Networking Utilities, Libraries, etc.#

New Utilities#

The Networking project includes the following new or updated utilities and services:

- sbin/
  - brconfig New bridge configuration utility
  - ifconfig Updated interface configure utility
  - ifwatchd New "interface watch" utility
  - pfctl New packet filter control utility
  - pppoectl Updated PPPoE control utility
  - route Updated route configuration utility
  - setkey Updated IPSec key manipulation utility
  - sysctl Updated stack configuration / control utility

- usr/bin/
  - altqstat New traffic shaping ("ALTernate Queuing") statistics utility
  - arp Updated address resolution protocol utility
  - ftp Updated FTP client
  - netstat Updated netstat utility
  - ping Updated ping utilities for v4.
  - ping6 Updated ping utilities for v6.
  - sockstat New socket status utility
  - tbrconfig New Token Bucket Regulator configuration utility for an output queue
  - traceroute Updated route tracing utilities for v4.
  - traceroute6 Updated route tracing utilities for v6.

- usr/sbin
  - dhcp.client Updated dhcp client (handles longer interface names)
  - ftpd Updated FTP daemon
  - hostapd New ~Wi-Fi Host Access Point Daemon
  - inetd Updated Internet Daemon
  - nicinfo Updated network interface card information utility
  - pppd Updated PPP daemon
  - pppoed Updated PPP over Ethernet Daemon
  - tcpdump New packet capture and display utility
  - wpa_cli New WPA command line utility
  - wpa_passphrase New WPA network block pre-shared key generation tool
  - wpa_supplicant New WPA / WPA2 supplicant

Libraries#

- libcrypto OpenSSL cryptography library
- libipsec Updated ipsec library from the ipsec-tools project
- libssl OpenSSL library
- libpcap Packet Capture library
- libwlconfig ~Wi-Fi configuration library (in development)

Protocols#

Transparent Distributed Processing (aka QNET)#

Please see the QNET wiki page

Auto IP (or IPv4 Link Local Address as in RFC)#

"AutoIP" is an implementation of RFC 3927, "Dynamic Configuration of IPv4 Link-Local Addresses".
The source lives in sys/lsm/autoip and is implemented as a shared object so it can be loaded into the stack dynamically.

The code is pretty straight forward. A filter function is hooked in which examines the in/out ARP packets. For incoming ARP packets, it checks to see if it is in conflict with the Link Local Address which we've selected, and it then defends or re-selects the address accordingly. For outgoing packets, an ARP REPLY packet broadcast is sent onto the network (instead of uni-cast).

"AutoIP" can be started same as other lsm modules:

* ~io-pkt -d <driver> -p autoip

The only thing to watch out for, is if you use autoip in a multi-interface environment (say you have 2 network card). Try to use the "if=" option to target the protocol to a specific interface. If you really want both of your interfaces to run in "autoip" mode, make sure you understand exactly what could happen. The RFC has a whole chapter of "Considerations for Multiple Interfaces" that discusses this (see section 3 of rfc3927).