PPPoE Primer

Introduction

With simple TCP/IP based networks there are two common ways to quickly establish a network connection.

- Providing a **DHCP server** for the devices connecting to the network. Your phone and laptop connecting to the WiFi router at home is a classic example.
- Providing a **PPPoE server** for devices connecting to the network. If you are a client of a WISP, the Customer premises equipment (CPE) at your home / office will most likely be a PPPoE client in order to provide connectivity to the WIPS's network.
- If you are not working for a WISP or ISP, the exposure to PPPoE might be limited and this page will serve as a background primer.

Why use PPPoE

- PPPoE comes in handy when you want to **manage access to a network**.
- The most basic PPPoE servers require a username and password combination before establishing a connection from the client.
- The client is thus configured with a username and password in order to identify itself to the PPPoE server.
- More advanced PPPoE servers include support for RADIUS which feature a central user store and options like bandwidth limiting or data limits and usage tracking.
- This all makes PPPoE a favorite method used by ISPs and WISPs to manage client connections.
- Compare this now with the home network of a LTE router. If someone connects to it (making use of DHCP) there is not really a way for you to prevent them from establishing a connection or limiting their bandwidth.
- On the home network you might as an alternative option to PPPoE use a Captive Portal on the break-out point of a network with a DHCP server in order to manage network access, bandwidth and usage quotas. (Like the typical Guest WiFi networks)

The PPPoE Protocol

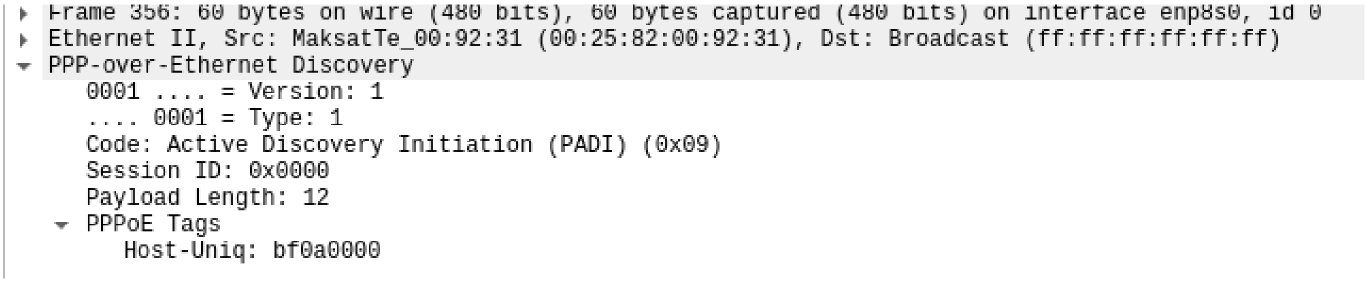
- There are lots of documentation on the inner workings of the PPPoE protocol.
- This section will just cover the main points.
- The PPPoE protocol is a **layer 2 protocol**.
- This means that it does not contain an IP Address and communication is between MAC Addresses (on the Data Link Layer)
- This is in a way similar to a DHCP discovery packet for instance where the request is broadcasted on the broadcast domain (data link layer). That packet is used in order to try and obtain an IP Address but does not have an IP Address
- For a PPPoE client to find out if there are any PPPoE servers around it, it starts with the **PPPoE Discovery** stage. (Broadcast)
- PPPoE servers will then reply to the client informing it that they are available. (Usually there is only one PPPoE server running in a broadcast domain.)
- The client will then proceed to communicate directly with the server (Unicast) on the MAC level / layer 2 in order to establish the **PPP Session**.
- Once the session is establish, the data transmitted between the PPPoE client and PPPoE server

will be wrapped inside the PPPoE and PPP protocols.

- Inside these packets will be the data used for normal TCP/IP communication.
- See the screenshots below of some packets captured on a PPPoE network.

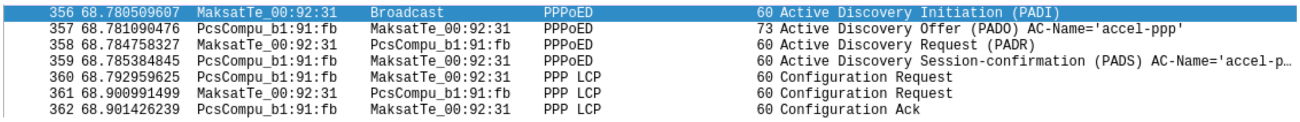
PPPoE Discovery

- Here's the content of the first packet started by the PPPoE Client. As you can see it is a Layer2 broadcast with destination ff:ff:ff:ff:ff:ff.



```
▸ Frame 356: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp8s0, id 0
▸ Ethernet II, Src: MaksatTe_00:92:31 (00:25:82:00:92:31), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▾ PPP-over-Ethernet Discovery
    0001 .... = Version: 1
    .... 0001 = Type: 1
    Code: Active Discovery Initiation (PADI) (0x09)
    Session ID: 0x0000
    Payload Length: 12
  ▾ PPPoE Tags
      Host-Uniq: bf0a0000
```

- The PPPoE server respond and now they start to communicate directly with each other.

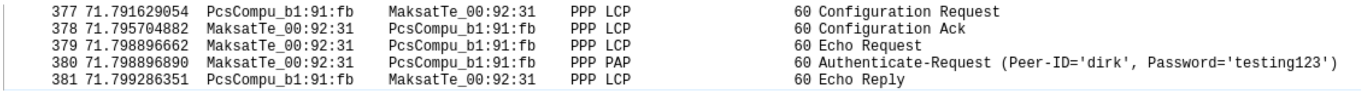


```
356 68.780509607  MaksatTe_00:92:31  Broadcast          PPPoED      60 Active Discovery Initiation (PADI)
357 68.781090476  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPPoED      73 Active Discovery Offer (PADO) AC-Name='accel-ppp'
358 68.784758327  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPPoED      60 Active Discovery Request (PADR)
359 68.785384845  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPPoED      60 Active Discovery Session-confirmation (PADS) AC-Name='accel-p…
360 68.792959625  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPP LCP     60 Configuration Request
361 68.900991499  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPP LCP     60 Configuration Request
362 68.901426239  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPP LCP     60 Configuration Ack
```

- The client is happy with the PPPoE server and will try next to authenticate (PPP protocol)

PPP Authentication

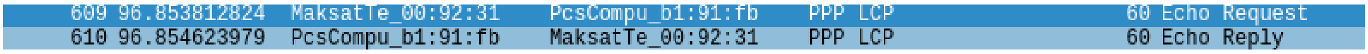
- We are using PAP in this sample which is why the password is in clear-text.



```
377 71.791629054  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPP LCP     60 Configuration Request
378 71.795704882  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPP LCP     60 Configuration Ack
379 71.798896662  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPP LCP     60 Echo Request
380 71.798896890  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPP PAP     60 Authenticate-Request (Peer-ID='dirk', Password='testing123')
381 71.799286351  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPP LCP     60 Echo Reply
```

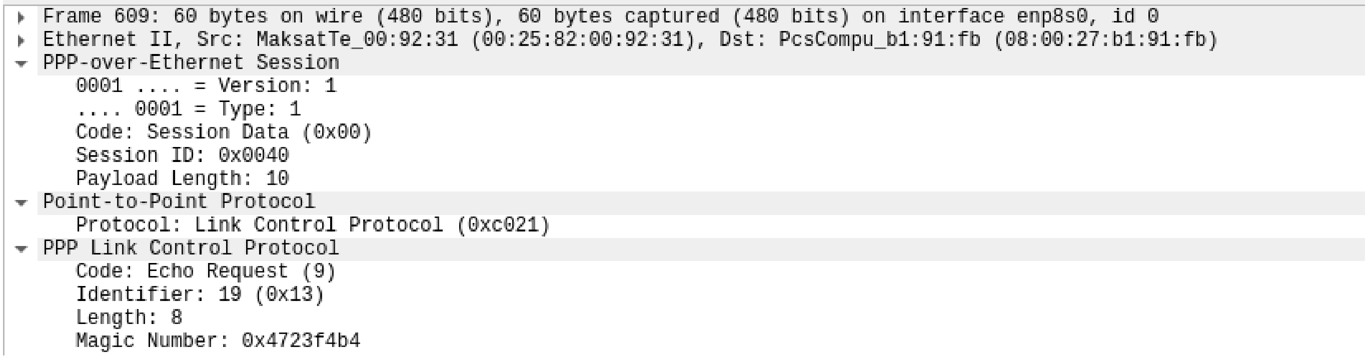
PPP Are you still there?

- Once the session is established, PPP will do a Ping to determine that the connection is still up.



```
609 96.853812824  MaksatTe_00:92:31  PcsCompu_b1:91:fb  PPP LCP     60 Echo Request
610 96.854623979  PcsCompu_b1:91:fb  MaksatTe_00:92:31  PPP LCP     60 Echo Reply
```

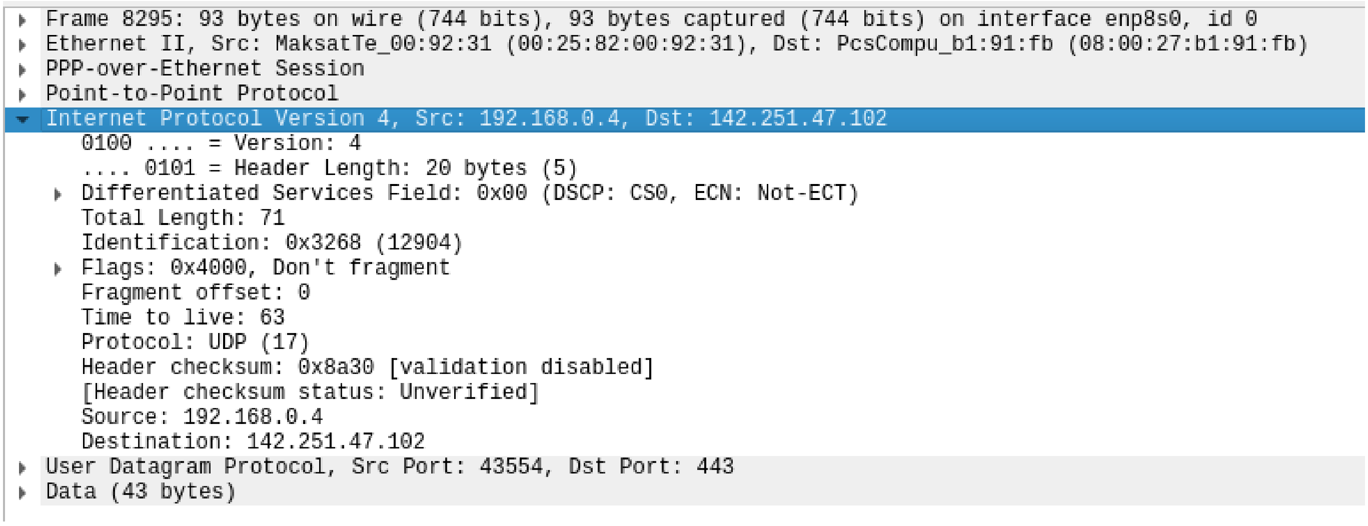
- Here's the packet's content



```
▸ Frame 609: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface enp8s0, id 0
▸ Ethernet II, Src: MaksatTe_00:92:31 (00:25:82:00:92:31), Dst: PcsCompu_b1:91:fb (08:00:27:b1:91:fb)
▾ PPP-over-Ethernet Session
    0001 .... = Version: 1
    .... 0001 = Type: 1
    Code: Session Data (0x00)
    Session ID: 0x0040
    Payload Length: 10
▾ Point-to-Point Protocol
    Protocol: Link Control Protocol (0xc021)
▾ PPP Link Control Protocol
    Code: Echo Request (9)
    Identifier: 19 (0x13)
    Length: 8
    Magic Number: 0x4723f4b4
```

PPPoE and PPP with Data

- Finally you can see how the payload is wrapped inside a packet with PPPoE and PPP on the outside to transport it between the PPPoE client and PPPoE Server.



```
▸ Frame 8295: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface enp8s0, id 0
▸ Ethernet II, Src: MaksatTe_00:92:31 (00:25:82:00:92:31), Dst: PcsCompu_b1:91:fb (08:00:27:b1:91:fb)
▸ PPP-over-Ethernet Session
▸ Point-to-Point Protocol
▾ Internet Protocol Version 4, Src: 192.168.0.4, Dst: 142.251.47.102
     0100 .... = Version: 4
     .... 0101 = Header Length: 20 bytes (5)
   ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
     Total Length: 71
     Identification: 0x3268 (12904)
   ▸ Flags: 0x4000, Don't fragment
     Fragment offset: 0
     Time to live: 63
     Protocol: UDP (17)
     Header checksum: 0x8a30 [validation disabled]
     [Header checksum status: Unverified]
     Source: 192.168.0.4
     Destination: 142.251.47.102
▸ User Datagram Protocol, Src Port: 43554, Dst Port: 443
▸ Data (43 bytes)
```

Conclusion

- As you can see the PPPoE protocol is plain and simple, similar to Ethernet (which is on a lower networking layer).
- This explain its popularity. As they say *Simplicity is the Ultimate Sophistication*.