Available online at http://www.journalcra.com

International Journal of Current Research
Vol.3, Issue, 4, pp.206-209, April, 2011

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

AN END-TO-END APPROACH FOR PACKET FORWARDING PRIORITIZATION

Tholkappiya, PFU. and Padmapriya, A.

Department Of Computer Science and Engineering, Alagappa University, Karaikudi, Tamil Nadu, India.

ARTICLE INFO

Article History:
Received 27th January, 2011
Received in revised form
24th February, 2011
Accepted 12th March, 2011
Published online 17th April, 2011

Key word:
PFP,
Network measurements,
Performance of applications,
POPI.

ABSTRACT

Packet Forwarding Prioritization (PFP) in routers is one of the mechanisms commonly available to network operators. PFP can have a significant impact on the accuracy of network measurements; the performance of applications and the effectiveness of network troubleshooting procedures. we present an end-to-end approach for PFP inference and its associated tool, POPI. This is the attempt to infer router packet forwarding priority through end-to-end measurement. POPI enables users to discover such network policies through measurements of packet losses of different packet types. We evaluated our approach via statistical analysis, simulation and wide-area experimentation in PlanetLab. Besides, we compared POPI with the inference mechanisms through other metrics such as packet reordering [called out-of-order (OOO)]. OOO is unable to find many priority paths such as those implemented via traffic policing.

INTRODUCTION

The Internet was designed with no gatekeepers over new content or services. A lightweight but enforceable neutrality rule is needed to ensure that the Internet continues to thrive. Packet forwarding prioritization has been available in off-the shelf routers (Cheng, 2007). Network operators have come to rely on these mechanisms for managing their networks, for example as a way of rate limiting certain classes of applications (e.g., peer-to-peer) (Rubenstein *et al.,* 2003). PFP can have a significant impact on the performance of applications, on the accuracy of measurement tools' output, and on the effectiveness of network troubleshooting procedures. Despite its potential impact, users, developers and most other network administrators have no information of such settings nor ways to procure it. There are a couple of challenges for designing and implementing POPI. First, background traffic fluctuations can severely affect the end-to-end inference accuracy of router properties. Second, probe traffic of a relatively large packet bursts are neither independent nor strong correlated. Third, we want to measure more than two packet types at the same time, so simply determining whether they are treated differently is not enough. To overcome these challenges, POPI takes the following three steps to infer packet forwarding priority inference. First, it sends a relatively large amount of traffic to temporarily saturate the bottleneck traffic class capacity, which gives POPI better resistance against background traffic fluctuations. Second, we apply a robust *nonparametric* method based on the *ranks* instead of pure loss rates. Thirdly, we assign a rank-based metric to each packet type and use a hierarchical clustering method to group them when there are more than two packet types.

We compared POPI with the inference mechanisms based on other metrics with less overhead such as packet reordering (called *out-of-order (OOO)*) OOO is unable to find many priority paths such as those implemented via traffic policing. On the other hand, it can detect existence of the mechanisms which induce delay differences among packet types such as slow processing path in the router and port-based load sharing.

RELATED WORK

To the best of our knowledge, this is the attempt to infer router packet-forwarding priority through end-to-end measurement. Perhaps the efforts most closely related to this work are those identifying shared congestion. Such efforts try to determine whether *two* congested flows are correlated and share a common congested queue along their paths. If we consider the flows of different packet types along a same path, our problem becomes to identify whether these flows do not share a common congested queue. While both problems are related clearly, we usually need to simultaneously consider a much larger number of packet types (e.g., 26 packet types in the Planet Lab experiment). Note that the correlation based method used for shared congestion identification methods requires back-to-back probing which, in our case, translates into pairs probing for packet types. In addition, those efforts focused on flows which experience congestion (ignoring uncongested ones), so their probe traffic rate is low and not bursty. To identify packet forwarding prioritization in routers, one must send relatively large amounts of traffic to temporarily force packet drops (by saturating the link). Thus, for better scalability and accuracy, our problem requires different measurement and statistical interference methods.

Kuzmanovic and Knightly proposed a framework for enabling network clients to measure a system's multiclass

---

***Corresponding author:** tholkamca@gmail.com mechanisms and parameters (Kuzmanovic and Knightly, 2001). The basic idea is similar to ours, i.e., to inject multiclass traffic into the system and use a statistical method to infer its scheduling types and parameters based on the output. PFP inference also has some goals in common with efforts on network tomography (Bu *et al.,* 2002). However, unlike in network tomography where loss information and topology information are combined to infer link losses.

INFERRING PACKET-FORWARDING PRIORITY

Background on Priority Mechanisms

Network administrators can enforce priority/link-sharing mechanisms in a router by defining a traffic class (usually IP protocol and TCP/UDP port number) and associating with it a particular queuing/scheduling mechanism (Jain and Dovrolis, 2003). Some of the commonly available mechanisms are as follows.

- *Priority Queuing (PQ).* This allows users to assign arbitrarily defined packet classes to queues with different priorities. Since queues are served based on their priority, this allows specified packet types to be always sent before other packet types.
- *Proportional Share Scheduling (PSS).* With PSS each traffic class is given a weight. Bandwidth is allocated to classes in proportion to their respective weights. There is no strict priority difference between classes.
- *Policing.* This restricts the maximum rate of a traffic class Traffic that exceeds the rate parameters is usually dropped. The traffic class cannot borrow unused bandwidth from others. Only the first mechanism sets absolute priorities between traffic classes. There is no absolute priority difference between the other two classes.

Choosing Inference Metric

Three basic end-to-end performance metrics, loss, delay and out-of-order, can all be used as inference metrics. This is because these metrics of different packet types can become different when a router is configured to treat them differently. Consider a *PQ* of two priorities, where the high priority queue is always served first. Low priority packets will experience larger loss rates and longer queueing delays than the high priority packets. Besides, a low priority packet may arrive earlier than a high priority packet but leave after it while the contrary will never happen. The reordering events between them are asymmetric. Here, the loss, delay, and reordering can all be used as a metric to infer priority settings. we'll use the loss, delay and Out-Of-Order (OOO) based method to name the inference methods based on these metrics.

The probe overhead of packet loss metric is larger than the other two.

Obviously, loss rates difference will not become evident until the associated link (or a sub link for a traffic class) is saturated and begins to drop packets. This simple observation defines the basis of loss-based inference approach: In order to reveal packet-forwarding priorities, one needs to saturate the path available bandwidth for a given class to produce loss rates difference among different classes.

Loss difference can be observed for all kinds of QoS mechanisms while the other two cannot.

Although using delay and reordering metrics can result in less probe overhead, they can not detect certain router QoS mechanisms simply because those mechanisms do not generate different delays at all. We found that many multi priority paths (MPPs) detected by the loss-based method could not be detected the other two methods.

Packet delay difference can be caused by many other mechanisms than QoS.

The root cause of packet reordering is the existence of parallel packet forwarding paths. Such paths can be in a router, parallel links between two routers, or different routes over several hops. When packets are split to these parallel paths according to their packet types and these paths have different delays, we'll observe asymmetric packet reordering and delay differences among different packet types.

Challenges for POPI

In designing and implementing POPI we addressed several interesting challenges.

The accuracy of end-to-end inference of router properties can be severely affected by background traffic fluctuations.

Clearly, if one's probing introduces relatively small additional traffic, whether the link is saturated or not depends solely on the amount of background traffic. To make our approach more resistant to background traffic fluctuations we opt for sending relatively large amount of traffic to temporarily saturate bottleneck traffic class capacity, which increases the probability of observing loss rates difference. To note, the sender may not be able to saturate the bottleneck link due to limited resources, which is an inherit limitation of this method.

Probe traffic of a relative large packet bursts are neither independent nor strongly correlated.

Once the loss rate for each packet type is obtained, we need to determine whether the loss rates difference among them is large enough to conclude that they are treated differently. When packet losses can be described with a good mathematical model, e.g., independent and identical distribution process, we can determine if the loss rates of different packet types were evidently different or not by comparing all, the loss rate of packet type, using parametric statistical methods.

Grouping is needed for multiple packet types probing.

If we only probe two packet types at one time, simply determining whether they are treated differently is enough. However, we sometimes probe more than two packet types and need to group them based on their priorities. Here, we assign a rank-based metric to each packet type and use hierarchical clustering method to group them.

DESIGN OF POPI

Probing the Path

POPI sends several bursts from a source to a destination. The interval between bursts is . Each burst consists of rounds, in which packets, one for each packet type studied, are interleaved in random order. So, there are back-to-back packets in each burst.

Deriving Ranks

For every burst, loss rate ranks are computed by first sorting packet types in ascending order according to their packet loss rates in that burst and then assigning ranks in order, i.e., the packet type with the largest loss rate has rank 1, the one with the second largest loss rate has rank 2 and etc.1 Similar to packet loss rates, due to randomness of packet losses, the ranks of different packet types are like random arrangements over the all bursts when the packet types are treated equally.

Partitioning Based on Ranks

Every packet burst can be regarded as an observation. Identifying whether there is consistent difference among ranks over observations is a well-known statistical problem called problem of rankings. Therefore, we proposed to use Average Normalized Ranks (ANR) to group packet types when there is consistent difference. The ANR is the average of the ranks for a packet type over all bursts.

Calculate ANR. Let $r_i^m = (1, 2, \ldots)$ denote the rank for packet type *i* in *m*th burst. The Normalized Rank NR_i^m is r_i^m / k. The range of NR_i^m is between and 1/*k*. The ANR_i for packet type *i* is

$$\mathrm{ANR}_i = \left(\sum_{m=1}^{n_b} NR_i^m \right) / n_b.$$

When k_i packets are in a same class *j*, the range of this class $(R = ANR_{\max} - ANR_{\min})$ for n_b bursts at confidence level 1-α is

$$\theta_{1-\alpha,k_j,n_b,k} = Q_{1-\alpha,k_j} \times \sqrt{k_j^2} - 1/k\sqrt{12 n_b} \qquad (2)$$

where $Q_{1-\alpha,k_j}$ is the 100(1-α) percentile of the range (of k_j i.i.d. standard normals) distribution.

EVALUATION WITH NS-2 SIMULATION

We implemented POPI in NS-2.We use a dumbbell topology. The output queue $R0 \rightarrow R1$ of is configured with or using Class-Based Queueing (CBQ). If not specified, is configured with PQ or PSS of two priority classes. Class 1 is the high priority class and class 2 is the low priority class. The queue length of high and low priority queues are 20 and 60, respectively. In the experiment, the size of both POPI packets and background packets are 1000 bytes. The probe traffic rate is 100 Mbps.

Table 1. 26 Packet Types Considered For Planetlab Experiments

PROT	Type/Port Number
ICMP	ICMP_ECHO
TCP	20, 21, 23, 110, 179, 443 (well-known app) 1214, 4661, 4662, 4663, 6346, 6347, 6881 (P2P applications) 161, 1000, 12432, 38523, 57845 (random)
UDP	110, 179, 161 (SNMP) 1000, 12432, 38523, 57845 (random)

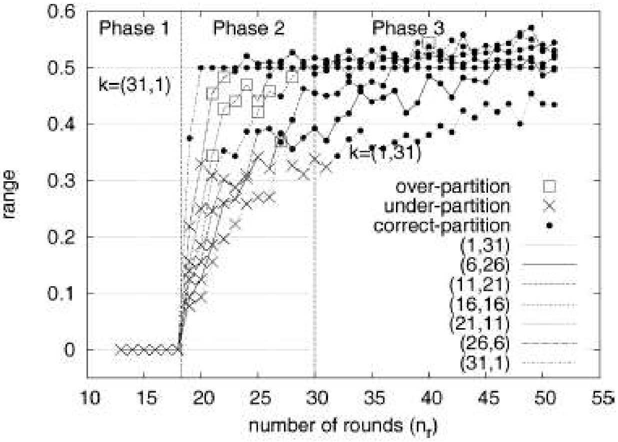

Fig.1 .Partition results for priority queuing.Available bandwidth=90 Mb.

The Effects of Probe Burst Size

The background traffic is 10 Mbps *Gamma* with shape 0.5, which consists of equal share of high and low priority traffic. The low priority packet types begin to experience losses, but since the losses are insufficient for POPI to properly classify the packets, the partition results are still incorrect. This phase can be further divided into two sub phases based on the amount of losses generated.

The Effects of Background Traffic Rate

We show the performance of using $n_r = 40$ under different background traffic rates, i.e., 10, 20, 40, 80, 90 Mbps. The background traffic consists of equal share of high and low priority traffic. For each rate, we let $k_1 = 1, 2, 3, \ldots, 31$. Once the background traffic rate and (k_1, k_2) are fixed, we run 10 simulations. Altogether, 1550 simulations are performed and all of them are correctly partitioned.

Proportional Share Scheduling

We use PSS and define two traffic classes in . The bandwidth allocation ratios for the two classes are 0.2 and 0.8, respectively. The background traffic rates are 2 and 8 Mbps. We let $n_r = 40$ and $k_1 = 1, 2, 3, \ldots, 31$. For every (k_1, k_2) combination, we run 10 simulations. For PSS, whether a class experiences packet drops at the router depends on its input traffic rate and allocated bandwidth.

PLANETLAB EXPERIMENTS

The sender sends multiple packet types toward the receiver. The receiver feedbacks certain information of every received packet to the sender, which is used by the sender to measure the end-to-end losses and reordering events along the path. HHP mode is used to locate the configured router or device by measuring the losses and reordering events to every router on the path towards the destination

Packet Types Tested

While it may seem necessary to test all packet types of different protocol/port number combinations to validate our approach, in practice there is only a small number of packet types that network administrators may want to treat differently. We selected 26 packet types as listed in Table IV. For UDP and TCP packets, 30002 are used as the destination port, because it is very unlikely that ISPs will set an explicit priority policy based on it. The port numbers listed in Table IV are used as source ports to measure the source port based priority policy.

Validation Method

Since it is very difficult to get the actual router configurations on the path, we use *HHP* method to locate the spot of difference. We find its corresponding organization using who is data base, and then send e-mails to the related technical support for validation. To minimize the traffic sent to routers, we usually chose three to six packet types according to the group pattern and set to n_T to 10.

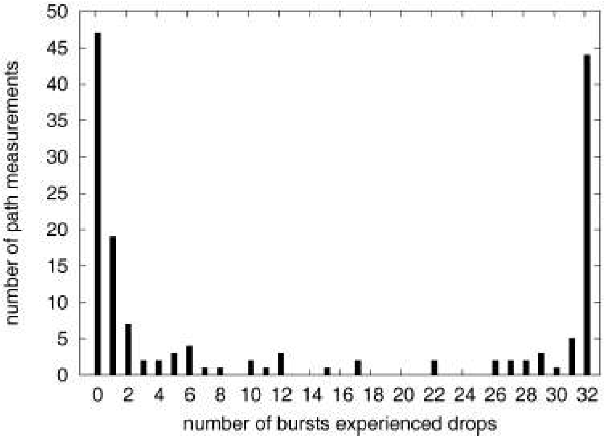

Fig.2 Histogram of number of lossy bursts for 156 probes.

First, we check how packet drops are distributed among bursts of a path measurement to see if there are any effects caused by background traffic fluctuations. Fig. 6 shows the distribution of number of path measurements in which a certain number of bursts experienced network packet drops. We can see that for most of the path measurements, either all bursts experienced drops or no bursts experienced drops. This is accordance with our notion that traffic remains relatively stable within a short period of time.

COMPARING PRIORITY INFERENCE WITH DIFFERENT METRICS

We compare the inference results of the loss-based and OOO-based method using PlanetLab experiments. We do not use packet delay because the reordering metric is more robust than the delay metric although they both reflect the packet delay differences. When the delay variation generated by the nonconfigured devices is large, a packet with a shorter delay at the configured box can have a larger end-to-end delay than a packet with a larger delay at the configured box. Hence, the delay differences between different packet types introduced by the configured box are overwhelmed by the large delay variation introduced by the non configured devices along the path. Large delay variation can often be observed for congested routers. OOO-based method is generally more accurate than the delay-based method.

Methodology

We performed the loss-based and OOO-based inference for every probe. We also ran *HHP* method for every OOO-based Multi-Group Path (OMGP) and Loss based Multi-Group Path (LMGP), and sent e-mails to the related network operators for confirmation. The OOO-based method is almost the same as the loss-based method except that we use reordering ranks instead of loss ranks to perform group partition. The reordering rank of packet type *i* is derived from its Packet Reordering Ratio (PRR_i), which is the fraction of reordered packets in all received packets of this packet type.

CONCLUSION

Here we see that POPI, an end-to-end priority inference tool, is able to accurately infer the router's packet forwarding priority. In the PlanetLab experiments, the loss-based method detected several multipriority paths in the Internet. In searching for a method with less probe overhead than the loss-based method, we used packet reordering and delay as the inference metrics and found they were not as effective as loss in detecting packet forwarding priorities.To avoid the drawback of the present system, we introduce the concept of Multiple Routing Configuration (MRC).MRC is based on keeping additional routing information in the routers,and allows packet forwarding to continue on an alternative output link immediately after the detection of a failure.

REFERENCES

Augustin, B., Friedman, T. and Teixeira, R. 2007. "Measuring load-balanced paths in the Internet," in *Proc. IMC*, 149–160.

Bu, T., Duffield, N., Presti, F. and Towsley, D. 2002. Network tomography on general topologies," in *Proc. ACM SIGMETRICS*, pp. 21–30.

Cheng, J. 2007. Evidence mounts that Comcast is targeting Bittorrenttraffic. Available:http://arstechnica.com/ news. ars/post/20071019-evidence-mounts-that-comcast-is-targeting-bittorrent- traffic.html

Jain, M. and Dovrolis, C. 2003. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput," *IEEE/ACM Trans. Netw.*, 11(4): 537–549.

Kuzmanovic, A. and Knightly, E. W. 2001. Measuring service in multiclass networks," in *Proc. IEEE INFOCOM*, pp. 1281–1289.

Lu, G., Chen, Y., Birrer, S., Bustamante, F. E., Cheung, C. Y. and Li, X. 2007. End-to-end inference of router packet forwarding priority," in *Proc. IEEE INFOCOM*, 2007, pp. 1784–1792.

Rubenstein, D., Kurose, J. and Towsley, D. 2002. Detecting shared congestion of flows via end-to-end measurement," *IEEE/ACM Trans. Netw.*, 10(3): 381–395.