each interface - with differing service names. Currently the throughput of the PPPoE server has been tested to 160 Mb/s on a Celeron 600 CPU. Using higher speed CPUs, throughput should increase proportionately.

The access concentrator name and PPPoE service name are used by clients to identify the access concentrator to register with. The access concentrator name is the same as the identity of the router displayed before the command prompt. The identity may be set within the /system identity submenu.

Note that if no service name is specified in WindowsXP, it will use only service with no name. So if you want to serve WindowsXP clients, leave your service name empty.

**Properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication</td>
<td>Authentication algorithm</td>
</tr>
<tr>
<td>default-profile</td>
<td>Default user profile to use</td>
</tr>
<tr>
<td>interface</td>
<td>Interface, which the clients are connected to</td>
</tr>
<tr>
<td>keepalive-timeout</td>
<td>Defines the time period (in seconds) after which the router is starting to send keepalive packets every second. If no traffic and no keepalive responses came for that period of time (i.e. 2 * keepalive-timeout), not responding client is proclaimed disconnected.</td>
</tr>
<tr>
<td>max-mru</td>
<td>Maximum Receive Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500-byte Ethernet link, set the MTU to 1480 to avoid fragmentation of packets)</td>
</tr>
<tr>
<td>max-mtu</td>
<td>Maximum Transmission Unit. The optimal value is the MTU of the interface the tunnel is working over decreased by 20 (so, for 1500-byte Ethernet link, set the MTU to 1480 to avoid fragmentation of packets)</td>
</tr>
<tr>
<td>max-sessions</td>
<td>Maximum number of clients that the AC can serve. '0': no limitations.</td>
</tr>
<tr>
<td>mrru</td>
<td>Maximum packet size that can be received on the link. If a packet is bigger than tunnel MTU, it will be split into multiple packets, allowing full size IP or Ethernet packets to be sent over the tunnel. Read more &gt;&gt;</td>
</tr>
<tr>
<td>one-session-per-host</td>
<td>Allow only one session per host (determined by MAC address). If a host will try to establish a new session, the old one will be closed</td>
</tr>
<tr>
<td>service-name</td>
<td>The PPPoE service name. Server will accept clients which sends PADI message with service-names that matches this setting or if service-name field in PADI message is not set.</td>
</tr>
</tbody>
</table>
Notes
The default keepalive-timeout value of 10 is OK in most cases. If you set it to 0, the router will not disconnect clients until they explicitly log out or the router is restarted. To resolve this problem, the one-session-per-host property can be used.

Security issue: do not assign an IP address to the interface you will be receiving the PPPoE requests on.

Specifying MRRU means enabling MP (Multilink PPP) over single link. This protocol is used to split big packets into smaller ones. Under Windows it can be enabled in Networking tag, Settings button, "Negotiate multi-link for single link connections". Their MRRU is hardcoded to 1614. This setting is useful to overcome PathMTU discovery failures. The MP should be enabled on both peers.

Example
To add PPPoE server on ether1 interface providing ex service and allowing only one connection per host:

```
[admin@MikroTik] interface pppoe-server server> add interface=ether1 service-name=ex one-session-per-host=yes
[admin@MikroTik] interface pppoe-server server> print
Flags: X - disabled
  0 X service-name="ex" interface=ether1 mtu=1480 mru=1480 mrru=disabled
      authentication=mschap2,mschap,chap,pap keepalive-timeout=10
      one-session-per-host=yes max-sessions=0 default-profile=default
[admin@MikroTik] interface pppoe-server server>
```

PPPoE Server

Sub-menu: /interface pppoe-server
There are two types of interface (tunnel) items in PPTP server configuration - static users and dynamic connections. An interface is created for each tunnel established to the given server. Static interfaces are added administratively if there is a need to reference the particular interface name (in firewall rules or elsewhere) created for the particular user. Dynamic interfaces are added to this list automatically whenever a user is connected and its username does not match any existing static entry (or in case the entry is active already, as there can not be two separate tunnel interfaces referenced by the same name). Dynamic interfaces appear when a user connects and disappear once the user disconnects, so it is impossible to reference the tunnel created for that use in router configuration (for example, in firewall), so if you need a persistent rules for that user, create a static entry for him/her. Otherwise it is safe to use dynamic configuration. Note that in both cases PPP users must be configured properly - static entries do not replace PPP configuration.

Property Description
- encoding (read-only: text) - encryption and encoding (if asymmetric, separated with '/') being used in this connection
- mru (read-only: integer) - client's MRU
- mtu (read-only: integer) - client's MTU
- name (name) - interface name
- remote-address (read-only: MAC address) - MAC address of the connected client
- service (name) - name of the service the user is connected to
- uptime (read-only: time) - shows how long the client is connected
- user (name) - the name of the connected user (must be present in the user database anyway)
Example

To view the currently connected users:

```
[admin@MikroTik] interface pppoe-server> print
Flags: X - disabled, D - dynamic, R - running
 #   NAME       USER      SERVICE   REMOTE... ENCODING  UPTIME
 0  DR <pppoe-ex> user      ex        00:0C:... MPPE12... 40m45s
[admin@MikroTik] interface pppoe-server>
```

To disconnect the user ex:

```
[admin@MikroTik] interface pppoe-server> remove [find user=ex]
[admin@MikroTik] interface pppoe-server> print
[admin@MikroTik] interface pppoe-server>
```

Application Examples

PPPoE in a multipoint wireless 802.11g network

In a wireless network, the PPPoE server may be attached to an Access Point (as well as to a regular station of wireless infrastructure). Either our RouterOS client or Windows PPPoE clients may connect to the Access Point for PPPoE authentication. Further, for RouterOS clients, the radio interface may be set to MTU 1600 so that the PPPoE interface may be set to MTU 1500. This optimizes the transmission of 1500 byte packets and avoids any problems associated with MTUs lower than 1500. It has not been determined how to change the MTU of the Windows wireless interface at this moment.

Let us consider the following setup where the MikroTik Wireless AP offers wireless clients transparent access to the local network with authentication:

First of all, the wireless interface should be configured:
Now, configure the Ethernet interface, add the IP address and set the default route:

```bash
[admin@PPPoE-Server] ip address> add address=10.1.0.3/24 interface=Local
Flags: X - disabled, I - invalid, D - dynamic

<table>
<thead>
<tr>
<th>#</th>
<th>ADDRESS</th>
<th>NETWORK</th>
<th>BROADCAST</th>
<th>INTERFACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10.1.0.3/24</td>
<td>10.1.0.0</td>
<td>10.1.0.255</td>
<td>Local</td>
</tr>
</tbody>
</table>
```

```bash
[admin@PPPoE-Server] ip route> add gateway=10.1.0.1
```

Finally, we can set up PPPoE clients:

```bash
[admin@PPPoE-Server] interface pppoe-server server> add interface=wlan1 service-name=mt one-session-per-host=yes disabled=no
```

```bash
[admin@PPPoE-Server] interface pppoe-server server> print
Flags: X - disabled

|   | service-name="mt" interface=wlan1 max-mtu=1480 max-mru=1480 mrru=disabled authentication=pap,chap,mschap1,mschap2 keepalive-timeout=10 one-session-per-host=yes max-sessions=0 default-profile=default |
```

Finally, we can set up PPPoE clients:
Thus we have completed the configuration and added two users: w and l who are able to connect to Internet, using PPPoE client software.

Note that Windows XP built-in client supports encryption, but RASPPPOE does not. So, if it is planned not to support Windows clients older than Windows XP, it is recommended not to require encryption. In other case, the server will accept clients that do not encrypt data.

**Troubleshooting**

- **I can connect to my PPPoE server. The ping goes even through it, but I still cannot open web pages**

  Make sure that you have specified a valid DNS server in the router (in /ip dns or in /ppp profile the dns-server parameter).

- **The PPPoE server shows more than one active user entry for one client, when the clients disconnect, they are still shown and active**

  Set the keepalive-timeout parameter (in the PPPoE server configuration) to 10 if You want clients to be considered logged off if they do not respond for 10 seconds.

  Note that if the keepalive-timeout parameter is set to 0 and the only-one parameter (in PPP profile settings) is set to yes then the clients might be able to connect only once. To resolve this problem one-session-per-host parameter in PPPoE server configuration should be set to yes.

- **My Windows XP client cannot connect to the PPPoE server**

  You have to specify the "Service Name" in the properties of the XP PPPoE client. If the service name is not set, or it does not match the service name of the MikroTik PPPoE server, you get the "line is busy" errors, or the system
shows “verifying password - unknown error”

- I want to have logs for PPPoE connection establishment

Configure the logging feature under the /system logging facility and enable the PPP type logs. Read more >>

References

Article Sources and Contributors


Image Sources, Licenses and Contributors

