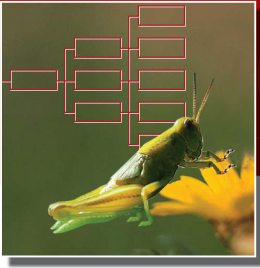


ClassHopper

Alternate Queuing Disciplines



ClassHopper is a small-footprint, high performance embedded alternate queuing discipline framework which includes classification mechanisms such as CBQ/WFQ for QoS management. It includes the ability to integrate with any classifiers in network packets including QoS tags in layer-2 (802.1p) and layer-3 headers (ToS bits). **ClassHopper** performs traffic classification, conditioning and scheduling in the input as well as output directions, enabling bandwidth control for different packet types. It also enables the network stack to handle overload conditions gracefully with selective packet discard and for effective avoidance of Denial of Service attacks at the network layer or higher.

ClassHopper is a robust, lean, high performance, alternate queuing discipline

framework for embedded devices.

ClassHopper's unique, advanced features include a class-based queues implementation and a framework for supporting custom queuing disciplines. Its small footprint and flexible APIs have been specifically designed for use in an embedded environment.

In addition, **ClassHopper** performs traffic classification, conditioning and scheduling in the input as well as output directions, enabling bandwidth control for different packet types. It offers unsurpassed flexibility through its built-in queuing implementations and customizable features.

ClassHopper is the optimal solution for voice and video applications that require guaranteed real-time throughput and bandwidth allocation for critical data

Features

- ❖ Implements CBQ, WFQ, and other queuing disciplines.
- ❖ Hierarchical bandwidth classes for both input & output directions.
- ❖ Scheduler combines priority, weighted round-robin and borrowing.
- ❖ Defends against denial of service (DoS) attacks by rate limiting based on traffic classification.
- ❖ Support for CPU types of either endian-ness including PowerPC, MIPS, X86, ARM/XScale.
- ❖ High-performance, adds minimal latency.
- ❖ Royalty-free!

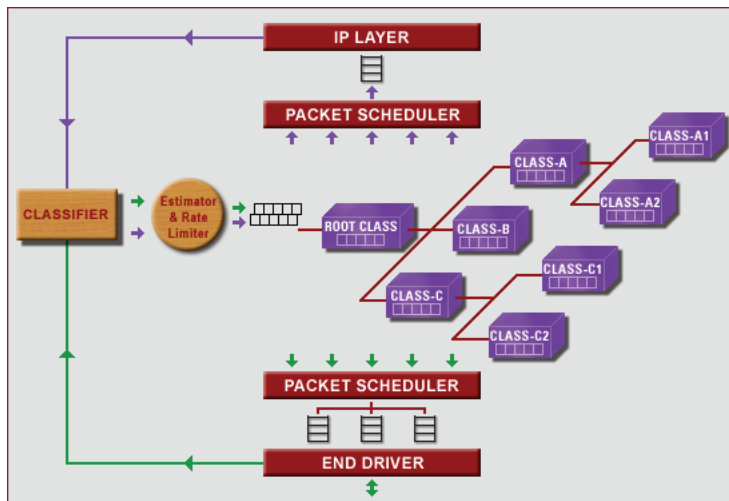


Fig 1: CLASS-BASED QUEUES

Special Features

- ❖ Provides QoS support within the network stack
- ❖ Works in both inbound and outbound directions
- ❖ Includes reference rules for Denial of Service (DoS) avoidance
- ❖ User customizable classifier and queue regulator functions
- ❖ Supports command line interface (CLI) and web management
- ❖ Enhanced memory management & partition support
- ❖ Ultra-small memory footprint and minimal dynamic memory usage

streams in the network stack.

It also enables the network stack to handle overload conditions gracefully with selective packet discard and for effective avoidance of Denial of Service attacks at the network layer or higher, by effectively policing the boundary between the driver and the stack using custom configured rules.

ClassHopper has been extensively validated on a variety of CPU architectures, and this minimizes development and integration efforts.

ClassHopper supports also multi-tasking and memory partitions.

Traffic Prioritization

Today's heterogeneous networks use a variety of protocols which creates an acute need to prioritize traffic to satisfy time-critical applications such as audio/video streaming while fulfilling the needs of less time-dependent applications, such as file transfers. Without active management of QoS in the network stack, different classes of traffic may share the same network queues and can interact with one another in ways that adversely affect application performance.

ClassHopper enables the use of sophisticated queuing disciplines instead of the standard tail-drop queues used in network stacks. This enables a tightly controlled yet flexible system of traffic prioritization and congestion management for network packets based on criteria such as Layer 2 fields (such as VLAN tags and 802.1p prioritization fields), Layer 3 fields (ToS bits, source and destination IP addresses or TCP/UDP ports), or any custom fields within a network packet.

Flexible Framework

The **ClassHopper** framework contains APIs that provide:

- ❖ Abstractions to support a wide variety of QoS functions.
- ❖ A platform to easily implement new queuing disciplines without knowledge of the underlying kernel implementation.
- ❖ A system model with 3 independent components.
 - **Classifier:** classifies a packet to a scheduling class based on predefined rules.
 - **Queuing Discipline:** implements packet scheduling and buffer management algorithms.
 - **Queue Regulator:** limits the amount that a driver can dequeue at a time.

- ❖ A string based command interface to configure the system parameters and functionality.

Queuing Disciplines

ClassHopper's ALTQ (alternate queues) system allows various queuing disciplines and traffic conditioning models to be seamlessly implemented using the framework.

These include:

- ❖ **CBQ:** Class Based Queuing.
- ❖ **WFQ:** Weighted-Fair Queuing.
- ❖ **RED:** Random Early Detection.
- ❖ **ECN:** Explicit Congestion Notification.
- ❖ **RIO:** RED with IN and OUT.
- ❖ **Flow-valve.**
- ❖ **H-FSC:** Hierarchical Fair Service Curve.

A complete reference implementation of CBQ and WFQ is included with **ClassHopper** and can be used as a guide to implement any of the above disciplines, or custom disciplines using the provided framework.

Management Support

ClassHopper supports an easily customizable management interface presented through a string-based command layer, which can be controlled through a web-server, with structured data files such as XML or via a CLI.

ClassHopper's rules can be very flexibly

Also Available

- ❖ **FireFly & GNAT**
IP-Filtering Firewall & NAT
- ❖ **GrandPPPrx**
Multi-Class / Multi-Link PPP
- ❖ **SSEcure Family**
SSL, SSH, IPsec/IKE
- ❖ **Switchcraft Family**
Layer 2 protocols - LACP, STP, RSTP, MSTP, 802.1x

Custom Solutions

Avail of the resources and expertise of **TeamF1**'s professional services team to help you build customized implementations of **ClassHopper** solutions such as support for custom queuing disciplines, hardware integration and custom configuration for easy provisioning.

Customization Flexibility

- ❖ Available in full-source format
- ❖ Modular design enables overriding the class management, estimator, and scheduler
- ❖ Controllable timer granularity for traffic bursts
- ❖ Framework for adding custom queuing disciplines
- ❖ Customization hooks and callouts
- ❖ Unwanted components can be scaled out

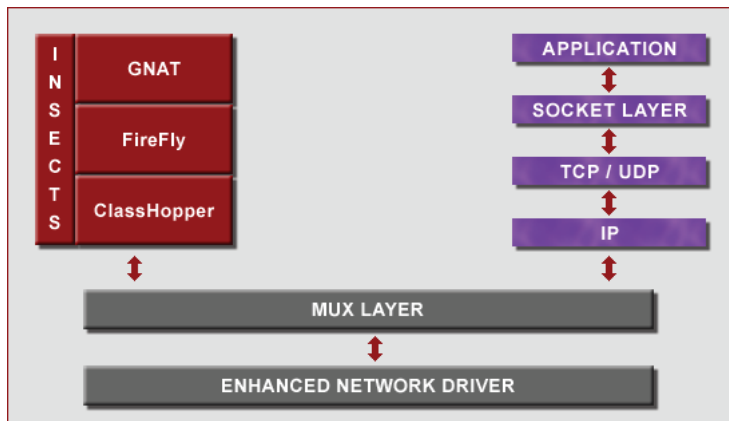


Fig 2: INSECTS "BUMP-ON-THE-STACK" MODEL

This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.

implemented using a variety of criteria including custom field specifications based on headers at Layer 2 (data-link layer) or higher. These fields can be used for classification of incoming traffic as well as for tagging outgoing traffic. This combined with the highly flexible framework for adding custom queuing disciplines enables the provisioning of highly sophisticated congestion management and traffic prioritization policies, and enabling end-to-end QoS.

Email: sales@TeamF1.com
 Web: www.TeamF1.com
 Ph: 510-505-9931 ext. 5
 Fax: (510) 505-9941



© 2011 TeamF1, Inc.