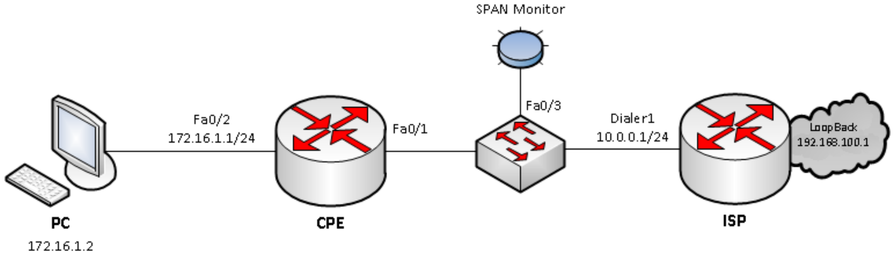
Lab 2.1 – PPPoE using Ethernet

Scenario



PPPoE provides an emulated (and optionally authenticated) point-to-point link across a shared medium, typically a broadband aggregation network such as those found in DSL service providers. In fact, a very common scenario is to run a PPPoE client on the customer side (commonly on a SOHO Linksys or similar brand router), which connects to and obtains its configuration from the PPPoE server (head-end router) at the ISP side. Note that ATM is typically run between the customer's modem and the DSLAM, though it will be transparent in this lab since our PPPoE client exists on a separate device.

Step 1 – Usual routers initial configuration

We've got two routers, CPE as client side and ISP as server side. CPE and ISP are interconnected with Ethernet link but it could be any contemporary L1/L2 technology in that case it's sometimes referred as PPPoEoS (over serial) or PPPoEoA (over ATM). Configure all of the following according to picture:

- Hostnames
- Ethernet interfaces
- VTY remote access and console synchronous logging

Step 2 – Server configuration

First task at the ISP end is to configure a Broadband Aggregation (BBA) group which will handle incoming PPPoE connection attempts. Here we can also apply PPPoE session limits. For example, we can limit the number of sessions established per client MAC address (setting this limit to 2 allows a new session to be established immediately if the prior session was orphaned and is waiting to expire).

```
(config)# bba-group pppoe SKUPINA
(config-bba-group)# virtual-template 1
(config-bba-group)# sessions per-mac limit 2
```

Next we'll create the virtual template for the customer-facing interface. When a PPPoE client initiates a session with this router, the router automatically spawns a virtual interface to represent that point-to-point connection. At a minimum, we'll need to configure two items on our virtual template: an IP address, and a pool of IP addresses from which clients are assigned a negotiated address (similar in operation to DHCP). Back in global configuration mode, we define a local IP pool named POOL with the starting and ending addresses of an IP range. DHCP could be used instead.

```
(config)# ip local pool POOL 10.0.0.2 10.0.0.10
```

```
(config)# interface virtual-template 1
(config-if)# ip address 10.0.0.1
(config-if)# ip mtu 1492
(config-if)# peer default ip address pool POOL
```

Last, we need to enable our PPPoE group on the interface facing the customer. Note that this interface should not have an IP address because the addressing is provided by our virtual template.

```
(config)# interface fastethernet 0/0
(config-if)# no ip address
(config-if)# pppoe enable group SKUPINA
(config-if)# no shutdown
```

Step 3 – Client configuration

Client configuration is relatively simple. We create a dialer interface to handle the PPPoE connection, and tie it to a physical interface which provides the transport.

```
(config)# interface dialer 1
(config-if)# dialer pool 1
(config-if)# encapsulation ppp
(config-if)# ip address negotiated
(config-if)# ip mtu 1492
```

The line **ip address negotiated** instructs the client to use an IP address provided by the PPPoE server. The PPP header adds 8 bytes of overhead to each frame. Assuming the default Ethernet MTU of 1500 bytes, we'll want to lower our MTU on the dialer interface to 1492 to avoid unnecessary fragmentation.

Lastly we assign our ISP-facing interface to our newly created PPPoE dial group.

```
(config)# interface f0/1
(config-if)# no ip address
(config-if)# pppoe-client dial-pool-number 1
(config-if)# no shutdown
```

If everything goes well you should observe following messages in console:

```
%DIALER-6-BIND: Interface Vi1 bound to profile Di1
%LINK-3-UPDOWN: Interface Virtual-Access1, changed
state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access1, changed state to up
```

Configure the NAT between inside company interface and dialer interface with your everlasting knowledge of CCNA and verify connection to ISP from host.

Step 4 – Verify setup

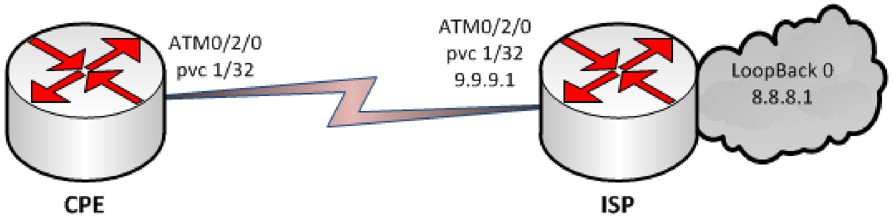
If you're working on GNS3 capture packets on inter-router Ethernet link. If you're working on real HW prepare sniff SNAP monitor on communication between ISP and CPE using L2 transparent switch.

Issue command **debug pppoe events** on ISP and flap dialer interface with **shutdown** and **no shutdown** commands. Observe PPPoE message exchange in console and also in wireshark of monitoring host.

Notice that fastethernet interfaces between CPE and ISP are used for data transfer even thou they have no IP address configured. Every such customer has point-to-point connection with ISP even that Ethernet is shared medium. What a miracle of PPPoE.

Lab 2.2 – PPPoE using DSL line

Scenario



Step 1 – Usual routers initial configuration

We've got two routers, CPE as client side and ISP as server side and they will be connected together back-to-back using RJ-11 telephone cable (pink). Configure all of the following according to picture

- Hostnames
- VTY remote access and console synchronous logging

Step 2 – Define properties of DSL lines

These two routers have WIC-1SHDSL-v2 cards installed. They both will have same few lines of controller configuration:

```
(config)# controller dsl 0/2/0
(config-controller)# dsl-mode shdsl symmetric annex B
(config-controller)# mode atm
```

But they will differ in line termination parametr on server side:

```
(config-controller)# line-term CO
```

…and on client side:

```
(config-controller)# line-term CPE
```

If both sides are configured properly and there is no problem in HW, you will see following messages in 30-45 seconds:

```
DSL 0/2/0 controller Link up! line rate: 2304 Kbps
%CONTROLLER-5-UPDOWN: Controller DSL 0/2/0, changed
state to up
%LINK-3-UPDOWN: Interface ATM 0/2/0, changed state to
up
%LINEPROTO-5-UPDOWN: Line protocol on Interface ATM
0/2/0, changed state to up
```

Step 3 – Configuring PPPoE on the client side

For DSL to work we need to specify which ATM virtual connection will be used and what encapsulation type will be used. For normal DSL use, PVC connection encapsulation does not need to be explicitly specified, because the default command **encapsulation aal5snap** is used. If ADSL is used, ATM interface is configured by default with the **dsl operationg mode auto** command. This default value should not be altered since it allows the client router to automatically detect the proper modulation method to use.

Next configure the PPPoE client encapsulation and specify which dialer interface to use. Next bind the ATM interface to to a dialer interface.

So without further delay enter following commands on client:

```
(config)# interface atm 0/2/0
(config-if)# pvc 1/32
(config-if-atm-vc)# pppoe-client dial-pool-number 1
```

```
(config)# interface dialer 0
(config-if)# ip address negotiated
(config-if)# ip mtu 1492
(config-if)# encapsulation ppp
(config-if)# dialer pool 1
```

Step 4 – Configure PPPoE on the server side

Server side of the aggregation router is much more complicated to configure. VPI/VCI number is the same as on the client side because we are connected back-to-back. However in real DSL deployment, CPE device is connected to the aggregation router via DSLAM and each DSLAM is connected through number of ATM switches. Aggregation router will therefore contain numerous different virtual circuits, for example each serving single DSLAM.

In compare with previous lab notice that we could bind the address of virtual template with loopback and also bba-group profile could be define as global which will be used if no appropriate named bba-group has been found.

```
(config)# bba-group pppoe global
(config-bba-group)# virtual-template 1
```

```
(config)# interface atm 0/2/0
(config-if)# pvc 1/32
(config-if-atm-vc)# protocol pppoe [group global]
```

```
(config)# ip local pool LOCAL 9.9.9.1 9.9.9.5
```

```
(config)# interface loopback 0
(config-if)# ip address 8.8.8.1
```

```
(config)# interface virtual-template 1
(config-if)# ip unnumbered loopback 0
(config-if)# ip mtu 1492
(config-if)# peer default ip address pool LOCAL
```

Step 5 – Secure PPPoE connection

On client side add to dialer interface appropriate authentication method:

```
(config-if)# ppp chap password prd31
```

…and on server side add user (hostname of client), its password and change authentication method:

```
(config)# username <client> password prd31
(config)# interface virtual-template 1
(config-if)# ppp authentication chap
```

Step 6 – Verify setup

List active sessions on client with **show pppoe session** command and check pool usage on server with **show ip local pool LOCAL** command. Then compare content of routing tables of both routers with classical **show ip route**.