Using Route Probing to Derive Link Traffic Load with Edge-based Measurements

Zhao Guofeng, Tang Hong, Zhang Yi, Gu Shangyu

Chongqing University of Post and Telecommunication, Chongqing, China, 400065
zhaoguof@cqupt.edu.cn

Abstract. Obtaining traffic load on internal links of the network is crucial for network management and control. Though collecting can be available on each link, such as applying traditional SNMP scheme, the approach would be expensive because it may cause heavy processing load and sharply degrade the throughput of the core routers in high-speed IP backbone. Then monitoring merely at the edge and estimating traffic in the core provides a good alternative way for overcoming such functionality limitations. In this paper, we explore a scheme on deriving internal link load of network with edge-based measurements. Contrast to collecting routing data from core routers that costs much, we propose a route probing method based on hash sampling techniques and IP Measurement protocol between node-pairs. Based on statistical theory, we prove that our approach is effective and present the algorithm. Performance simulation results show the potential of our approach.

1 Introduction

Knowing the volume of traffic on each internal link is beneficial for network management and control. Basically, there exist three kinds of approaches based on passive measurement [1] that may observe link load traffic. In traditional IP-based networks, traffic is derived by per-link approach such that using simple network management protocol (SNMP) and remote monitoring (RMON) [2] mechanism. But this approach has the disadvantage that it will cause heavy processing load and sharply degrade the throughput of the core routers in high-speed IP backbone. So monitoring at ingress nodes and computing the link traffic load from these measurements provides another way to overcome such functionality limitations. With this notion, A.Feldmann et al. propose a flow-based measurement approach that traffic flows are measured only at the ingress and routing configuration are collected from routers [3]. But the approach has some handicaps: (1) difficult to measure each flow at the edge of backbone network since there may be simultaneously ten thousands of flows. (2) collected data sets are enormous and computation is time-consuming. (3) need of acquiring routing configuration from core routers costs great. Then the third type of traffic measurement is brought forward as direct observation. Trajectory sampling [4] is a method that pro-

vides an estimator of the path matrix using packet sampling technology. It doesn't need to know anything about network topology and routing information. It involves sampling packets that traverse each link within the network and regards the set of sampled packets as a representative of the overall traffic. However, selecting the exact hash sampling function to meet real world is too hard.

The main contribution of this work is to develop a scheme that link traffic load on each link of a measurement domain will be estimated with edge-based measurements and route probing results. We apply node-pair based measurements at ingress nodes without enabling measurements in the core of the network. In order to know how the traffic is routed, routing matrix is constructed with route probing that using hash-based packet sampling and applying IP measurement protocol (IPMP) [5] to transmit path information. Based on statistical theory, we prove that our approach is feasible. Further, we propose an algorithm for link traffic computation.

2 Our Model for Link Traffic Measurement

In our model, we measure traffic at edge routers and transmit measurement data to a server named NCU(Network Collector Unit) where have traffic computation periodically as shown in Figure 1.

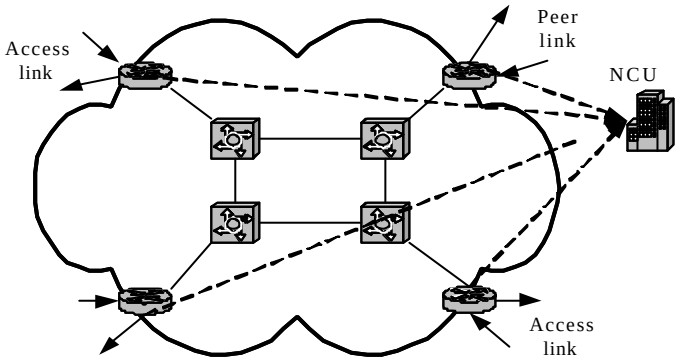

Fig.1. Our model for link traffic measurement

At edge nodes, we measure aggregate traffic on node pairs. Edge routers send and receive IPMP-based route probing packets and extract path information from those packets. Such path information is collected by NCU to construct routing matrix.

Definition 1. Assume a direct graph $D=(V,E)$, $|V|=n$, $|E|=m$. Name $Y=(y_1,y_2,\cdots,y_i,\cdots,y_m)$ link traffic vector, where y_i denotes traffic on link e_i. Let $X=(x_1,x_2,\cdots,x_j,\cdots,x_L)$ be SD(Source-destination) measurement vector, where x_j denotes measured traffic on path over j^{th} SD pair and L denotes number of SD pairs within an interval T.

Definition 2. Let A be a routing likelihood matrix with scale $m\times L$. For link e_i and j^{th} SD pair, a_{ij} denotes the likelihood of traffic on j^{th} SD pair traverse over link e_i.

Then according to the following equation, we can obtain vector *Y*.

$$Y^T = AX^T \tag{1}$$

Obviously, edge node-pair based measurement can obtain vector *X* easily. Then the main problem is that routing likelihood matrix *A* should be constructed before link traffic can be derived from equation (1).

3 Route Probing

The intention of route probing rests on the idea that routers process probing packets in the same way as other packets. Based on IPMP, path can be recorded in the probing packet when it traverses the network. So relation between links and routes over SD pairs will be inferred from probing packets sent and received.

The IP Measurement Protocol (IPMP) is based on packet-probes. It supports forward and reverse path measurements of a single packet. The protocol has been designed so measurement packets can be processed with approximately the same level of computation as needed for IP packet forwarding. IPMP is implemented in AMP measurement system [6] developed by National Laboratory of Applied Networks Research.

3.1 Route Probing Based on Hash Sampling

In this section, first we give two theorems for route probing scheme. Secondly, we present a framework for hash-based probing system.

Theorem 1. For routing likelihood matrix *A*, with random and independent route probing, element a_{ij} has standard deviation

$$s = \frac{\sqrt{n_{ij}(n_j - n_{ij})}}{n_j} \tag{2}$$

where n_j denotes probing packets sent over j^{th} SD pair and n_{ij} denotes probing packets traveled through the link e_i.

Proof: Since route probing packets are sent out to their destination in random and independent way and processed in the same way as common packets on routers. Path records can be extracted from these probing packets, so accurate routing information in the network can be revealed after enough probing.

Suppose within measurement period *T*, source node *s* has sent n_j number of probing packets over j^{th} SD pair. On link e_i, we obtain n_{ij} number of probing packets traveled through the link.

Then the routing likelihood of link e_i on paths over j^{th} SD pair is given by

$$a_{ij} = \frac{n_{ij}}{n_j} \tag{3}$$

With large enough number of probing, obviously, a_{ij} has a Bernoulli distribution. So its standard deviation has

$$s = \sqrt{\frac{a_{ij}(1 - a_{ij})}{n_j}} = \frac{\sqrt{n_{ij}(n_j - n_{ij})}}{n_j} \tag{4}$$

Thus give the proof of Theorem 1.

At each edge node, route probing is performed periodically as shown in Fig.2. Sampled packets are constructed as probing packets based on IPMP and sent out to destination.

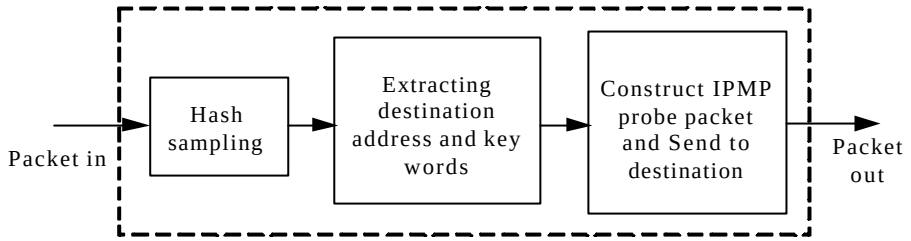

Fig.2. Framework for route probing at edge router

When probing system receives a packet, it extracts specific bits from the packet to make a key including the packet's destination IP address. Then the key is matched to a predefined key mask. If they do not match, the packet is not chosen for sampling. If the packet is selected for sampling, a new IPMP request packet will be constructed as a probe packet using the sampled packet's destination IP address as its destination address. Then the new packet is sent out to its destination as a common IP packet in the network.

Definition 3. For a packet x, $f(x)$ denotes the key that made up of specific bits extracted from x and $f(m)$ the mask.

Hash-based sampling satisfies

$$h(f(x)) = \begin{cases} 1 & if\ f(x) = f(m) \\ 0 & otherwise \end{cases} \tag{5}$$

When $h(f(x)) = 1$, the packet is selected for sampling and a probe packet will be constructed. The length of mask bits determines sampling probability. Suppose a mask length *m*, there exist $M = 2^m$ different values for sampling and $p = 1/M$ is the probability of sampling a packet. Selecting adequate value of *m* may obtain the expected sampling results. We make *m*=1024 for performance simulations in section 4.

Theorem 2. For any link e_i, within a measurement period *T*, with random and independent route probing, the relation between measurement error and route probing packet sampling probability *p* satisfies

$$s_i \le \frac{1}{2\sqrt{n \cdot p}} + \frac{n \cdot p}{n_i} \tag{6}$$

where *n* denotes all traffic(packets) traveled over *network* and n_i denotes traffic(packets) traveled through the link e_i.

Proof: Within a measurement period T, with random and independent route probing, measurement error on link e_i contains two parts, one for influence of probing packets on background traffic denotes s_{back} and the other for route probing error s_{prob}.

So, measurement error sums that

$$s = s_{back} + s_{prob} \tag{7}$$

i. Suppose n_c number of packets are inserted into network for route probing, then packet sampling probability p satisfies

$$p = \frac{n_c}{n} \tag{8}$$

For link e_i, the maximum of measurement error is

$$\max s^{i}_{back} = \frac{n_c}{n_i} = \frac{n \cdot p}{n_i} \tag{9}$$

ii. For route probing error, from equation (4) we have

$$s^{i}_{prob} = \sqrt{\frac{c}{n_c}} = \sqrt{\frac{c}{n \cdot p}} \tag{10}$$

where $c = a_i(1 - a_i)$. Then we have

$$\max(c) = 1/4, \quad 0 \le a_i \le 1 \tag{11}$$

So the following inequality satisfies

$$s^{i}_{prob} \le \frac{1}{2\sqrt{n \cdot p}} \tag{12}$$

Then with equation (7), equation (9) and equation (12), we obtain equation (6).

Thus give the proof of Theorem 2.

3.2 Algorithm

Now we present the algorithm for link load computation.

Step 1. To initialize link traffic vector and routing likelihood matrix, let Y=0 and A=0.

Step 2. To compute routing likelihood matrix based on route probing scheme, we obtain the likelihood of a_{ij} is n_{ij}/n_j.

Step 3. To derive load on link e_i, we have

$$y_i = \sum_{j=1}^{L} a_{ij} \cdot x_j \tag{13}$$

where y_i denotes traffic load on link e_i, x_j denotes measured traffic on path over j^{th} SD pair and L denotes number of SD pairs within an interval T.

Step 4. Repeat step 3 to compute traffic load on other links.

End.

The computation for routing likelihood matrix in step 2 of the algorithm is $O(mL)$. The maximum number of SD pairs has $\max(L) = n(n-1)$. Then we conclude the computation complexity of the algorithm is $O(mn^2)$ (m denotes links and n denotes nodes of network).

4 Performance Simulations

The main goal behind the simulation is to evaluate the performances of our model. The topology of simulation is shown in Fig.3. It includes five core routers and four access subnets as ingress.

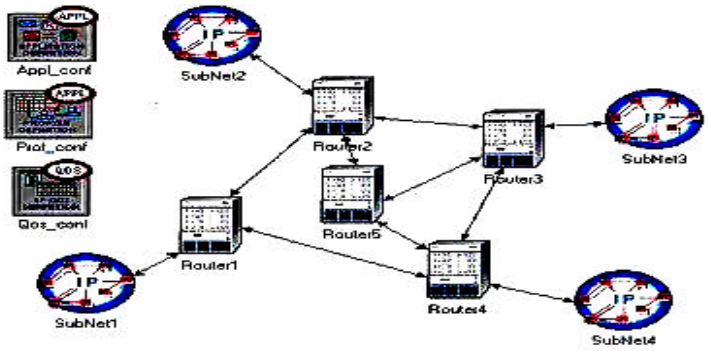

Fig.3. Topology of simulation network

For paper limitation, we just show simulation results of traffic on link Router2 to Router3. In the following figures, title including text as Traffic of Measure denotes throughput measured by our approach and title including text as point-to-point throughput denotes throughput measured by per-link approach. We show simulation results that the link works respectively in light-load as Fig.4 and Fig.5 shown, and heavy-load as Fig.6 and Fig.7 shown.

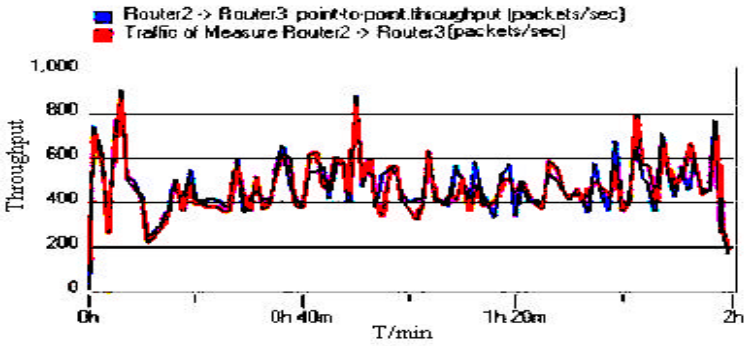

Fig.4 Throughput on link Router2 to Router3 measured by our approach and per-link approach simultaneously when the link works in light-load situation for two hours

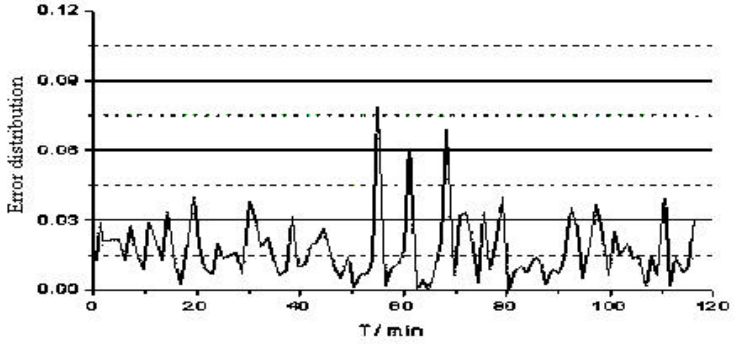

Fig.5 Error distribution between our approach and per-link approach when the link works in light-load situation

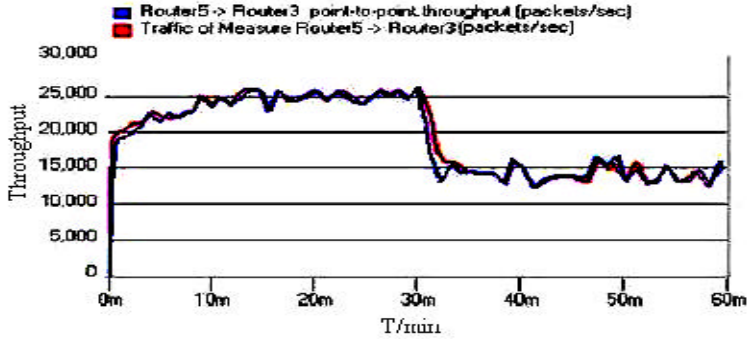

Fig.6 Throughput on link Router2 to Router3 measured by our approach and per-link approach simultaneously when the link works in heavy-load situation for two hours

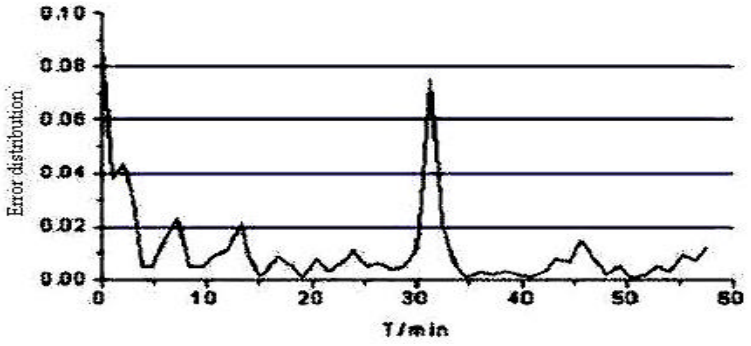

Fig.7 Error distribution between our approach and per-link approach when the link works in heavy-load situation

Investigating on simulation results concludes that: (1) results have little discrepancy between per-link approach and ours on link traffic measurement. It implies that

our approach is effective. (2) lower error is shown in light-load situation compared to heavy-load situation. We think it's mostly caused by route probing traffic because that makes a higher overhead when link traffic is light than heavy. (3) error will increase when traffic decreasing abruptly as shown in Fig.6 and Fig.7. We think it caused by route change that makes some traffic go to other path and transmit over this link no more. But our route probing has a delay on detecting such routing change.

5 Conclusion

The work is to explore a new edge-based link load traffic measurement problem and results several contributions: (1) a model for deriving link load when measurement performing only at edge without enabling in the core. (2) a scheme for route probing based on Hash based packet sampling and IPMP context. (3) an algorithm for deriving link traffic load.

Compared with the SNMP scheme, our approach causes few overhead in the core because measurement is only at the edge of the network. With flow-based method, since node-pair measurement involves aggregate flows, our scheme produces less computation. Moreover, our scheme can probe route at the edge while need not extract routing data from the core routers. And with direct observation, our approach doesn't require label buffers and sampling operations in the core.

6 Acknowledgement

This work is supported by Chunhui project funded by Ministry of Education, Nature Science Foundation of Chongqing and Special Fund on 4G-research of CQUPT.

References

1. Matthias Grossglauser and Jennifer Rexford, Passive Traffic Measurement for IP Operations, http://www.research.att.com/ ~jrex/papers/sfi.ps, March 2,2003
2. S. Waldbusser. Remote Network Monitoring Management Information Base, IETF RFC 2819, May 2000
3. Feldmann A, et al. Deriving traffic demands for operational IP networks: methodology and experience. Proceedings of ACM SIGCOMM'2000, 2000. 257-270.
4. N.G.Duffield et al. Trajectory Sampling for Direct Traffic Observation, ACM Computer Communication Review, vol.30, NO.4, Oct. 2000.
5. A. McGregor and M. Luckie. IP Measurement Protocol (IPMP), IETF draft: draft-mcgregor-ipmp-04.txt, Feb.2004.
6. A.J.McGregor, and H.W.Braun. Balancing cost and utility in active monitoring: The AMP example, In Proceedings of INET2000, 2000.