

Notes on the per-flow packet count flow classifier

Mika Ilvesmäki and Jouni Karvo
Networking Laboratory
Helsinki University of Technology, Finland
Email:lynx@tct.hut.fi,jouni.karvo@hut.fi

Abstract

To realize a packet count classifier, in addition to the active flow table, a candidate table is needed, where information on flow candidates is kept. We will observe the temporal behavior of both the active flow table and flow candidate table size using actual traffic traces. The results indicate that the performance bottleneck in a packet count classifier lies within the candidate table management. Also, the changes in the candidate table size occur much faster than in the active flow table. Therefore, fast methods of creating entries and deletions in the candidate table are needed.

1. Introduction

According to Kumar et al. [2] flow classification has two possible tasks in the network: First, the process of identifying flows is a useful network property for optimizing resource usage in network equipment, and second, flows may also be associated with real-time performance guarantees by matching packet headers with pre-specified filters. In this work, we will examine the packet count classifier in more detail. Since our purpose is to illustrate the behavior of the packet count classifier we use real 1-hour traces (see Table 1).

Table 1: Traffic traces used in this work

Trace information	
Name	Number of packets
ebb900 ^a	1.1 million
dec-pkt-1 (dec1) ^b	2.9 million

^aHelsinki University of Technology (HUT) / Campus Area Network 09.00 May 29, 1997

^bAll dec-pkt-x traces are freely available at <http://ita.ee.lbl.gov/html/contrib/DEC-PKT.html>

The use of the packet count classifier as a method for resource based traffic classification has been studied in [3, 4]. In [1] a dynamic classifier where the values of packet count thresholds and timeouts are updated based on the resource usage of the IP router is presented. In practice, the packet count classifier is ap-

plied in a per-flow environment. While possibly feasible in the edges (or in restricted areas) of the network this scheme may not be efficiently implemented through the core network on the account of the scalability issues [5].

2. Packet count flow classifier

The packet count classifier observes the arrival of packets for flow candidates. As a packet arrives into a network node, an entry in the flow candidate table is created, or updated, if the packet can not be assigned to an active flow. If the packet count threshold is reached, the candidate flow is deleted, active flow is created and the flow is prioritized appropriately. The flow candidate table size is affected mainly by the following parameters: First, the candidate table size depends on the packet count threshold X . Second, in some packet count classifier schemes, there is a certain time window, T_X (Y of the X/Y -classifier), within which the threshold must be reached. If $T_X < \infty$, the classifier has to keep the timestamps of *every* packet of the candidate flow up to the threshold value X , since the timeout value T_X would be counted successively for each of the packets. We consider this as waste of resources and choose $T_X = \infty$, so that the remaining lifetime of the candidate flow table entry depends only on the last packet arrived in the flow. Third, the candidate deletion time, T_C , indicates the time that the candidate is held in the candidate table. If no new packets have been detected within T_C , the candidate is erased from the candidate table. Finally, the active flow timeout value, T_F , indicates the time that a flow stays in the active flow table after the last packet arrived. In order for the packet count classifier to function two tables are needed: the flow candidate table and the active flow table. This characteristic makes the packet count flow classifier unique, no candidate tables exist in other forms of classifiers.

3. Packet count classifier tables

We analyzed the behavior of the active flow table and the candidate flow table feeding the traces in Ta-

ble 1 to a classifier software that returned the size and changes with the size of the active and candidate tables. Figure 1 shows the active flow table and candidate table behavior as a function of time compared, respectively, with number of changes (additions and deletions summed) made to the tables within the measurement interval (1 second). The beginning of each trace shows

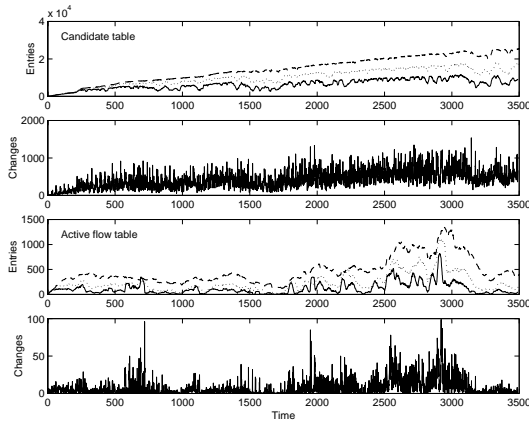


Figure 1: Comparison of active and candidate table changes with ebb900, $X = 5$ and $T_C = T_F = 30(, 60, 120)$ s

a transient, which becomes longer when the timeout T_C is increased. Also, the size of the candidate table grows when the timeout value increases. Since the traffic traces and the candidate birth rate stay the same this leads to longer transient period before reaching a steady-state. Note also, that even though the value of the active flow timeout, T_F , is increasing together with T_C , it does not seem to have any notable effect on reducing the candidate table size. The average ratio of the change rates indicates that changes occur in the candidate table at least hundred times faster than in the active table. In Figure 2, we observe the ratio of the sizes for candidate and active flow table. We can see three lines with varying candidate timeout values T_C . Even in the best case, the size of the candidate table is at least ten times that of the active flow table.

4. Conclusions

In this work we have studied the performance and behavior of the active flow table and flow candidate table of the packet count flow classifier. First, the size of the active table seems to reach steady state quite fast but the size is varying according to traffic characteristics. Second, the time for the candidate table size to reach the steady state is independent of the packet

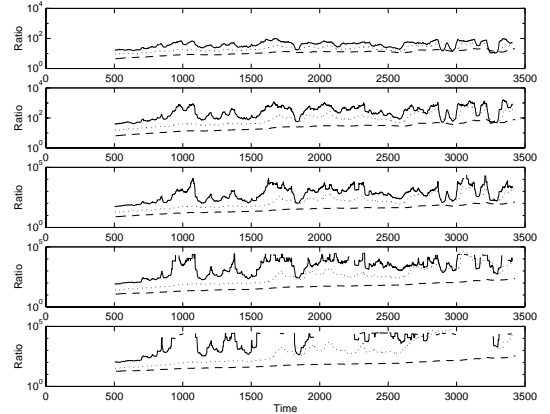


Figure 2: The ratio between candidate table size and active flow table size with decl, $X = 1, 3, 5, 10, 20$ starting from the top in the subplot, $T_C = T_F = 30, 60, 120$ s

count threshold, but depends heavily on the candidate timeout T_C values. Third, compared to the candidate table size, the active flow table is small. The candidate table seems to be the “hot spot” for packet count classifier implementations. Finally, the flow candidate table suffers from additions or deletions, easily at the speeds of thousands per second, thus forming a possible performance bottleneck in the packet count classifier.

5. Acknowledgments

The authors thank Mr. Markus Peuhkuri for helping with the traces used in this work. This work was in part sponsored by the IMELIO-project in the Networking laboratory.

References

- [1] H. Che, S.-Q. Li, and A. Lin. Adaptive resource management for flow-based IP/ATM hybrid switching systems. *IEEE/ACM Transactions on Networking*, 6(5):544–557, October 1998.
- [2] V. P. Kumar, T. Lakshman, and D. Stiliadis. Beyond best effort: Router architectures for the differentiated services of tomorrow’s internet. *IEEE Communications Magazine*, 36(5):152–164, May 1998.
- [3] S. Lin and N. McKeown. A simulation study of IP switching. In *ACM SIGCOMM '97*, 1997.
- [4] P. Newman, G. Minshall, and T. L. Lyon. IP switching – ATM under IP. *IEEE/ACM Transactions on Networking*, 6(2):117–129, April 1998.
- [5] T. Worster and A. Doria. Levels of aggregation in flow switching networks. In *Electronics Industries Forum*. IEEE, 1997.