New Networking Features in FreeBSD 6.0

André Oppermann <andre@FreeBSD.org>

EuroBSDCon 05
Basel, 27. November 2005

About

This talk gives an overview on what is new or has changed in FreeBSD 6.0 Networking Code compared to the FreeBSD 5-Series.

It is a continuation of the talk “FreeBSD 5 Network Enhancements” given at SUCON 04 on 3. September 2004 in Zürich.

It is by no means exhaustive and touches only the most important improvements.

Internal Changes - Stuff under the hood

Mbuf UMA

UMA (Universal Memory Allocator) is the primary memory allocator for fixed sized data structures in the FreeBSD kernel.

SLAB type allocator, fully SMP-aware and maintains per-CPU caches.

Mbufs of 256 bytes (minus attributes) and Mbuf clusters of 2048 bytes.

Cluster are attached to Mbufs. Mbuf serves as a descriptor for the cluster containing all associated attributes and packet information.

Special packet secondary zone holding pre-combined Mbuf+cluster pairs for more efficient allocation.

Internal Changes - Stuff under the hood

General Kernel SMP Locking

Main theme of FreeBSD 6.0 was finalizing the SMP locking of network related data structures.

Locking is necessary to prevent two CPUs accessing or manipulating the same data structure at the same time.

It is desirable to break down locking into fine-grained portions to avoid lock contention when multiple CPUs want to access related but independent data structures.

Too fine-grained locking is introducing overhead as each locking and unlocking operation has to be reliably propagated to all other CPUs.

Internal Changes - Stuff under the hood

Socket Buffer Locking

Every active or listening network connection is represented as a socket structure in the kernel.

The socket structure contains general bookkeeping on the socket and two socket buffers for transmitted and received packets and data.

Each socket structure has a general lock and two separate send and receive socket buffer locks.

Sending and receiving may happen concurrently.

Internal Changes - Stuff under the hood

Network Interface Structure Locking

The “ifnet” structure contains all information the kernel knows about network interfaces.

Network interfaces drivers may be loaded and unloaded any time as KLDs (Kernel Loadable Modules) or may arrive or depart as hot-plug interfaces like PCCARDs in laptops.

Parts of it are accessed by the upper half of the stack and parts by the drivers.

Any such unnecessary contention point has been identified and each party has got their own fields which they can manipulate independently without stalling the other when locking the structure.

Internal Changes - Stuff under the hood

Netgraph Locking

Netgraph is a concept where a number of small, single-job modules are stringed together to process packets through stages.

Netgraph may be best explained as an assembly line with many little functions along a conveyor belt versus one big opaque machine doing all work in one step.

Depending on the function and task of the module it was either locked as whole or every instance of it separately.

Internal Changes - Stuff under the hood

ARP Locking

ARP maps an IPv4 address to a hardware (MAC) address used on the ethernet wire.

ARP lookups and timeouts can happen at any time and may be triggered at any time from other machines on the network.

On SMP this has led to priority inversions and race conditions where one CPU was changing parts of an ARP entry when a second CPU tried to do the same.

An extensive rework and locking has been done to make ARP SMP-safe.

Internal Changes - Stuff under the hood

IP Multicast Locking

IP Multicast had many races too.

Most of them were related to changes of IP addresses on network interfaces and disappearing interfaces due to unload or unplug events.

Proper locking and ordering of locks has been instituted to make IP Multicast SMP-safe.

Internal Changes - Stuff under the hood

UDP Locking

All global variables have been removed to prevent locking contention and allow for parallel processing of incoming and outgoing packets.

IPX/SPX Locking

IPX/SPX is still in use at a non-negligible number of sites.

Significant effort has been made to lock SPX data structures and to make them SMP-safe.

IPX/SPX fortunately is not as complex as TCP/IP.

Internal Changes - Stuff under the hood

TCP Improvements

SACK has received many optimizations and interoperability bug fixes.

T/TCP support according to RFC1644 has been removed. The associated socket level changes however remain intact and functional.

FreeBSD was the only mainstream operating system that ever implemented T/TCP and its intrusive changes to the TCP processing made code maintenance hard.

NFS Improvements

NFS has been extensively regression-tested and received numerous bug fixes for many edge cases.

New Features

ng_netflow

Accounting of TCP and UDP flows in ISP backbones. It accumulates statistics on all TCP and UDP sessions and sends a summary UDP packet in the Netflow 5 format to a statistics collector for further processing.

ng_ipfw

Provides a way for injecting arbitrary matched IP packets into netgraph using ipfw. It works like an ipfw divert rule diverting to netgraph.

ng_nat

Provides netgraph access to the kernel-level libalias for network address translation.

ng_tcpmss

Changes the MSS (Maximum Segment Size) option of TCP SYN packets.

New Features

New DHCP Client

Port of the OpenBSD dhclient and adapted to FreeBSD specific needs.

It has many security features like privilege separation to prevent spoofed DHCP packets from exploiting the machine.

Additionally it is network interface link state aware and will re-probe for a new IP address when the link comes back up.

This is very convenient for laptop users who may connect to many different networks – be it wired or wireless LAN – many times a day.

New Features

IPFW Firewall

IPv6 rule support for matching and filtering of IPv6 packets.

ALTQ tagging of packets. ALTQ is an alternative queuing implementation for network interfaces and provides extensive QoS features.

ALTQ allows to define different queuing strategies on network interfaces to prioritize, for example, TCP ACKs on slow ADSL uplinks or delay and jitter sensitive VoIP (Voice over IP) packets.

IPDIVERT Loadable Module

The IPDIVERT module is used for NAT (Network address Translation) with IPFW. It is now a loadable module that can be loaded at runtime.

New Features

IPFILTER

Upgraded to version 4.1.8.

New Features

ICMP Replies

ICMP Source Quench support has been removed as it is deprecated for a long time now.

ICMP replies can now be sent from the IP address of the interface the packet came through into the system.

tcpdrop

The tcpdrop utility allows the administrator to drop or disconnect any active TCP connection on the machine.

This tool was ported from OpenBSD.

New Features

IP and TCP Socket Options

IP_MINTTL specifies the minimum TTL (Time To Live) a packet must have to be accepted on this socket. For GTSM RFC3682 support. Example is the Cisco IOS BGP implementation command "neighbor ttl-security".

IP_DONTFRAG sets the “Don’t Fragment” bit in the IP header and prevents sending of packets larger than the egress interface MTU with an EMSGSIZE error return value. Only for UDP and RAW sockets. On TCP it is controlled through the path MTU discovery option.

TCP_INFO allows the retrieval of vital metrics of an active TCP session such as estimated RTT, negotiated MSS and current window sizes. It is supposed to be compatible with a similar Linux socket option but still experimental.

New Features

Interface Polling

The network interface polling has been re-implemented to work correctly in SMP environments.

Polling is no longer a global configuration variable but enabled or disabled individually per interface if the driver supports it.

Most commonly found network drivers support polling.

For more information see polling(4).

New Features

Ethernet Bridge if_bridge

if_bridge is a fully fledged ethernet bridge supporting spanning tree and layer 2 or layer 3 packet filters on bridged packets.

It has been ported from NetBSD and replaces the previous bridge(4) implementation of FreeBSD.

Spanning tree is very important in bridged networks because it prevents loops in the topology. Ethernet packets do not have a TTL that is decremented on each hop and all packets in a looped bridge topology would cycle for an infinite amount of time in the network bringing it to a total standstill.

New Features

802.11 Wireless LAN

The Wireless LAN subsystem has been enhanced to support WPA authentication and encryption in addition to WEP.

It may be operated in client (Station) mode or AP (Access Point) mode. In both modes it supports the full WPA authentication and encryption.

The availability of the AP mode depends on the wireless LAN chip vendor, obtainable documentation (w/o NDA) and driver implementation.

All cited features are implemented in the “ath” driver for Atheros-based wireless cards which have the best documentation available.

New Features

CARP Common Address Redundancy Protocol

CARP is a special network interface and protocol that allows two or more routers to share the same IP address.

For all hosts using that router any fail-over from one to another is transparent and no service interruption occurs.

Routers in a CARP system may do hot-standby with priorities or loadsharing among them.

CARP has been ported from OpenBSD and is similar in functionality to VRRP from Cisco.

New Features

NDIS Driver Compatibility - Project Evil

Project Evil provides binary compatibility with Windows NDIS miniport drivers.

The NDIS compatibility layer emulates the Windows XP/Vista kernel network driver interface and allows Windows network card drivers to be run on FreeBSD. It supports wired and wireless LAN cards.

Many parts have been rewritten and updated as more Windows drivers could be tested, better documentation became available and a more throughout understanding of the NDIS nits developed.

While NDIS emulation works well it is only a last resort when all attempts of obtaining network chip documentation have failed. A native driver is always preferred!

That’s it. Any questions?