TSVWG
Internet-Draft
Intended status: Informational
Expires: September 6, 2019

Y. Li
X. Zhou
Huawei

March 05, 2019

LOOPS (Localized Optimization of Path Segments) Problem Statement and Opportunities
draft-li-tsvwg-loops-problem-opportunities-00

Abstract

Various overlay tunnels are used in networks including WAN, enterprise campus and others. End to end paths are partitioned into multiple segments using overlay tunnels to achieve better path selection, lower latency and so on. Traditional end-to-end transport layers respond to packet loss slowly especially in long-haul networks: They either wait for some signal from the receiver to indicate a loss and then retransmit from the sender or rely on sender's timeout which is often quite long.

LOOPS (Localized Optimization of Path Segments) attempts to provide non end-to-end local based in-network recovery to achieve better data delivery by making packet loss recovery faster. In an overlay network scenario, LOOPS can be performed over the existing, or purposely created, overlay tunnel based path segments.

This document illustrates the slow packet loss recovery problems LOOPS tries to solve in some use cases and analyzes the impacts when local in-network recovery is employed as a LOOPS mechanism.

Table of Contents

1. Introduction

Overlay tunnels are widely deployed for various networks, including long haul WAN interconnection, enterprise wireless access networks, etc. The end to end connection is partitioned into multiple path segments using overlay tunnels. This serves a number of purposes, for instance, selecting a better path over the WAN or delivering the packets over heterogeneous network, such as enterprise access and core networks.

A reliable transport layer normally employs some end-to-end retransmission mechanisms which also address congestion control [RFC0793] [RFC5681]. The sender either waits for the receiver to

send some signals on a packet loss or sets some form of timeout for retransmission. For unreliable transport layer protocols such as RTP [RFC3550], optional and limited usage of end-to-end retransmission is employed to recover from packet loss [RFC4585] [RFC4588].

End-to-end retransmission to recover lost packets is slow especially when the network is long haul. When a path is partitioned into multiple path segments that are realized as overlay tunnels, LOOPS (Localized Optimization of Path Segments) tries to enhance transport over some path segment instead of end-to-end. Local in-network recovery is one example of LOOPS to make recovery from packet loss faster. Figure 1 shows a basic LOOPS usage scenario.

This document illustrates the slow packet loss recovery problems LOOPS tries to solve in some use cases and analyzes the impacts when local in-network recovery is employed as a LOOPS mechanism.

Section 3 presents some of the issues and opportunities found in Cloud-Internet overlay network that require higher performance and more reliable packet transmission in best effort networks. Section 4 describes the corresponding solution features and the impact of them on existing network technologies.

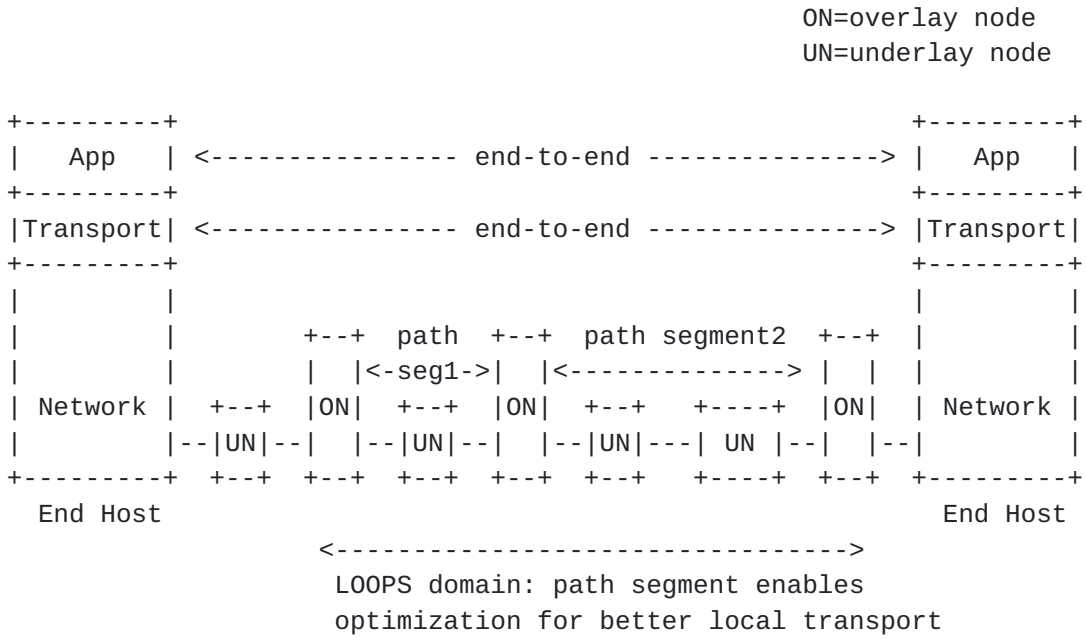

Figure 1: LOOPS in Overlay Network Usage Scenario

1.1. Terminology

LOOPS: Localized Optimization of Path Segments. LOOPS includes the local in-network (i.e. non end-to-end) recovery function, for instance, loss detection and measurements.

LOOPS Node: Node supporting LOOPS functions.

Overlay Node (ON): Node having overlay functions (like overlay protocol encapsulation/decapsulation, header modification, TLV inspection) and LOOPS functions in LOOPS overlay network usage scenario. Both OR and OE are Overlay Nodes.

Overlay Tunnel: It specifies a tunnel with designated ingress and egress nodes using some network overlay protocol as encapsulation.

Overlay Path: It specifies a channel within the overlay tunnel, and the traffic transmitted on the channel needs to pass through none or any number of designated intermediate overlay node. There may be more than one overlay path within an overlay tunnel when the different sets of designated intermediate overlay nodes are specified. An overlay path may contain multiple path segments. When an overlay tunnel contains only one overlay path without any intermediate overlay node specified, overlay path and overlay tunnel are used interchangeably.

Overlay Edge (OE): Edge node of an overlay tunnel.

Overlay Relay (OR): Intermediate overlay node on an overlay path. Overlay path may not contain any OR.

Path segment: Part of an overlay path between two neighbor overlay nodes. It is used interchangeably with overlay segment in this document when the context wants to emphasize on its overlay encapsulated nature. An overlay path may contain multiple path segments. When an overlay path contains only one path segment, i.e. the segment is between two OEs, the path segment is equivalent to the overlay path. It is also called segment for simplicity in this document.

Overlay segment: Refers to path segment.

Underlay Node (UN): Nodes not participating in overlay network function.

2. Cloud-Internet Overlay Network

The Internet is a huge network of networks. The interconnections of end devices using this global network are normally provided by ISPs (Internet Service Provider). This ISP provided huge network is considered as the traditional Internet. CSPs (Cloud Service Providers) are connecting their data centers using the Internet or via self-constructed networks/links. This expands the Internet's infrastructure and, together with the original ISP's infrastructure, forms the Internet underlay.

NFV (network function virtualization) further makes it easier to dynamically provision a new virtual node as a work load in a cloud for CPU/storage intensive functions. With the aid of various mechanisms such as kernel bypassing and Virtual IO, forwarding based on virtual nodes is becoming more and more effective. The interconnections among the purposely positioned virtual nodes and/or the existing nodes with virtualization functions potentially form an overlay of Internet. It is called the Cloud-Internet Overlay Network (CION) in this document.

CION makes use of overlay technologies to direct the traffic going through the specific overlay path regardless of the underlying physical topology, in order to achieve better service delivery. It purposely creates or selects overlay nodes (ON) from providers. By continuously measuring the delay of path segments and use them as metrics for path selection, when the number of overlay nodes is sufficiently large, there is a high chance that a better path could be found [DOI_10.1109_ICDCS.2016.49].Figure 2 shows an example of an overlay path over large geographic distances. The path between two OEs (Overlay Edges) is an overlay path. OEs are ON1 & ON4 in figure 2. Part of the path between ONs is a path segment. Figure 2 shows the overlay path with 3 segments, i.e. ON1-ON2-ON3-ON4. ON is usually a virtual node, though it does not have to be. Overlay path transmits packets in some form of network overlay protocol encapsulation. ON has the computing and memory resources that can be used for some functions like packet loss detection, network measurement and feedback, packet recovery.

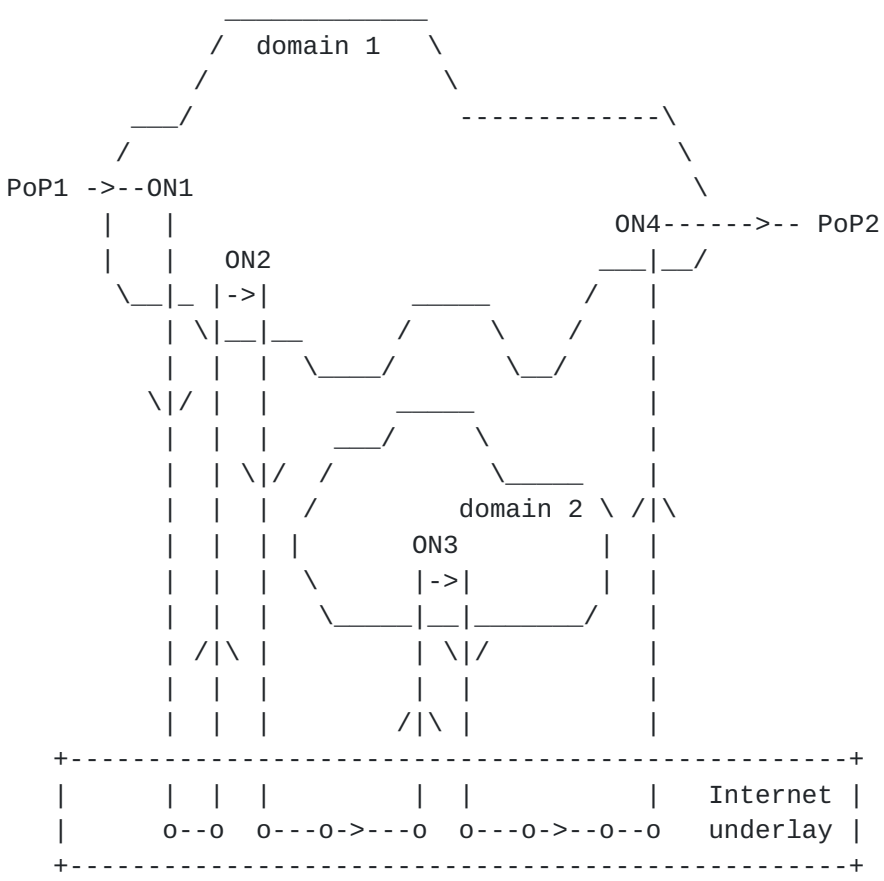

Figure 2: Cloud-Internet Overlay Network (CION)

We tested based on 37 overlay nodes from multiple cloud providers globally [DOI_10.1109_ICNP.2018.00034]. Each pair of the overlay nodes are used as sender and receiver. When the traffic is not intentionally directed to go through any intermediate virtual nodes, we call the path that the traffic takes the _default path_ in the test. When any of the virtual nodes is intentionally used as an intermediate node to forward the traffic, the path that the traffic takes is an _overlay path_ in the test. The preliminary experiments showed that the delay of an overlay path is shorter than that of the default path in 69% of cases at 99% percentile and improvement is 17.5% at 99% percentile when we probe Ping packets every second for a week.

Lower delay does not necessarily mean higher throughput. Different path segments may have different packet loss rates. Loss rate is another major factor impacting TCP throughput. From some customer requirements, we set the target loss rate to be less than 1% at 99% percentile and 99.9% percentile, respectively. The loss was measured between any two overlay nodes, i.e. any potential path segment. Two thousand Ping packets were sent every 20 seconds between two overlay

nodes for 55 hours. This preliminary experiment showed that the packet loss rate satisfaction are 44.27% and 29.51% at the 99% and 99.9% percentiles respectively.

Hence packet loss in an overlay segment is a key issue to be solved in CION. In long-haul networks, the end-to-end retransmission of lost packet can result in an extra round trip time. Such extra time is not acceptable in some cases. As CION naturally consists of multiple overlay segments, LOOPS tries to leverage it to do the local optimization for a single hop between two overlay nodes. ("Local" here is a concept relative to end-to-end, it does not mean such optimization is limited to LAN networks.)

The following subsections present different scenarios using multiple segment based overlay paths with a common need of local in-network loss recovery in best effort networks.

2.1. Tail Loss or Loss in Short Flows

When the lost segments are at the end of the transactions, TCP's fast retransmit algorithm does not work here as there are no ACKs to trigger it. When an ACK for a given segment is not received in a certain amount of time called retransmission timeout (RTO), the segment is resent [RFC6298]. RTO can be as long as several seconds. Hence the recovery of lost segments triggered by RTO is lengthy. [I-D.dukkipati-tcpm-tcp-loss-probe] indicates that large RTOs make a significant contribution to the long tail on the latency statistics of short flows like web pages.

The short flow often completes in one or two RTTs. Even when the loss is not a tail loss, it can possibly add another RTT because of end-to-end retransmission (not enough packets are in flight to trigger fast retransmit). In long haul networks, it can result in extra time of tens or even hundreds of milliseconds.

An overlay segment transmits the aggregated flows from ON to ON. As short flows are aggregated, the probability of tail loss over this specific overlay segment decreases compared to an individual flow. The overlay segment is much shorter than the end-to-end path in a Cloud- Internet overlay network, hence loss recovery over an overlay segment is faster.

2.2. Packet Loss in Real Time Media Streams

The Real-time transport protocol (RTP) is widely used in interactive audio and video. Packet loss degrades the quality of the received media. When the latency tolerance of the application is sufficiently large, the RTP sender may use RTCP NACK feedback from the receiver

[RFC4585] to trigger the retransmission of the lost packets before the playout time is reached at the receiver.

In a Cloud-Internet overlay network, the end-to-end path can be hundreds of milliseconds. End-to-end feedback based retransmission may be not be very useful when applications can not tolerate one more RTT of this length. Loss recovery over an overlay segment can then be used for the scenarios where RTCP NACK triggered retransmission is not appropriate.

2.3. Packet Loss and Congestion Control in Bulk Data Transfer

TCP congestion control algorithms such as Reno and CUBIC basically interpret packet loss as congestion experienced somewhere in the path. When a loss is detected, the congestion window will be decreased at the sender to make the sending slower. It has been observed that packet loss is not an accurate way to detect congestion in the current Internet [I-D.cardwell-iccrg-bbr-congestion-control]. In long-haul links, when the loss is caused by non-persistent burst which is extremely short and pretty random, the sender's reaction of reducing sending rate is not able to respond in time to the instantaneous path situation or to mitigate such bursts. On the contrary, reducing window size at the sender unnecessarily or too aggressively harms the throughput for application's long lasting traffic like bulk data transfer.

The overlay nodes are distributed over the path with computing capability, they are in a better position than the end hosts to deduce the underlying links' instantaneous situation from measuring the delay, loss or other metrics over the segment. Shorter round trip time over a path segment will benefit more accurate and immediate measurements for the maximum recent bandwidth available, the minimum recent latency, or trend of change. ONs can further decide if the sending rate reduction at the sender is necessary when a loss happened. Section 4.2 talks more details on this.

2.4. Multipathing

As an overlay path may suffer from an impairment of the underlying network, two or more overlay paths between the same set of ingress and egress overlay nodes can be combined for reliability purpose. During a transient time when a network impairment is detected, sending replicating traffic over two paths can improve reliability.

When two or more disjoint overlay paths are available as shown in figure 3 from ON1 to ON2, different sets of traffic may use different overlay paths. For instance, one path is for low latency and the

other is for higher bandwidth, or they can be simply used as load balancing for better bandwidth utilization.

Two disjoint paths can usually be found by measuring to figure out the segments with very low mathematical correlation in latency change. When the number of overlay nodes is large, it is easy to find disjoint or partially disjoint segments.

Different overlay paths may have varying characteristics. The overlay tunnel should allow the overlay path to handle the packet loss depending on its own path measurements.

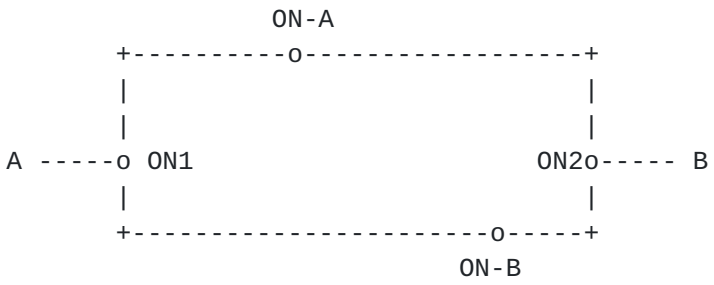

Figure 3: Multiple Overlay Paths

3. Features and Impacts to be Considered for LOOPS

LOOPS (Localized Optimization of Path Segments) tries to leverage the virtual nodes in a selected path to improve the transport performance "locally" instead of end-to-end as those nodes have partitioned the path to multiple segments. With the technologies like NFV (Network function virtualization) and virtual IO, it is easier to add functions to virtual nodes and even the forwarding on those virtual nodes is getting more efficient. Some overlay protocols such as VXLAN [RFC7348], GENEVE [I-D.ietf-nvo3-geneve], LISP [RFC6830] or CAPWAP [RFC5415] are assumed to be employed in the network. LOOPS is expected to use sequence number space independent from that of the transport layer. Acknowledgment should be used. To reduce overhead, negative ACK over each path segment is a good choice here. Measurement over segment or overlay path is required to perform local recovery. LOOPS does not look into the traffic payload. Elements to be considered in LOOPS are discussed briefly here.

3.1. Local Recovery and End-to-end Retransmission

There are basically two ways to perform local recovery, retransmission and FEC (forward error correction). They are possibly used together in some cases. Such approaches between two overlay nodes recover the lost packet in relatively shorter distance and thus

shorter latency. Therefore the local recovery is always faster compared to end-to- end.

At the same time, most transport layer protocols have their own end-to-end retransmission to recover the lost packet. It would be ideal that end-to-end retransmission at the sender was not triggered if the local recovery was successful.

End-to-end retransmission is normally triggered by a NACK like in RTCP or multiple duplicate ACKs like in TCP.

When FEC is used for local recovery, it may come with a buffer to make sure the recovered packets delivered are in order subsequently. Therefore the receiver side is unlikely to see the out-of-order packets and then send a NACK or multiple duplicate ACKs. The side effect to unnecessarily trigger end-to-end retransmit is minimum. When FEC is used, if redundancy and block size are determined, extra latency required to recover lost packets is also bounded. Then RTT variation caused by it is predictable. In some extreme case like a large number of packet loss caused by persistent burst, FEC may not be able to recover it. Then end-to-end retransmit will work as a last resort. In summary, when FEC is used as local recovery, the impact on end-to-end retransmission is limited.

When retransmission is used, more care is required.

For packet loss in RTP streaming, retransmission can recover those packets which would not be retransmitted end-to-end otherwise due to long RTT. It would be ideal if the retransmitted packet reaches the receiver before a NACK for the lost packet would be sent out. Therefore when the segment(s) being retransmitted is a small portion of the whole end to end path, the retransmission will have a significant effect of improving the quality at receiver. When the sender also re-transmits the packet based on a NACK received, the receiver will receive the duplicated retransmitted packets and should ignore the duplication.

For packet loss in TCP flows, TCP RENO and CUBIC use duplicate ACKs as a loss signal to trigger the fast retransmit. There are different ways to prevent that the sender's end-to-end retransmission is triggered prematurely:

- The egress overlay node can buffer the out-of-order packets for a while to give a limited time for a packet retransmitted somewhere in the overlay path to reach it. The retransmitted packet and the buffered packets caused by it may increase the RTT variation at the sender. When the retransmitted latency is a small portion of RTT or the loss is rare, such RTT variation will be smoothed

without much impact. Another possible way is to make the sender exclude such packets from the RTT measurement. The buffer management is nontrivial. It has to be determined how many out-of-order packets can be buffered at the egress overlay node before it gives up waiting for a successful local retransmission. As the lost packet is not always recovered successfully locally, the sender may invoke end-to-end fast retransmit slower than it would be in classic TCP.

- o If LOOPS network does not buffer the out-of-order packets caused by packet loss, TCP sender can use a time based loss detection like RACK [I-D.ietf-tcpm-rack] to prevent the TCP sender from invoking the fast retransmit too early. RACK uses the notion of time to replace the conventional DUPACK threshold approach to detect losses. RACK is required to be tuned to fit the local retransmission better. If there are n segments over the path, segment retransmission will at least add RTT/n to the reordering window by average when the packet is lost only once over the whole overlay path. This approach is more preferred than one described in previous bullet.

3.1.1. OE to OE Measurement, Recovery and Multipathing

When local recovery is between two neighbor ONs, it is called per-hop recovery. It can be between overlay relays or between overlay relay and overlay edge. Another type of local recovery is called OE to OE recovery which performs between overlay edge nodes. When the segments of an overlay path have similar characteristics and/or only OE has the expected processing capability, OE to OE based local recovery can be used instead of per-hop recovery.

If there are more than one overlay path in an overlay tunnel, multipathing splits and recombines the traffic. Measurement like round trip time and loss rate between OEs has to be path based. The ingress OE can use the feedback measurement to determine the FEC parameter settings for different path. FEC can also be configured to work over the combined path. The egress OE must be able to remove the replicated packet when overlay path is switched during impairment.

OE to OE measurement can help each segment determine its proportion in edge to edge delay. It is useful for ON to decide if it is necessary to turn on the per-hop recovery or how to fine tune the parameter settings. When the segment delay ratio is small, the segment retransmission is more effective.

3.2. Congestion Control Interaction

When a TCP-like transport layer protocol is used, local recovery in LOOPS has to interact with the upper layer transport congestion control. Classic TCP adjusts the congestion window when a loss is detected and fast retransmit is invoked.

Local recovery mechanism breaks the assumption of the necessary and sufficient conditional relationship between detected packet loss and congestion control trigger at the sender in classic TCP. A locally recovered packet can be caused by a non-persistent congestion such as a microburst or a random loss which ideally would not let sender invoke the congestion control reduction mechanism. And it can also possibly caused by a real persistent congestion which should let the sender invoke reduction. In either case, the sender does not detect such a loss if local recovery succeeds.

When the local recovery takes effect, we consider the following two cases. Firstly, the classic TCP sender does not see the enough number of duplicate ACKs to trigger fast retransmit. This could be the result of in-order packet delivery including locally recovered ones to the receiver as mentioned in last subsection. Classic TCP sender in this case will not reduce congestion window as no loss is detected. Secondly, if a time based loss detection such as RACK is used, as long as the locally recovered packet's ACK reaches the sender before the reordering window expires, the congestion window will not be reduced.

Such behavior brings great throughput improvement and it is desirable when the recovered packet was lost due to non-persistent congestion or random factors. It solves the throughput problem mentioned in section 3.3. However, it also brings the risk that the sender is not able to detect the real persistent congestion in time and then overshoot. Eventually a severe congestion that is not recoverable by a local recovery mechanism may occur. In addition, it may be unfriendly to other flows (possibly pushing them out) if those flows are running over the same underlying bottleneck links.

There is a spectrum of approaches. On one end, each locally recovered packet can be treated exactly as a loss in order to invoke the congestion control at the sender to guarantee the fair sharing as classic TCP by setting its CE (Congestion Experienced) bit. Explicit Congestion Notification (ECN) can be used here as ECN marking was required to be equivalent to a packet drop [RFC3168]. Congestion control at the sender works as usual and no throughput improvement could be achieved (although the benefit of faster recovery is still there). On the other hand, ON can perform its congestion measurement over the segment, for instance local RTT variation and throughput

change trend. Then the lost packet can be determined if it was caused by congestion or other factors. It will further decide if it is necessary to set CE marking or even what ratio is set to make the sender adjust the sending rate more correctly.

There are possible cases that the sender detects the loss even with local recovery in function. For example, when the re-ordering window in RACK is not optimally adapted, the sender may trigger the congestion control at the same time of end-to-end retransmission. If spurious retransmission detection based on DSACK [RFC3708] is used, such end-to-end retransmission will be found out unnecessary when locally recovered packets reaches the receiver successfully. Then congestion control changes might be undone at the sender. This results in similar pros and cons as described earlier. Pros are preventing the necessary window reduction and improving the throughput when the loss is considered caused by non-persistent congestion or random loss. Cons are some mechanisms like ECN or its variants should be used wisely to make sure the congestion control is invoked in case of persistent congestion.

An approach where the losses on a path segment are not immediately made known to the end-to-end congestion control can be combined with a "circuit breaker" style congestion control on the path segment. When the usage of path segment by the overlay flow starts to become unfair, the path segment sends congestion signals up to the end-to-end congestion control. This must be carefully tuned to avoid unwanted oscillation.

3.3. Overlay Protocol Extensions

The overlay usually has no control over how packets are routed in the underlying network between two overlay nodes, but it can control, for example, the sequence of overlay nodes a message traverses before reaching its destination. LOOPS assumes the overlay protocol can deliver the packets in such designated sequence. Most forms of overlay networking use some sort of "encapsulation". The whole path taken can be performed by stitching multiple short overlay paths, like VXLAN[RFC7348], GENEVE [I-D.ietf-nvo3-geneve], or it can be a single overlay path with a sequence of intermediate overlay nodes specified, like SRv6 [I-D.ietf-6man-segment-routing-header]. In either way, LOOPS requires to extend the protocol to support the data plane measurement and feedback, retransmission or FEC based loss recovery either per ON-hop based or OE to OE based.

LOOPS alone has no setup requirement on control plane. Some overlay protocol, e.g. CAPWAP [RFC5415], has session setup phase, we can use it to exchange the infomation like dynamic FEC parameters.

3.4. Summary

LOOPS is expected to extend the existing overlay protocols in data plane. Path selection is assumed a feature provided by the overlay protocols via SDN or other approaches and is not a part of LOOPS. LOOPS is a set of functions to be implemented on ONs in a long haul overlay network. LOOPS includes the following features.

1. Local recovery. Retransmission, FEC or hybrid can be used as local recovery method. Such recovery mechanism is in-network. It is performed by two network nodes with computing and memory resources.

2. Local congestion measurement. Sender ON measures the local segment RTT and/or loss to immediately get the overlay segment status.

3. Determination on how to set ECN CE marking based on local recovery and/or local congestion measurement information to feedback the end host sender to adjust the sending rate correctly.

4. Security Considerations

LOOPS does not look at the traffic payload, so encrypted payload does not affect functionality of LOOPS. The use of LOOPS introduces some issues which impact security. ON with LOOPS function represents a point in the network where the traffic can be potentially manipulated. Denial of service attack can be launched from an ON. A rogue ON might be able to spoof packet as if it come from a legitimate ON. It may also modify the ECN CE marking in packets to influence the sender's rate. In order to protected from such attacks, the overlay protocol itself should have some build-in security protection which inherently be used by LOOPS. The operator should use some authentication mechanism to make sure ONs are valid and non-compromised.

5. IANA Considerations

No IANA action is required.

6. Informative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/info/rfc3550>.

[RFC3708] Blanton, E. and M. Allman, "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions", RFC 3708, DOI 10.17487/RFC3708, February 2004, <https://www.rfc-editor.org/info/rfc3708>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <https://www.rfc-editor.org/info/rfc4585>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <https://www.rfc-editor.org/info/rfc4588>.

[RFC5415] Calhoun, P., Ed., Montemurro, M., Ed., and D. Stanley, Ed., "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", RFC 5415, DOI 10.17487/RFC5415, March 2009, <https://www.rfc-editor.org/info/rfc5415>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/info/rfc5681>.

[RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <https://www.rfc-editor.org/info/rfc6298>.

[RFC6830] Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, DOI 10.17487/RFC6830, January 2013, <https://www.rfc-editor.org/info/rfc6830>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC8257] Bensley, S., Thaler, D., Balasubramanian, P., Eggert, L., and G. Judd, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", RFC 8257, DOI 10.17487/RFC8257, October 2017, <https://www.rfc-editor.org/info/rfc8257>.

[I-D.dukkipati-tcpm-tcp-loss-probe] Dukkipati, N., Cardwell, N., Cheng, Y., and M. Mathis, "Tail Loss Probe (TLP): An Algorithm for Fast Recovery of Tail Losses", draft-dukkipati-tcpm-tcp-loss-probe-01 (work in progress), February 2013.

[I-D.ietf-nvo3-geneve] Gross, J., Ganga, I., and T. Sridhar, "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-10 (work in progress), March 2019.

[I-D.ietf-tcpm-rack] Cheng, Y., Cardwell, N., Dukkipati, N., and P. Jha, "RACK: a time-based fast loss detection algorithm for TCP", draft-ietf-tcpm-rack-04 (work in progress), July 2018.

[I-D.ietf-6man-segment-routing-header] Filsfils, C., Previdi, S., Leddy, J., Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment Routing Header (SRH)", draft-ietf-6man-segment-routing-header-16 (work in progress), February 2019.

[I-D.cardwell-iccrg-bbr-congestion-control] Cardwell, N., Cheng, Y., Yeganeh, S., and V. Jacobson, "BBR Congestion Control", draft-cardwell-iccrg-bbr-congestion-control-00 (work in progress), July 2017.

[DOI_10.1109_ICNP.2018.00034] Gu, L., Ju, R., Xu, Z., Li, J., and F. Li, "LTSM: Lightweight and Time Sliced Measurement for Link State", 2018 IEEE 26th International Conference on Network Protocols (ICNP), DOI 10.1109/icnp.2018.00034, September 2018.

[DOI_10.1109_ICDCS.2016.49]
Cai, C., Le, F., Sun, X., Xie, G., Jamjoom, H., and R. Campbell, "CRONets: Cloud-Routed Overlay Networks", 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), DOI 10.1109/icdcs.2016.49, June 2016.

Authors' Addresses

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Phone: +86-25-56624584
Email: liyizhou@huawei.com

Xingwang Zhou
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Email: zhouxingwang@huawei.com