Method of Measuring VoIP Traffic Fluctuation with Selective sFlow

Atsushi Kobayashi and Katsuyasu Toyama
NTT Information Sharing Platform Laboratories
3-9-11, Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
{kobayashi.atsushi, toyama.katsuyasu}@lab.ntt.co.jp

Abstract

We present a method of measuring VoIP traffic fluctuation by using Netflow or sFlow. As the traffic of critical applications such as VoIP has increased in backbone networks, network operators need to measure the QoS performance of traffic in real networks more easily and passively. We have studied the delay and fluctuation of VoIP traffic in a real network and focused on traffic fluctuation that can be measured at one observation point. We evaluated the effectiveness of this method using selective sFlow, which has been implemented in a commercial router.

1. Introduction

Realtime critical application traffic such as VoIP (voice over Internet protocol) has recently increased in the Internet backbone network. In particular, VoIP traffic is reported to have several security vulnerabilities related to quality of service (QoS)[1], such as “QoS abuse” or “conversation degradation”. Moreover, DoS (denial of service) or DDoS (distributed denial of service) attacks on an enterprise network may consume a large amount of bandwidth and cause QoS degradation.

On the one hand, PSAMP[2], which exports on a per-packet basis, is being developed in IETF as a QoS performance measurement method. In this method, observed packets are selected based on a hash function, and traffic attributes with the timestamp of the selected packet are exported from a router to one or more collectors. At the collector side, the timestamps of the same packet are gathered and the collector can measure the delay and the jitter between exported routers.

These days, however, flow measurement solutions, such as sFlow[3] and NetFlow[4], are widely used in several networks. Although PSAMP is useful, it will be several years before it can be implemented in commercial routers and operators will need to replace all the routers in the backbone network by PSAMP-capable routers. We studied QoS performance in real networks and focused on a method for measuring traffic fluctuation by using sFlow and NetFlow. It can easily perform measurements at one observation point without the need to modify all the routers in the whole network. It can investigate a degradation point anywhere in the whole network domain if the customer inquires about network performance. Furthermore, measurement at the peering point between autonomous systems is enabled to distinguish poor performance of the network domain.

The organization of this paper is as follows. Section 2 describes the approach of our traffic fluctuation measurement method. The effectiveness of this method, determined using real traffic data, is described in Section 3. Section 4 describes the measurement method with router, and especially Section 5 introduces the selective sFlow function. Finally, we evaluate this method using selective sFlow in Section 6.

2. Measurement Method’s Approach

It is difficult to measure all of the traffic in its entirety, so we must select some specific traffic by filtering and sampling. To measure the traffic performance, a target RTP session is filtered by the conditions of source, destination address, and port number, just like an access list. And then these selected packets are sampled systematically at a rate of \(1 \text{ in } m\), not randomly.

On the collector side, we can measure the interval \(\Delta\) of exported packets. This interval is measured as the difference between arrival timestamps of a continuous
stream of exported packets. So, we assume that the variance of the interval \(\text{Var} [\Delta] \) has already been measured. How to obtain the variance of each packet \(\text{Var} [\delta] \) is discussed below.

Let \( \delta_i \) be the arrival interval of RTP packet \( i \). \( \text{Var} [\Delta] \) is a function of \( \delta_i \):

\[
\text{Var} [\Delta] = \text{Var} \left[ \sum_{i=1}^{m} \delta_i \right].
\]

It is also given by the covariance of the arrival intervals of RTP packets \( i \) and \( j \).

\[
= \sum_{i=1}^{m} \text{Var} [\delta_i] + 2 \sum_{i<j} \text{Cov} [\delta_i, \delta_j]
= m \text{Var} [\delta] + 2 \sum_{j=1}^{m-1} (m-j) \text{Cov} [\delta_i, \delta_{i+j}]
\]

Thus, \( \text{Var} [\Delta] \) is given by \( \text{Var} [\delta] \) and the autocorrelation \( \rho_{\delta} (j) \) of \( \delta \) with lag \( j \).

\[
= \text{Var} [\delta] \left( m + 2 \sum_{j=1}^{m-1} (m-j) \rho_{\delta} (j) \right)
\]

If we assume that \( \rho_{\delta} (j) \) is nearly 0, then \( \text{Var} [\delta] \) is given by \( \text{Var} [\Delta] / m \), which can be calculated simply and easily. In Section 3, we show whether or not \( \rho_{\delta} (j) \) can be ignored based on data measured in a real network.

3. Effectiveness in a Real Network

We evaluated the variance of receiving interval \( \text{Var} [\delta] \) and the correlation \( \rho_{\delta} (1) \) in a real network.

The times at which measurements were made in two end sites were synchronized exactly: one side sent packets at 20-ms intervals, which is the usual interval of a G.711 codec, and the other side received them and measured the delay of each packet. During a period about 900 s, we measured the delay of each packet and repeated the procedure 30 times.

![Figure 1. Comparison of estimated and directly measured values](image)

The results showed that \( \rho_{\delta} (1) \) was in the range from -0.05 to 0.10. If we assume that \( \rho_{\delta} (1) \) is 0, then the relationship between \( \text{Var} [\Delta] / m \) and \( \text{Var} [\delta] \) is as shown in Figure 1, which shows the case of \( m = 4 \).

We also recognized that the estimated value was close to the directly measured value in several cases if \( \rho_{\delta} (1) \) was ignored. Thus, this method enables us to grasp the performance status of some specific traffic.

4. Measurement Method with Router

Our approach uses the filtering and sampling functions in a router. In the case of NetFlow, the filtering function has been already implemented in Cisco routers[5]. After the filtering, these systematically sampled packets are exported with a timestamp that represents the time when they reached this router. For the router to export on a per-packet basis rather than on a per-flow basis, an inactive timer should be configured as 0. At the collector, “first switched” or “last switched” information elements should be examined in the measurement.

Although sFlow can export on a per-packet basis, the sampling method is only random sampling, and the timestamp, which is included only in the sFlow Header, is put in the uptime field. Originally, this timestamp in the sFlow Header meant the time when the sFlow agent exported the sFlow datagram from this router. The performance of the sFlow agent is higher if the agent is implemented in an ASIC or dedicated CPU, this time seems to mean the time when the last packet reached this router. Moreover, we can recognize the number of flow samples in one sFlow datagram, because the RTP packets in the same session are generally the same length. In fact, the number of flow samples in one sFlow datagram means sampling rate \( m \). In that case, this router should disable the counter sampling function and configure the sampling rate as 1.

In sFlow, we can measure not only the traffic fluctuation but also traffic loss by counting the sequence number in RTP packets. This is more useful.

We have implemented selective sFlow in an commercial router. This function is described in the next Section.

5. Selective sFlow

Selective sFlow filters the observed packets before sampling. It enables us to monitor only important traffic in detail. Moreover, it enables us to investigate the behavior of specific packets after indications of suspicious traffic. We implemented this function in commercial routers with 10-Gbps interfaces. The architecture of selective sFlow is shown in Figure 2. The filtering parameters that are configured include
addresses and port numbers of the source and destination. In this way, we can distinguish a specific RTP session.

Prior to this examination, we evaluated whether this function could select specific packets from a large amount of network traffic, such as a 10-Gbps interface, without excessive overload. We found that it could. In future, it could solve several problems in the backbone network and might be recognized as being useful for the next-generation backbone network.

**Figure 2. Selective sFlow implementation**

**6. Effectiveness of Using Selective sFlow**

We evaluated the effectiveness of our method using selective sFlow, as shown in Figure 2, by using dummynet [6] on a PC bridge as an experimental network environment emulating a real network. The delay $d_i$ of packet $i$ changed dynamically as

$$d_i = \rho_d(l) \times (d_{i-1} - c) + (1 - \rho_d(l)) \times e \times w + c$$

The autocorrelation $\rho_d(l)$ value of delay with lag 1 was set in the range from 0.85 to 0.95, which was chosen based on the results of measurements in the real network. Here, $e$ means the white noise, $c$ is the delay, which was set to 200 ms, and $w$ is the fluctuation width of the delay. The experimental network is shown in Figure 3.

The traffic generator sent emulated RTP packets for 20 ms. Then, the delay of each packet was measured when it was received via the router. The router filtered the specific RTP session when the observed packets reached the sFlow-enabled port, and these packets were forwarded to sFlow Agent. The sFlow Agent created a sFlow datagram that included the specific flow samples and uptime in part of the header. At the sFlow collector, the values of uptime of each sFlow datagram were measured and calculated as $Var[\Delta]/m$. The behavior of $\Delta$ against elapsed time for the case where we set $\rho_d(1)$ to 1, which means no traffic fluctuation, is shown in Figure 4.

**Figure 3. Experimental network environment**

**Figure 4. Behavior of the interval between contiguous uptime, which is included in sFlow Header**

In G.711-type codecs, RTP packets with a frame length of 214 bytes are generally sent for a 20-ms period. If we configure the value of sFlowMaximumDatagramSize to be 1400 bytes and the value of sFlowMaximumHeaderSize to be 256 bytes, then one sFlow datagram can usually include 4 RTP packets as a flow sample. That means that interval between contiguous uptime is close to 80 ms. If it is not 80 ms, the number of RTP packets included in one sFlow datagram is not 4. We presume that this is caused by the fluctuation of the sFlow agent process. This means it is difficult for this method to produce an exact value. However, it might bring an effective result on some level. The behaviors of $\delta$ and $\Delta$ against elapsed time in the case of emulating traffic fluctuation are shown in Figures 5 and 6, respectively. Comparing these two figures, we can recognize that the traffic fluctuation of the targeted packets causes the fluctuation of interval $\Delta$.

As a summary, the results of several fluctuation patterns are shown in Figure 7. Here, the margin of error is larger if the variance of fluctuation is close to zero in the result of assuming all sFlow datagrams including 4 flow samples. Originally, the sFlow agent
process of this router has some fluctuation periodically as shown by Figure 4. This process fluctuation causes the margin of error. It might be caused by the effect of another process. So, we selected only sFlow datagrams including 4 flow samples to get rid of the fluctuation of this agent, and then calculate the variance precisely. In regard to this result, the true values of $Var[δ]$ and the values of $Var[Δ]/m$ are close to same value.

7. Conclusion and Future Work

We described our method of measuring VoIP traffic fluctuation by using selective sFlow. The results of our examination were not precise, especially when the fluctuation was small, but they enable us to grasp the rough status of performance. And also, we can bring results more precise by getting rid of agent fluctuation. By using this measurement method, we can grasp rough status of performance. It is important to measure to any traffic in passively.

In the future, we will try to evaluate NetFlow and make the estimated value more precise. In sFlow, the value might become precise if the sFlow agent were separated from the router’s main processor and implemented in a dedicated processor module or if the sFlow agent were able to export a flow sample with timestamps as extended information.

Furthermore, this method becomes more flexible with the correlation of a SIP-proxy server and BGP route information. It can measure the QoS performance of any traffic by getting SDP information from a SIP-proxy, and then we can specify the router that passed targeted RTP packets based on bgp next-hop. After all, we can set filtering condition on appropriate selective sFlow router.

Acknowledgments

This study was supported by the Ministry of Internal Affairs and Communications of Japan.

References