Treck TCP/IP Release 5.0 features List:

Note: For complete documentation, please see trusersupplement50.doc in the pdf directory, unless otherwise noted.

TCP

- TCP Retransmission Timer Control
    - Pause, resume or reset TCP retransmission timer on a per-socket basis
- Added user notification when the TCP connection transitions to the CLOSED state.

Timer

Changes to ease the load of the Timer Execute thread.

- Use a separate queue for suspended timers so that the Timer Execute Thread does not have to scan them.
- The task of adding and removing a timer is now done by the calling thread, instead of being deferred to the Timer Execute thread.
- Conditionally added multiple active timer queues with a cache of the next timer to expire per queue. (off by default.)

Sockets: Modifications to speed up socket lookup and random port insertion

- Added modification to delay inserting a socket in the socket lookup table until either connect/bind to a non zero port for TCP, bind without restriction for UDP, sendto, or listen occurs. This is so a socket is not visible to the outside world when it is not meant to be, and to avoid having to insert a connecting TCP socket twice (the first time when the socket is opened, and the second time when connect is called.)

- The socket lookup table has been split into two separate tables: one for non connected (listening) TCP sockets, and other sockets, and one for TCP connected sockets. The TCP connected socket lookup table now uses the 4-tuple as a search criteria (i.e. the local address is included in the search). Prior to this the local address search was done in a separate linked list. This is to speed up the TCP connected sockets lookup.

- We have conditionally added a hash table for each socket lookup table. The two hash tables can be used either with a doubly linked list per bucket, or a Red Black tree per bucket. Using a doubly linked list or a Red Black tree per bucket is a compile time decision.
    - By default the Red Black tree code is not enabled.
    - By default the hash tables are enabled, and the hash tables' sizes are TM_SOC_INDEX_MAX.
    - The hash table sizes are configurable at compile time and at run time, but prior to starting the stack.
    - You can use 1) two hash tables with a doubly-linked list per bucket (default), 2) two hash tables with a red black tree per bucket, 3) no hash table with a single doubly-linked list per socket lookup table, or 4) no hash table with a single red black tree per socket lookup table. We recommend combination #1.

- We changed tfRandPortInsert() to speed up picking a random local port for a socket. This is important when thousands of TCP connecting sockets are used concurrently.

IGMP

- Added support for IGMPv3 (please see IGMPv3 document in the pdf directory)

Web Server

Note: All web server documentation can be found in the web server document in the pdf directory

- Added TM_HTTPD_FLUSH flag to tfHttpdUserSendBuffer and tfHttpdUserSendResponseBody

DHCPv4 Client

- Added ability for the user to set the FQDN S bit off, or to set the FQDN N bit on in the DHCP messages sent to the server. (TM_USE_DHCP_FQDN)

DHCPv6 Client

Note: All DHCPv6 documentation can be found in the DHCPv6 document in the pdf directory

- Added support for DHCPv6 authentication
- Added tf6DhcpUserStop API

Crypto Library

Note: All crypto documentation can be found in the crypto and IPSec documents in the pdf directory

- Added support for SHA256

DNS Resolver

- Patch to address the CERT Vulnerability Note VU#800113. Treck now randomizes the source port number and Transaction ID for DNS requests on a per-request basis.

IPv6

- New API tf6GetAddrLifetimes(). This function retrieves the valid and preferred lifetimes for a given address.
- New tfInterfaceSetOptions option TM_6_DEV_OPTIONS_NO_INIT_DELAY. If this option is non-zero, Treck will not delay before sending a Neighbor Advertisement for the newly configured address. Treck will not send Router Solicitations. This violates the RFC!

Device

- Allow the user to enable/disable IP-level promiscuous mode without closing and re-opening the interface with TM_DEV_OPTIONS_IP_PROMISCUOUS.