USNET® PPPoE Quick Guide

Version 1.01
August 2004

U S SOFTWARE®
EMBEDDED EXCELLENCE

Copyright and Trademark Information

Lantronix, Inc.
15353 Barranca Parkway
Irvine, CA 92618
(949)453-3990
Fax (949) 453-3995

For Support Contact:
Micro Digital Associates, Inc.
2900 Bristol Street, #G204
Costa Mesa, CA 92626
(714) 437-7333
support@smxinfo.com
www.smxinfo.com

This quick guide describes the necessary steps to configure and build PPPoE (PPP over Ethernet) with USNET®. Note that PPPoE support can be selected to create either a PPPoE host, or a PPPoE Access Concentrator.

1. Test USNET® on the target without PPPoE integration.

This includes running ltest and at least some of the other tests such as mttest or emtest.

Note that PPP must be configured even though there may not be a serial interface on the target. In such a case, add ppp to the PRODLIST in config.mak and build as normal.

```
PRODLIST += ppp
```

2. Install PPPoE

Install USNET®, PPP and PPPoE into the same directory tree. If they were purchased together, this is already the case. Otherwise, install PPPoE into a copy of the already existing USNET® and PPP tree. USNET® makefiles will automatically detect the presence of PPPoE.

3. Define the target interface

In netconf.c, netdata must contain a host entry for the PPPoE interface. Replace the Ethernet field with PPPOE in the already functional entry.

For example:

Previously, the entry may have been:

```
"test", "pppoe", C, {192,168,1,1}, EA0, 0, Ethernet, PCI, 0, 0,
```

It should then be changed to the following:

```
"test", "pppoe", C, {192,168,1,1}, EA0, 0, PPPOE, PCI, 0, 0,
```

If the IP address is defined by the Access Concentrator, define the IP address as 0.0.0.0. Additionally, create an entry for the peer host so that PPP can store the remote IP address for later.

netconf.c includes a symbolic constant that defines X as {0,0,0,0}, so this shorthand for the 0 IP address is used in the example entries that follow.

```
"ac", "pppoe", C, X, EA0, ROUTER, 0, 0, 0, 0,
"test", "pppoe", C, X, EA0, 0, PPPOE, PCI, 0, 0,
```

If USNET is being run as an Access Concentrator, additional entries in the netdata[] table can be set up so that an IP address is delivered to a PPPoE host as part of PPP configuration.

As an example, consider an Access Concentrator that is located between a PPPoE host and a gateway. The connection between the PPPoE host and the Access Concentrator, and between the Access Concentrator and the upstream gateway are both Ethernet, and are on separate physical networks. The Access Concentrator is configured so that each network interface is associated with a different subnet.

The gateway is configured with a subnet mask that makes both the Access Concentrator and the PPPoE host appear to be on the same subnet. In this example, only the Access Concentrator is aware of what IP address is assigned to the PPPoE host, so the configuration of the gateway is simplified by suggesting that the Access Concentrator and the PPPoE host are on the same subnet.

This configuration is useful for testing and demonstrating the features of the Access Concentrator, but other configurations may be more appropriate for different applications.

Here is a sample configuration for this example, which is discussed further below:

```
#define CC {0xff,0xff,0xff,0xe0}

"host", "tnet", CC, {192,168,0,33}, EA0, PROXYARP, 0, 0, 0, 0,
"test", "tnet", CC, {192,168,0,32}, EA0, 0, PPPOE, NE2000, 0,
        "IRNO=10 PORT=0x300",
"test", "enet", CC, {192,168,0,31}, EA0, 0, Ethernet, NE2000, 0,
        "IRNO=5 PORT=0x320",
"gw",   "enet", CC, {192,168,0,1}, EA0, ROUTER, Ethernet, 0, 0, 0,
```

Note that different network names ("enet" and "tnet") are used for each of the subnets. This is useful to help document which hosts and interfaces are associated with each network. Also, the network name could be specified by an application in order to establish a connection using a particular network interface.

The example includes a special subnet mask "CC". This subdivides a Class C network so that the upper 3 bits of the last octet in the IP address can be used to further define a network. So the hosts at addresses .33 and .32 are on the "tnet" subnet, and the hosts at addresses .31 and .1 are on the "enet" subnet.

The first entry has the name “host”, and is used for a PPPoE host that is served by the Access Concentrator. The Access Concentrator will provide this host with an IP address during PPP configuration, and the value for this address is specified in the address field. The PROXYARP flag is set so that the Access Concentrator will provide proxy ARP replies on the “enet” subnet for this host. With this proxy ARP feature in place, the gateway can be configured so that the Access Concentrator and the PPPoE host appear to be on the same subnet.

The next two entries are for the Access Concentrator, named “test”. The example Access Concentrator has two Ethernet interfaces, with each interface on a separate subnet. The “tnet” subnet is shared with the PPPoE host, and is configured for PPPoE. The “enet” subnet connects the Access Concentrator to the upstream gateway. This entry is a regular Ethernet entry.

An entry is also included for the gateway, and it includes the ROUTER flag to indicate that it should be used as the next hop for default datagram routing.

In addition, for use as an Access Concentrator, the following settings are suggested for local.h.

```
#define RELAYING 1
#define USS_PROXYARP
#define ACCESS_CONCENTRATOR
```

4. Build the test program

The test program is poetest.c. It depends on the presence of the console I/O functions Nchkchr() and Ngetchr(). Build it as a normal test application after performing all of the previous steps.

```
cd \ussw
omake poetest
```

The result should be appsrc\poetest.exe (.exe will vary by target) Instructions are printed to the screen upon execution of the test.

5. Further configuration items

Items specific to the operations of PPPoE host are contained within netsrc\pppoe.c. Edit the file configuration options as necessary.

The corresponding file for the Access Concentrator version is netsrc\pppoeac.c. The following notes describe the configurable values at the top of the file.

```
    #define PPPOE_TIMER_GRANULE 1000
```

The PPP timeout function for PPPoE sessions will be called using a period defined by this constant. The default value sets a frequency of once per second.

```
    #define PPPOE_ACNAME "AC-0000"
```

This string is delivered in the AC-Name tag when the Access Concentrator sends its PPPoE Active Discovery Offer (PADO) packet. This Access Concentrator name may be useful to the PPPoE host in deciding whether or not to set up a PPPoE with this Access Concentrator. In practice, this information is commonly ignored.

```
    #define MAX_SERVICE_NAME_LEN 16
```

This defines the size of the buffer that stores the string associated with the Service-Name tag. The Access Concentrator is set up to use a liberal policy on service names, accepting any name that is suggested by the host. This policy is suggested in the Security Considerations section of RFC 2516.

Similar buffer length definitions exist for the Host-Uniq, AC-Cookie and Relay-Session-Id tags.

```
    #define PNETS 2
```

This defines the number of physical network interfaces. State information for PPP sessions is stored in the network interface structure nets[]. Typically, each network interface is associated with a physical network interface, which may be a serial interface for PPP, or an Ethernet interface for a PPPoE host. A PPPoE Access Concentrator may support multiple PPPoE sessions over the same Ethernet interface.

In order to support this, some interface structures are used as "virtual interfaces". Interfaces with an index between 0 and PNETS - 1 correspond to physical interfaces. Indices between PNETS and NNETS - 1 correspond to virtual interfaces, which are mainly used to store PPP session state.

Note that NNETS which is defined in local.h needs to be larger than the number of physical interfaces. The default value of 4 happens to provide a little room for this.

The Access Concentrator will provide up to NNETS - PNETS PPPoE sessions. Once this limit is reached, the Access Concentrator will respond to incoming PPPoE Active Discovery Request (PADR) packets with a PPPoE Active Discovery Session-confirmation (PADS) packet that contains an AC-System-Error tag.