Network Path Monitoring

Buse Barcin Halis, Florian Wiedner*, Max Helm*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: ge23jod@mytum.de, wiedner@net.in.tum.de, helm@net.in.tum.de

***Abstract*—Network path monitoring is an important feature of modern networks. It enables to understand the behavior of the network. However, a network is a complex structure, and therefore it is a challenge to measure network characteristics from only the endpoints.**

Since the early days of the Internet, various network monitoring methods have been proposed. This paper focuses on methods for measuring network metrics such as packet loss, packet reordering, point of failure, round trip time, and bottleneck router buffer size. The differences in their implementation are highlighted, and some of their limitations are pointed out. It is concluded that OneProbe is a reliable method, but needs further development to measure more metrics.

***Index Terms*—network path monitoring, network metrics, network measuring methods**

1. Introduction

Networks are the foundation of many applications. When a problem occurs in the network, the state of the network directly affects the applications. Therefore, it is important to monitor networks and gain insights into the network state. This makes it possible to understand the behavior of the network, measure network performance, identify the problems of a network and find the causes for the problems. However, monitoring the network path is a difficult process. The network is complex, and when a fault occurs, it is difficult to determine what the problem is or where in the network the problem occurs. Therefore, we need accurate and efficient methods to monitor network paths.

Many methods have been proposed that focus on measuring network metrics [1]–[4]. This paper presents network monitoring methods that focus on the metrics of packet loss, packet reordering, point of failure, round trip time, and bottleneck router buffer size.

The rest of this paper is structured as follows: Section 2 provides background information. In Section 3, the detailed process of the methods for each presented metric is explained. Section 4 provides a comparison between the introduced methods, and Section 5 concludes the paper.

2. Background Information

In this section, network metrics are defined for which measurement methods are proposed in the next section.

Packet Loss: Loss of data packets during transmission on the network path from the source to the destination and non-arrival of the data packet at the destination [5].

Packet Reordering: The arrival of data packets at the destination not in the same order as they were sent from the source [5].

Failure Point: The link where a network problem occurs, such as packet loss or reordering [2].

Round Trip Time (RTT): Time interval between sending a packet from the source to the destination and receiving an acknowledgement for this packet at the source from the destination [1].

Buffer Size: The buffer is a storage area on a router where the data packets are temporarily stored. The buffer size indicates the capacity of this storage area [4].

3. Network Path Monitoring Methods

This section presents the methods categorized according to their measured metrics.

3.1. Packet Loss

Many methods have been developed to measure packet loss behavior in network paths. In this subsection, three of them are introduced, namely OneProbe, Tulip and Sting.

OneProbe: OneProbe is a TCP probing method for monitoring network paths and measuring network metrics, proposed by Luo et al. [1]. OneProbe uses TCP data probes. Each probe sent to the destination contains two TCP data packets as probe packets and triggers two new TCP data packets from the destination as response packets. The probe packets are used to inspect the forward path, while the response packets are used to inspect the reverse path. The probing process works as shown in Figure 1. The TCP data probe packets sent by OneProbe are declared as Cm|n and the TCP data response packets sent by the server are declared as Sm|n, where m corresponds to the sequence number and n to the acknowledgment number of a TCP data segment. Ŝm|n denotes a data retransmission.

There are 5 different possibilities for two packets on a path:

1) F0/R0: The server/OneProbe receives both probe/response packets in correct order.
2) FR/RR: The server/OneProbe receives both probe/response packets in reverse order.
3) F1/R1: The server/OneProbe receives only the second probe/response packet, the first packet is lost.

doi: 10.2313/NET-2022-11-1_03

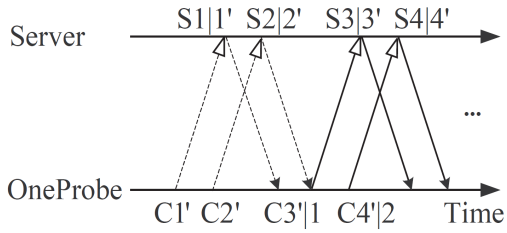

Figure 1: Probing process of OneProbe [1]

TABLE 1: 18 path events in one round of OneProbe [1]

Path events	1st response packets	2nd response packets	3rd response packets
1. F0 x R0	S3|3'	S4|4'	-
2. F0 x RR	S4|4'	S3|3'	-
3. F0 x R1	S4|4'	Ŝ3|4'	-
4. F0 x R2	S3|3'	Ŝ3|4'	-
5. F0 x R3	Ŝ3|4'	-	-
6. FR x R0	S3|2'	S4|2'	Ŝ3|4'
7. FR x RR	S4|2'	S3|2'	Ŝ3|4'
8. FR x R1	S4|2'	Ŝ3|4'	-
9. FR x R2	S3|2'	Ŝ3|4'	-
10. FR x R3	Ŝ3|4'	-	-
11. F1 x R0	S3|2'	S4|2'	Ŝ3|2'
12. F1 x RR	S4|2'	S3|2'	Ŝ3|2'
13. F1 x R1	S4|2'	Ŝ3|2'	-
14. F1 x R2	S3|2'	Ŝ3|2'	-
15. F1 x R3	Ŝ3|2'	-	-
16. F2 x R0	S3|3'	Ŝ2|3'	-
17. F2 x R1	Ŝ2|3'	-	-
18. F3	Ŝ1|2'	-	-

4) F2/R2: The server/OneProbe receives only the first probe/response packet, the second packet is lost.
5) F3/R3: The server/OneProbe receives no probe/response packet, both packets are lost.

There are 18 possible scenarios for packet loss and reordering in a probe round, resulting from the combination of the above events on the forward and reverse paths. These combinations and the response packets that the source host receives for all these 18 scenarios are listed in Table 1. Based on the response packets the source receives, OneProbe can identify which scenario is present and whether there is packet loss or reordering in the forward or reverse path. Scenarios 11 to 18 describe the scenarios of packet loss in the forward path. Scenarios 3, 4, 5, 8, 9, 10, 13, 14, 15, 17 describe the scenarios of packet loss in the reverse path. There are only 3 cases where the scenario cannot be distinguished because the responses are not unique:

- Scenario 14 and 15
- Scenario 12 and 13
- Scenario 5 and 10

The packet loss rate is measured by OneProbe by sending successive probe rounds to the destination. OneProbe only considers the first packet for measuring the loss rate. The packet loss rate for the forward path is calculated as follows:

$$\frac{\text{\#Probe rounds with first probe packet loss}}{\text{\#Total probe rounds}} \quad (1)$$

The packet loss rate for the reverse path is calculated as follows:

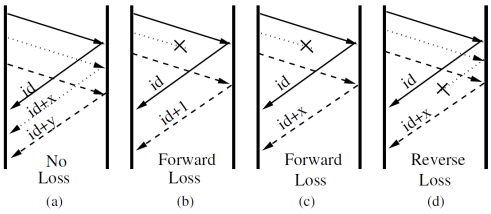

Figure 2: Packet loss scenarios in Tulip [2]

$$\frac{\text{\#Probe rounds with first response packet loss}}{\text{\#Total probe rounds}} \quad (2)$$

Tulip: One of the other tool which detects packet loss on a network path is Tulip proposed by Mahajan et al. [2]. Except detecting packet loss, Tulip can also identify the location of the packet loss within three hops. Tulip measures network characteristics in the forward path. If we want to meaure the reverse path, tulip can be used at the destination point. Tulip uses the IP identifier counters feature of routers to detect packet loss. Each IP packet contains a unique identification field (IP-ID) to enable IP fragments to be reassembled. Most routers implement the IP-ID using a counter, and the IP-ID is incremented with each packet generated. Tulip exploits these IP-IDs.

The loss detection mechanism of Tulip works as follows: The source sends three probe packets: two control packets and one data packet in the middle, to the router. Different protocols such as UDP, TCP or ICMP can be used for these probe packets. Each of these probe packets generates a response packet from the router. There are 3 possibilities for the data packet loss, as shown in Figure 2:

1) The source receives all three responses: No Loss
2) The source receives only two responses triggered by control packets and the IP-IDs of the response packets are consecutive: Forward Loss
3) The source receives only two responses triggered by control packets and the IP-IDs of the response packets are not consecutive: Indistinguishable whether forward loss or reverse loss

A few prerequisites exist for Tulip's loss detection mechanism: The control packets should always be retained, because if a control packet or its response is lost, it is not possible to detect the direction of data loss. The probe packets should arrive at the router close together in time and in the correct order so that they can obtain consecutive IP-IDs. The packet loss rate is calculated as follows:

$$\frac{\text{\#Probe rounds with forward loss}}{\text{\#Total probe rounds}} \quad (3)$$

Sting: Sting, introduced by Savage, is another tool that can measure packet loss rates along both the forward and reverse paths between a source and a destination [3]. Sting's loss deduction algorithm measures packet loss rate by leveraging the features of the TCP protocol. For packet loss measurement, Sting uses TCP acknowledgments.

The algorithm measuring the loss rate on forward path consists two phases, namely *data seeding* and *hole filling*. In the data seeding phase, sequential TCP packets are

doi: 10.2313/NET-2022-11-1_03

sent from the source to the destination. In the hole filling phase, the source sends another TCP data packet with a sequence number one higher than the last TCP data packet in the data seeding phase. If the source receives an acknowledgment for this packet, it concludes that no packet was lost in the data seeding phase. If the source receives a duplicate acknowledgement, it means a packet loss, and the number of the acknowledgement indicates which packet was lost. The source resends the corresponding packet. This process is continued until the last data packet sent in the data seeding phase is acknowledged. In this way, the total number of lost data packets is obtained.

Measuring the loss rate in the reverse path can be problematic. The source cannot count the acknowledgments that the destination sends. This is where ack parity is used. Sting ensures ack parity using a method that will not be elaborated on here. Ack parity guarentees that destination sends an acknowledgment for every packet it receives.

Five attributes are defined for the calculation of the forward and backward path loss rate:

dataSend: The total number of data packets sent from source to the destination, can be measured directly at the source.

dataLost: The total number of lost data packets measured with Sting's loss deduction algorithm.

dataReceived: dataReceived = dataSend - dataLost

ackSent: ackSent = dataReceived. Due to ack parity, an acknowledgement is issued for each received packet.

ackReceived: The total number of acknowledgments that have reached the source, can be measured directly at the source.

The loss rate for the forward path is calculated as follows:

$$1 - (\text{dataReceived}/\text{dataSent}) \tag{4}$$

The loss rate for the reverse path is calculated as follows:

$$1 - (\text{ackReceived}/\text{ackSent}) \tag{5}$$

3.2. Packet Reordering

Many methods have been developed to measure packet reordering behavior on network paths. In this subsection, two of them are introduced, namely OneProbe and Tulip.

OneProbe: All 18 possible scenarios for packet loss and reordering in a probe round for the OneProbe method have already been shown in Table 1. Scenarios 6 to 10 describe the scenarios of packet reordering in the forward path. Scenarios 2, 7, 12 describe the scenarios of packet reordering in the reverse path. The packet reordering rate is measured by OneProbe by sending successive probe rounds to the destination. OneProbe only considers the first packet for measuring the loss rate.

The packet reordering rate for the forward path is calculated as follows:

$$\frac{\#\text{Probe rounds with reordered probe packets}}{\#\text{Total probe rounds}} \tag{6}$$

The packet reordering rate for the reverse path is calculated as follows:

$$\frac{\#\text{Probe rounds with reordered response packets}}{\#\text{Total probe rounds}} \tag{7}$$

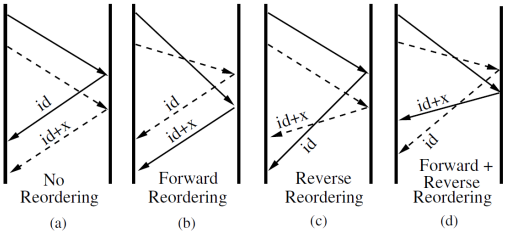

Figure 3: Packet reordering scenarios in Tulip [2]

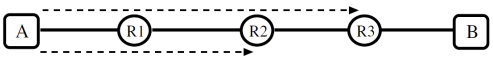

Figure 4: Building Blocks: The properties of link R2 → R3 can be estimated by subtracting the measured properties of path A → R2 from the measured properties of path A → R3. [2]

Tulip: Tulip uses IP-IDs to obtain information about packet reordering on the forward path [2]. To measure the reordering of the packets, the source sends two probe packets to the router. Each of these probe packets generates a response packet from the router that contains the headers of the probe packets so that they can be differentiated. Tulip uses IP-IDs to obtain information about the order in which packets reach the router.

There are 4 possibilities to reorder these two packages in the forward and backward paths, as shown in Figure 3:

1) No reordering: The responses are received in order and the second probe's response has a higher IP-ID.
2) Forward path reordering: The second probe's response has a lower IP-ID and reaches the source first.
3) Reverse path reordering: The second probe's response has a higher IP-ID but reaches the source first.
4) Forward and reverse path reordering: The second probe's response reaches the source second but has a lower IP-ID.

The packet reordering rate is calculated as follows:

$$\frac{\#\text{Reordered probe pairs}}{\#\text{Probe pairs for which both responses are received}} \tag{8}$$

3.3. Point of Failure for Packet Loss and Packet Reordering

The characteristics of the single links in the network can not be measured individually. In order to locate the point of a failure, we first make forward path measurements to both ends of a link and then compare the resulting measurements. This method is called Building Blocks, and is illustrated in Figure 4.

Tulip: The third metric that Tulip can measure is the point of failure on the network path using the Building Blocks method [2]. Tulip has 2 steps for locating the failure point. In the first step, the path from the source to the destination is determined using traceroute. In the second step, tulip performs either a parallel search or a binary search. In parallel search, the forward path to each router is measured one after another. In binary search, the forward path to the destination is measured first. If there is

doi: 10.2313/NET-2022-11-1_03

a fault, the forward path to middle point is measured, and depending on which part of the path contains the fault, the measurement continues with that part, and this continues recursively until the faulty link is found.

3.4. Round Trip Time

OneProbe: The third metric that OneProbe can measure is round trip time (RTT) [1]. OneProbe only considers the first packet for measuring round trip time. This is because the RTT of the second probe packet can be affected by the first packet [6]. The RTT is calculated as follows:

$$\begin{matrix}\text{First response packet}\\ \text{receive time}\end{matrix} \quad - \quad \begin{matrix}\text{First probe packet}\\ \text{sending time}\end{matrix} \tag{9}$$

3.5. Bottleneck Router Buffer Size

Loss Pairs: Liu and Crovella develop a tool called Loss Pairs for specify network characteristics such as the packet dropping behavior of a bottleneck router [4]. A loss pair defines a pair of packets where exactly one of the packet is discarded in the network while they were traveling on the same path and were close to each other in time [4]. The network conditions observed by these two packets are very similar. Thus, if one of the packets gets lost, very accurate estimates of the network conditions at the time of packet loss can be made based on the residual packet. This idea of loss pair method is used to identify router properties in the network and estimate the buffer size at a bottleneck router, assuming that most packet losses and delays occur at the bottleneck. If one packet of the pair is dropped, it indicates that the queue of the bottleneck router is full, and the calculated RTT for the residual packet includes the drain time of a full queue on the router. Therefore, the focus lies on the round trip time of the residual packet. The round trip time is obtained from the TCP data packets and their acknowledgments. And the calculated RTT is used to estimate the buffer size. It is assumed that we already know the bandwidth of the bottleneck link and the propagation delay along the path. Then the buffer size is calculated as follows:

$$\text{Bandwidth} \cdot (\text{RTT} - \text{Propagation Delay}) \tag{10}$$

4. Comparison of the Methods

Traditional ICMP-based tools such as Ping and Traceroute work universally and can be run on only one endpoint, but they provide limited and inaccurate results [3]. One problem with ICMP-based tools is that it is impossible to determine whether packet loss has occurred on the forward or reverse path [3]. The second problem is that routers and end hosts do not always respond to ICMP Ping and Traceroute [7], resulting in an inflated packet loss rate.

Sting overcomes these problems by exploiting the properties of the TCP protocol. Sting can distinguish in which direction packet loss occurs while still running on only one endpoint. One problem with Sting is that Sting uses TCP ACKs on the reverse path, even though it uses TCP data packets on the forward path. That is why Sting's reverse path measurement does not support different response packet sizes. Moreover, Sting's TCP ACKs based measurement fails for large response packet sizes. Because the TCP ACKs based measurement of reverse path loss may be underestimated for larger packets [8]. In an evaluation, Sting shows a failure rate of 54.8% for 41-byte probes and nearly 100% failure rate for 1053-byte probes [1].

Tulip can measure multiple metrics. Compared to Sting, it uses different patterns of probes. Sting targets end hosts running TCP-based servers on known ports, while Tulip can be used with both routers and hosts. Tulip provides reliable results for TCP data packets, but unreliable results for other packet types, such as UDP packets [9], since most routers in the network do not respond to UDP packets [10]. In addition, Tulip requires routers to support consecutive IP-ID measurement. Router without IP-ID counters do not support Tulip. In an evaluation, Tulip fails 80% of the time on packet loss and reordering measurements, 50% of the failures are due to Tulip using UDP probes, and 30% are due to some routers not supporting IP-ID measurements [1]. Also, Tulip cannot measure some packet loss scenarios [2].

Loss pairs enable the determination of router characteristics that were previously not directly measurable, such as the buffer size of bottleneck routers in the network. This method provides sufficiently accurate and robust results over a wide range of network configurations as well as under noisy network conditions [4]. In [4], it is claimed that whether the remaining packet is the first or the second in a loss pair makes no difference in determining the queuing delay of a congested router. However, an analysis performed in [11] shows that using the delay of the first packet tends to be more accurate than the delay of the second packet.

OneProbe is another tool capable of measuring multiple metrics, like Tulip. The packet sizes in OneProbe are configurable so that it can measure path metrics with different response packet sizes. This feature of OneProbe is used in [11] to confirm that the accuracy of delay estimation generally increases with a smaller packet size. OneProbe also overcomes some limitations of Tulip: OneProbe can measure multiple path metrics on the forward and reverse paths simultaneously with the same probe. Tulip's probe packets, on the other hand, differ in the loss and reordering measurements in the method itself. One problem with OneProbe is that it cannot distinguish some path events due to the ambiguity of some responses.

5. Conclusion and future work

In this paper an overview of network path monitoring methods with respect to packet loss, packet reordering, point of failure, round trip time, and bottleneck router buffer size is given. The technniques they use are explained, their missing features are pointed out, and a comparison between them is made.

OneProbe is the most reliable of the presented methods and provides correct and accurate results. It was tested on 39 systems and 35 web servers and found to be successful [1]. However, it does not cover a wide range of metrics. It could be promising to develop it to include more metrics. In the future, existing methods can be

further developed to overcome their limitations. Then, attempts can be made to add more metrics from other methods to OneProbe. In this way, a reliable method with a wide range of metrics can be developed.

References

[1] X. Luo, E. W. Chan, and R. K. Chang, "Design and Implementation of TCP Data Probes for Reliable and Metric-Rich Network Path Monitoring," in *USENIX Annual Technical Conference*, 2009.

[2] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "User-level Internet Path Diagnosis," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 106–119, 2003.

[3] S. Savage, "Sting: a TCP-based Network Measurement Tool," in *USENIX symposium on Internet Technologies and Systems*, vol. 2, 1999, pp. 7–7.

[4] J. Liu and M. Crovella, "Using Loss Pairs to Discover Network Properties," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 127–138.

[5] A. Lamberti, "How to Measure Network Performance: 9 Network Metrics," https://obkio.com/blog/how-to-measure-network-performance-metrics/#how-to-measure-network-performance, 2022, [Online; accessed 01-April-2022].

[6] J.-C. Bolot, "End-to-End Packet Delay and Loss Behavior in the Internet," in *Conference proceedings on Communications architectures, protocols and applications*, 1993, pp. 289–298.

[7] M. Luckie, Y. Hyun, and B. Huffaker, "Traceroute Probe Method and Forward IP Path Inference," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, 2008, pp. 311–324.

[8] S. Floyd and E. Kohler, "Tools for the Evaluation of Simulation and Testbed Scenarios," Internet Draft: draft-irtf-tmrg-tools, Tech. Rep., 2008.

[9] C. Parsa and J. Garcia-Luna-Aceves, "TULIP: A Link-Level Protocol for Improving TCP over Wireless Links," in *WCNC. 1999 IEEE Wireless Communications and Networking Conference (Cat. No. 99TH8466)*, vol. 3. IEEE, 1999, pp. 1253–1257.

[10] A. Haeberlen, M. Dischinger, K. P. Gummadi, and S. Saroiu, "Monarch: A Tool to Emulate Transport Protocol Flows over the Internet at Large," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 105–118.

[11] E. W. Chan, X. Luo, W. Li, W. W. Fok, and R. K. Chang, "Measurement of Loss Pairs in Network Paths," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, 2010, pp. 88–101.