

# Evaluation of a Fast and Accurate Flow-Counting Algorithm for IP Networks

Shan Zhu, Satoru Ohta (Toyama Prefectural University)

## 1. Introduction

An important aspect of IP network management is to count the number of flows. For this purpose, the authors have proposed a fast and accurate flow-counting algorithm [1]. This paper reports the performance of the algorithm for real-world network data and estimates its scalability.

## 2. Problem Description

The authors' algorithm [1] is based on the timestamp vector (TV) algorithm [2], which utilizes a TV and a hash function. The algorithm inputs the flow identifier of an arriving packet to the hash function. The timestamp is stored in the TV element, whose index is the hash output. Then, the number of flows is calculated by the recently updated elements. Based on this procedure, the authors' method achieves smaller computational time by using the following techniques:

- Multiple counters that eliminate the vector scanning process, which is very time-consuming, and
- Packet sampling.

In addition, the algorithm provides better accuracy by utilizing TCP FIN/RST messages.

The effectiveness of the algorithm was confirmed by artificially generating traffic on a small experimental network. However, it is not known whether the algorithm is fast enough to process the traffic of a real-world network. It is also unclear how the algorithm scales for high-speed links. The purpose of this study is to clarify these points.

## 3. Evaluation Method

The authors' algorithm was implemented with the use of the `pcap` library, which enables the program to read publicly available real-world network data. As real-world data, this study employed the data set distributed by the MAWI working group of the WIDE project [3]. The data set includes files gathered at various times at six sample points. Among these files, we examined the file of the traffic at 18:15, June 30, 2006 at sample point B, as well as the file of the traffic at 14:00, July 28, 2010 at sample point F. Hereafter, the former is referred to as file #1, while the latter is referred to as file #2.

The computational times taken to process files #1 and #2 were measured. The program of the authors' algorithm was executed on a Linux PC (Intel Core 2, 1.80GHz), and the time was measured by the `GNU time` command. For comparison purposes, the computational time was also measured for the original

TV algorithm [2].

The computational time greatly depends on the choice of the TV size. For many flows, the TV size must be large in order to decrease estimation errors. For file #2, it was found that the maximum number of concurrent flows is greater than  $3.5 \times 10^4$ . To estimate these flows accurately, the TV size was set to 1000003. The frequency of flow-number estimation also affects the computational time. In this study, the flow number was estimated every 1 s.

Figure 1 shows the computational time obtained by the authors' algorithm, as well as that obtained by the original TV algorithm. The figure clearly shows the advantage of the authors' method. For file #1, the computation time taken by using the authors' method is 57% smaller than that by using the original TV algorithm. For file #2, the authors' method was less effective. Nevertheless, the computational time is smaller for the authors' method than for the original TV algorithm, as well as for file #2.

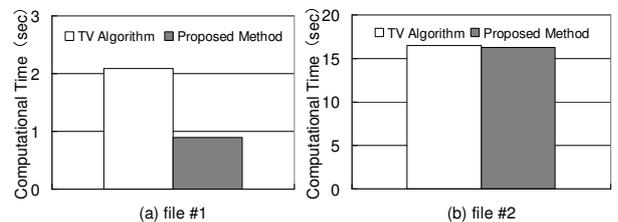


Figure 1: Comparison of computational times.

File #2 includes 36328883 packets. The algorithm processes these packets in 16.31 s. Thus, a packet was processed in  $0.45 \mu\text{s}$ . This is sufficiently small to process packets on 26 Gb/s link, if we assume that the average packet length is 1500 bytes. Therefore, the authors' algorithm will successfully count the number of flows on recent high-speed links.

## 4. Conclusion

The flow-counting algorithm of [1] was examined for real-world network traffic and was shown to work for high-capacity links whose bit rate exceeds 20 Gb/s.

## References

- [1] S. Zhu and S. Ohta, "Fast and accurate flow counting algorithm," NOMS 2010, pp.918-921, Osaka, Apr. 2010.
- [2] H.-A. Kim and D.R. O'Hallaron, "Counting network flows in real time," GLOBECOM 2003, pp.3888-3893, 2003.
- [3] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," USENIX 2000 Annual Technical Conference: FREENIX track, pp.263-270, Jun. 2000.