

## PPPoE\_Client-Server\_setup\_and\_configuration

This is an example of configuring PPPoE in a back-to-back scenario. The objective here is to simulate a PPPoE server (typically found on the ISP end)

Client--Fa0/0-----Fa0/0--Server

In this example, we're using 2 routers connected back-to-back on their Fa0/0 ports. Here're the parameters being simulated

1. Authentication using Chap/Pap (Username: cisco, password: sisco)
2. Client being authenticated by the server (one way authentication)
3. IP address being negotiated using IPCP

Here's the relevant configuration needed on the client router

```
interface FastEthernet0/0
no ip address
no ip redirects
no ip unreachable
no ip proxy-arp
ip virtual-reassembly
ip tcp adjust-mss 1452
duplex auto
speed auto
pppoe enable
pppoe-client dial-pool-number 1
!

interface Dialer1
ip address negotiated
encapsulation ppp
dialer pool 1
dialer-group 1
no cdp enable
ppp authentication pap chap callin
ppp pap sent-username cisco password sisco
ppp chap hostname cisco
ppp chap password sisco
!
ip route 0.0.0.0 0.0.0.0 Dialer1
```

Here's the relevant configuration needed on the Server

```
username cisco password sisco
!
bba-group pppoe global
virtual-template 1
!
```

## PPPoE\_Client-Server\_setup\_and\_configuration

```
interface FastEthernet0/0
ip address 10.252.102.49 255.255.255.240
load-interval 30
duplex auto
speed auto
pppoe enable group global
!

interface Virtual-Template1
mtu 1492
ip unnumbered FastEthernet0/0
peer default ip address pool pppoepool
ppp authentication pap chap
!
ip local pool pppoepool 10.10.10.1 10.10.10.200
!
```

Here's how the flow will be

1. Client negotiates PPPoE using PADO, PADi and PADr with the server, both client and server move into PPP-->LCP phase
2. Client/Server negotiate authentication, other parameters
3. Server asks the client for a username/password (using pap, as pap is configured before chap)
4. Client sends out a username/password configured in it's dialer
5. Server authenticates this username/password against its global username/password list (alternatively a AAA/Radius server)
6. Client and Server both move on to IPCP phase
7. Client sends an ip address of 0.0.0.0 (asking for an ip address from the Server)
8. Server hands out an IP address from its pool (in this case pppoepool)
9. Client/Server finish IPCP phase and the link comes up

At this time, the link should be up and able to pass traffic