CS268 Project Proposal: Packet Classification Using Explicit Coordination

Mosharaf Chowdhury and Sameer Agarwal
{mosharaf,sameerag}@cs.berkeley.edu

October 13, 2009

Motivation

Packet classification is the key building block of many network services and functionalities (e.g., switching, forwarding, filtering, load balancing, policy-based routing and, QoS) in different layers [2]. In each case, a classifier determines which flow an arrived packet belongs to and takes action based on some given rules. However, high speed packet classification, in presence of arbitrarily large number of classification rules, is known to be computation-intensive, power hungry, and consequently, expensive [1].

Despite its prevalence, packet classification is implemented and deployed in an *ad-hoc* manner at different layers of the protocol stack [2]. This leads to (i) high implementation and configuration complexity, and (ii) difficulties in accurately capturing classification semantics.

State of the Art

Existing solutions try to tackle this problem by exploiting structure and redundancy in classification rules [1], by using faster or more powerful hardware [2,6], and by developing efficient mechanisms and algorithms [5,8]. Note that, all these solutions completely ignore the source and the destination of a packet and focus solely on the classifier.

In order to improve classification performance and to reduce semantic gaps, several solutions addressing these challenges (e.g., MPLS, DiffServ, HTTP cookies etc.) advocate pushing classification tasks to end hosts and edge network elements with the help of packet labeling/tagging mechanisms [4, 7]. In general, classification is reduced to simple label matching, where end hosts/edge network elements put labels on packets (as instructed by core network elements) and core network elements take actions associated with particular labels. Despite sharing common ideas, all these solutions are highly application- and protocol layer-specific without any general mechanism.

Project Overview

Our primary objective in this project is to design and develop a **cross-layer network primitive** that will improve scalability and performance by *offloading* classification tasks to end hosts and edge network elements (hereinafter referred to as *helpers*), and generalize common classification functionalities across different layers.

Initial Approach

Based on some initial packets, a classifier will decide on a label for a particular flow. It will then let the originating helper know, using a *signaling/handshaking protocol*, what label to put on packets from that particular flow. Once signaling is complete, the classifier will take decisions based on labels without resorting to complicated rule matching for future packets of that flow.

Milestones

M1 First of all, we will design a signaling protocol and suggest secure labeling mechanisms that will be able to withstand (a) asymmetric paths and path changes, (b) component failures and restarts (stateless vs (soft- and hard-) state-full design), and (c) trust, security, and privacy concerns (e.g., label forging, label guessing etc.). [**10.20.2009**]

M2 Next we will identify common classification functionalities, and decide upon logical placement of the classification primitive in the network protocol stack and actual additions/changes to packet headers (if required: e.g., using IP options might not require any changes). [**10.29.2009**]

M3 After that we will design and implement an API interface (along with the network primitive itself) to allow programmers to access the network primitive. We plan to implement the system on linux-based machines using the user-level Click modular router [3]. [**11.24.2009**]

M4 For evaluation, we will implement a firewall that takes advantage of the proposed cross-layer network primitive and compare its performance and scalability against a vanilla firewall on multiple large classification rule sets. We might require access to testbeds like EmuLab or PlanetLab for evaluation purposes. [**12.08.2009**]

References

[1] Pankaj Gupta and Nick McKeown. Packet classification on multiple fields. In *Proceedings of ACM SIGCOMM*, 1999.

[2] Sundar Iyer, Ramana Rao Kompella, and Ajit Shelat. ClassiPI: An architecture for fast and flexible packet classification. *IEEE Network*, pages 33–41, March-April 2001.

[3] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M. Frans Kaashoek. The Click modular router. *ACM Transactions on Computer Systems*, 18(3):263–297, 2000.

[4] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol label switching architecture, 2001.

[5] Sumeet Singh, Florin Baboescu, George Varghese, and Jia Wang. Packet classification using multidimensional cutting. In *Proceedings of ACM SIGCOMM*, pages 213–224, 2003.

[6] Ed Spitznagel, David Taylor, and Jonathan Turner. Packet classification using extended tcams. In *IEEE ICNP*, 2003.

[7] Ion Stoica, Scott Shenker, and Hui Zhang. Core-stateless fair queueing: A scalable architecture to approximate fair bandwidth allocations in high-speed networks. *IEEE/ACM Transactions on Networking*, 11(1):33–46, 2003.

[8] Fang Yu, T. V. Lakshman, Martin Austin Motoyama, and Randy H. Katz. SSA: A power and memory efficient scheme to multi-match packet classification. In *Proceedings of ACM ANCS*, pages 105–113, 2005.