Abstract

In various network deployments, end-to-end forwarding paths are partitioned into multiple segments. For example, in some cloud-based WAN communications, stitching multiple overlay tunnels are used for traffic policy enforcement matters such as to optimize traffic distribution or to select paths exposing a lower latency. Likewise, in satellite communications, the communication path is decomposed into two terrestrial segments and a satellite segment. Such long-haul paths are naturally composed of multiple network segments with various encapsulation schemes. Packet loss may show different characteristics on different segments.

Traditional transport protocols (e.g., TCP) 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. Non-congestive loss may make the TCP sender over-reduce the sending rate unnecessarily. With the increase of end-to-end transport encryption (e.g., QUIC), traditional PEP (performance enhancing proxy) techniques such as TCP splitting are no longer applicable.

LOOPS (Localized Optimizations on Path Segments) is a network-assisted performance enhancement over path segment and it aims to provide local in-network recovery to achieve better data delivery by making packet loss recovery faster and by avoiding the senders over-reducing their sending rate. In an overlay network scenario, LOOPS can be performed over a variety of the existing, or purposely created, tunnel-based path segments.
Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 8, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
   1.1. The Problem ............................................. 3
   1.2. Sketching a Work Direction: Rationale & Goals ......... 4
2. Terminology ................................................... 6
3. Cloud-Internet Overlay Network ............................... 7
   3.1. Tail Loss or Loss in Short Flows ....................... 9
   3.2. Packet Loss in Real Time Media Streams ............... 9
   3.3. Packet Loss and Congestion Control in Bulk Data Transfer 10
3.4. Multipathing .............................................. 10
4. Satellite Communication ....................................... 11
5. Branch Office WAN Connection ................................ 13
6. Features and Impacts to be Considered for LOOPS .......... 14
   6.1. Local Recovery and End-to-end Retransmission .......... 15
       6.1.1. OE to OE Measurement, Recovery, and Multipathing .. 17
6.2. Congestion Control Interaction .......................... 18
6.3. Overlay Protocol Extensions .............................. 19
6.4. Summary .................................................. 20
7. Security Considerations ................................... 20
8. IANA Considerations ....................................... 21
9. Acknowledgements ......................................... 21
10. Informative References .................................... 21
Authors’ Addresses ............................................. 24

1. Introduction

1.1. The Problem

Tunnels are widely deployed within many networks to achieve various engineering goals, including long-haul WAN interconnection or enterprise wireless access networks. A connection between two endpoints can be decomposed into many connection legs. As such, the corresponding forwarding path can be partitioned into multiple path segments that some of them are using network overlays by means of tunnels. This design serves a number of purposes such as steering the traffic, optimize egress/ingress link utilization, optimize traffic performance metrics (such as delay, delay variation, or loss), optimize resource utilization by invoking resource bonding, provide high-availability, etc.

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 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 typically as overlay tunnels, LOOPS (Local Optimizations on Path Segments) aims to provide local segment based in-network recovery to achieve better data delivery by making packet loss recovery faster and by avoiding the senders over-reducing their sending rate. In an overlay network scenario, LOOPS can be performed over the existing, or purposely created, overlay tunnel based path segments. Figure 1 show a basic usage scenario of LOOPS.

Some link types (satellite, microwave, drone-based networking, etc.) may exhibit unusually high loss rate in special conditions (e.g., fades due to heavy rain). The traditional TCP sender interprets loss as congestion and over-reduces the sending rate, degrading the
throughput. LOOPS is also applicable to such scenarios to improve the throughput.

Also, multiple paths may be available in the network that may be used for better performance. These paths are not visible to endpoints. Means to make use of these paths while ensuring the overall performance is enhanced would contribute to customer satisfaction. Blindly implementing link aggregation may lead to undesired effects (e.g., underperform compared to single path).

1.2. Sketching a Work Direction: Rationale & Goals

This document sketches a proposal that is meant to experimentally investigate to what extent a network-assisted approach can contribute to increase the overall perceived quality of experience in specific situations (e.g., Sections 3.5 and 3.6 of [RFC8517]) without requiring access to internal transport primitives. The rationale beneath this approach is that some information (loss detection, better visibility on available paths and their characteristics, etc.) can be used to trigger local actions while avoiding as much as possible undesired side effects (e.g., expose a behavior that would be interpreted by an endpoint as an anomaly (corrupt data) and which would lead to exacerbate end-to-end recovery. Such local actions would have a faster effect (e.g., faster recovery, used multiple paths simultaneously).

To that aim, the work is structured into two (2) phased stages:

- Stage 1: Network-assisted optimization. This one assumes that optimizations (e.g., support latency-sensitive applications) can be implemented at the network without requiring defining new interaction with the endpoint. Existing tools such as ECN will be used. Some of these optimizations may be valuable in deployments where communications are established over paths that are not exposing the same performance characteristics.

- Stage 2: Collaborative networking optimization. This one requires more interaction between the network and an endpoint to implement coordinated and more surgical network-assisted optimizations based on information/instructions shared by an endpoint or sharing locally-visible information with endpoint for better and faster recovery.

The document focuses on the first stage. Effort related to the second stage is out of scope of the initial planned work. Nevertheless, future work will be planned once progress is (hopefully) made on the first stage.
The proposed mechanism is not meant to be applied to all traffic, but only to a subset which is eligible to the network-assisted optimization service.

Which traffic is eligible is deployment-specific and policy-based. For example, techniques for dynamic information of optimization function (e.g., SFC) may be leveraged to unambiguously identify the aggregate of traffic that is eligible to the service. Such identification may be triggered by subscription actions made by customers or be provided by a network provider (e.g., specific-applications, during specific events such as during severe DDoS attack or flash crowds events).

Likewise, whether the optimization function is permanently instantiated or on-demand is deployment-specific.

This document does not intend to provide a comprehensive list of target deployment cases. Sample scenarios are described to illustrate some LOOPS potentials. Similar issues and optimizations may be helpful in other deployments such as enhancing the reliability of data transfer when a fleet of drones are used for specific missions (e.g., site inspection, live streaming, and emergency service). Captured data should be reliably transmitted via paths involving radio connections.

It is not required that all segments are LOOPS-aware to benefit from LOOPS advantages.

Section 3 presents some of the issues and opportunities found in Cloud-Internet overlay networks that require higher performance and more reliable packet transmission over best effort networks. Section 4 discusses applications of LOOPS in satellite communication. Section 6 describes the corresponding solution features and their impact on existing network technologies.
2. Terminology

This document makes use of the following terms:

LOOPS: Local Optimizations on Path Segments. LOOPS includes to the local in-network (i.e., non end-to-end) recovery functions and other supporting features such as local measurement, loss detection, and congestion feedback.

LOOPS Node: A node supporting LOOPS functions.

Overlay Node (ON): A node having overlay functions (e.g., overlay protocol encapsulation/decapsulation, header modification, TLV inspection) and LOOPS functions in LOOPS overlay network usage scenario.

Overlay Tunnel: A tunnel with designated ingress and egress nodes using some network overlay protocol as encapsulation, optionally with a specific traffic type.

Overlay Edge (OE): Edge node of an overlay tunnel. It can behave as ingress or egress as a function of the traffic direction.

Path segment: A LOOPS enabled tunnel-based network subpath. It is used interchangeably with overlay segment in this document when the context wants to emphasize on its overlay encapsulated nature. It is also called segment for simplicity in this document.

Figure 1: LOOPS in Overlay Network Usage Scenario

---

[Diagram showing the usage of LOOPS in an overlay network]

---
Overlay segment: Refers to path segment.

Underlay Node (UN): A node not participating in the overlay network.

3. Cloud-Internet Overlay Network

CSPs (Cloud Service Providers) are connecting their data centers using the Internet or via self-constructed networks/links. This expands the traditional Internet’s infrastructure and, together with the original ISP’s infrastructure, forms the Internet underlay.

Automation techniques and NFV (Network Function Virtualization) further ambitions to make it easier to dynamically provision a new virtual node/function as a workload 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 interconnection among the purposely positioned virtual nodes and/or the existing nodes with virtualization functions potentially form an overlay infrastructure. It is called the Cloud-Internet Overlay Network (CION) in this document for short.

This architecture scenario 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] [DOI_10.1145_3038912.3052560]. [DOI_10.1145_3038912.3052560] further shows all cloud providers experience random loss episodes and random loss accounts for more than 35% of total loss.

Some of the considerations that are discussed below may also apply for interconnecting DCs owned by a network provider.

Figure 2 shows an example of an overlay path over large geographic distances. Three path segments, i.e., ON1-ON2, ON2-ON3, ON3-ON4 are shown. ON is usually a virtual node, though it does not have to be. Each segment transmits packets using 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, and packet recovery. ONs are managed by a single administrator though they can be workloads created from different CSPs.
We tested based on 37 overlay nodes from multiple cloud providers globally. 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 followed by the traffic in the test as the default path. When any of the virtual nodes is intentionally used as an intermediate node to forward the traffic, the path that the traffic takes is called an overlay path. The preliminary experiments showed that the delay of an overlay path is shorter than the one 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. More experimental information can be found in [OCN].

Lower delay does not necessarily mean higher throughput. Different path segments may have different packet loss rates. Loss rate is another major factor impacting the overall TCP throughput. From some customer requirements, the target loss rate is set in the test 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 such architecture. In long-haul networks, the end-to-end retransmission of lost packet can result in an extra round trip time (RTT). Such extra time is not acceptable in some latency-sensitive applications. As CION naturally consists of multiple overlay segments, LOOPS leverages this to perform local optimizations on 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.

3.1. Tail Loss or Loss in Short Flows

When the lost segments are at the end of a transaction, TCP’s fast retransmit algorithm does not work as there are no ACKs to trigger it. When a sender does not receive an ACK for a given segment within a certain amount of time called retransmission timeout (RTO), it re-sends the segment [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 such as loading 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-lived 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.

3.2. 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.

3.3. 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 quickly 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 6.2 talks more details on this.

3.4. 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 be, for example, found by measurement 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. This information may be available if the ONs are managed by the network provider managing the underlying forwarding paths.

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

![Diagram of multiple overlay paths]

In reference to Figure 3, both A and B are not aware of the existence of these multiple paths. A network-assistance would be valuable for the sake of better resilience and performance. Note that in a collaborative context (a.k.a., stage 2 mentioned in Section 1.2) LOOPS may target means to advertise the available path characteristics to an endpoint A/B, to allow an endpoint A/B to control the traffic distribution policy to be enforced by ON1/ON2, or to let endpoint A/B notify ON1/ON2 with their multipathing preference.

4. Satellite Communication

Traditionally, satellite communications deploy PEP (performance enhancing proxy [RFC3135]) nodes around the satellite link to enhance end-to-end performance. TCP splitting is a common approach employed by such PEPs, where the TCP connection is split into three: the segment before the satellite hop, the satellite section (uplink, downlink), and the segment behind the satellite hop. This requires heavy interactions with the end-to-end transport protocols, usually without the explicit consent of the end hosts. Unfortunately, this is indistinguishable from a man-in-the-middle attack on TCP. With end-to-end encryption moving under the transport (QUIC), this approach is no longer useful.
Geosynchronous Earth Orbit (GEO) satellites have a one-way delay (up to the satellite and back) on the order of 250 milliseconds. This does not include queuing, coding and other delays in the satellite ground equipment. The Round Trip Time for a TCP or QUIC connection going over a satellite hop in both directions, in the best case, will be on the order of 600 milliseconds. And, it may be considerably longer. RTTs on this order of magnitude have significant performance implications.

Packet loss recovery is an area where splitting the TCP connection into different parts helps. Packets lost on the terrestrial links can be recovered at terrestrial latencies. Packet loss on the satellite link can be recovered more quickly by an optimized satellite protocol between the PEPs and/or link layer FEC than they could be end to end. Again, encryption makes TCP splitting no longer applicable. Enhanced error recovery at the satellite link layer helps for the loss on the satellite link but doesn’t help for the terrestrial links. Even when the terrestrial segments are short, any loss must be recovered across the satellite link delay. And, there are cases when a satellite ground station connects to the general Internet with a potentially larger terrestrial segment (e.g., to a correspondent host in another country). Faster recovery over such long terrestrial segments is desirable.

Another aspect of recovery is that terrestrial loss is highly likely to be congestion related but satellite loss is more likely to be transmission errors due to link conditions. A transport endpoint slowing down because of mis-interpreting these errors as congestion losses unnecessarily reduces performance. But, at the end points, the difference between the two is not easily distinguished. To elaborate more on the loss recovery for satellite communications, while the error rate on the satellite paths is generally very low most of the time, it might get higher during special link conditions (e.g., fades due to heavy rain). The satellite hop itself does know which losses are due to link conditions as opposed to congestion, but it has no mechanism to signal this difference to the end hosts.

We will need the protocol under QUIC to try to minimize non-congestion packet drop. Specific link layers may have techniques such as satellite FEC to recover. Where the capabilities of that may be exceeded (e.g., rain fade), we can look at LOOPS-like approaches.

There are two high level classes of solutions for making encrypted transport traffic like QUIC work well over satellite:

- Hooks in the transport protocol which can adapt to large BDPs where both the bandwidth and the latency are large. This would require end to end enhancement.
o Capabilities (such as LOOPS) under the transport protocol to improve performance over specific segments of the path. In particular, separating the terrestrial from the satellite losses. Fixing the terrestrial loss quickly and keeping throughput high over satellite segment by not causing the end-hosts to over-reduce their sending window in case of non-congestion loss.

This document focuses on the latter.

5. Branch Office WAN Connection

Enterprises usually require network connections between the branch offices or between branch offices and cloud data center over geographic distances. With the increasing deployment of vCPE (virtual CPE), some services usually hosted on the CPE are moved to the provider network from the customer site. Such vCPE approach enables some value added service to be provided such as WAN optimization and traffic steering.

Figure 4 shows an example of two branch offices WAN connection via Internet. Figure 5 shows a branch office access to public cloud via a selected PoP (point of presence). vCPE connects to that PoP which can be hundreds of kilometers away via Internet. In both cases, the path segments over Internet is subject to loss. Similar problems presented in subsections of Section 3 should be solved. The GW1 may be reachable via multiple paths.

Requirements to steer traffic through different sub-paths for latency optimization, resource optimization, balancing, or other purposes are increasing. For example, directing the traffic from vCPE to a lightly loaded PoP rather than to the closest one. Mere best effort transport is not sufficient. New technologies like SFC (Service Function Chaining), SRv6 (segment routing over IPv6), and NFV/SDN used together with vCPE to enable the potentials to embed more complicated loss recovery functions at intermediate nodes in end-to-end path.

```
+------+-+       +-----+   Internet    +------+-+       +-----+
  | GW1  |-------|vCPE1|---------------| vCPE2|-------+ GW2 |
  +------+-+       +-----+               +------+-+       +-----+
    Site A                                              Site B
```

Figure 4: Branch Office WAN Connection via Internet
6. Features and Impacts to be Considered for LOOPS

This section provides an overview of the proposed LOOPS solution. This section is not meant to document a detailed specification, but it is meant to highlight some design choices that may be followed during the solution design phase.

LOOPS aims to improve the transport performance "locally" in addition to native end-to-end mechanism supported by a given transport protocol. This is possible because LOOPS nodes will be instantiated to partition the path into multiple segments. With the advent of automation and technologies like NFV and virtual IO, it is possible to dynamically instantiate functions to nodes. Some overlay protocols such as VXLAN [RFC7348], GENEVE [I-D.ietf-nvo3-geneve], LISP [RFC6830] or CAPWAP [RFC5415] may be used in the network. In overlay network usage scenario, LOOPS can extend a specific overlay...
protocol header to perform local measurement and local recovery functions, like the example shown in Figure 6.

```
+------------+------------+-----------------+---------+---------+
|Outer IP hdr|Overlay hdr |LOOPS information|Inner hdr|payload  |
+------------+------------+-----------------+---------+---------+
```

Figure 6: LOOPS Extension Header Example

LOOPS should be designed to minimize its overhead while increasing the benefit (e.g., reduces the completion time of a video application, reduces the loss). Also, LOOPS should be designed to auto-tune itself in case its overhead is exceeding a threshold.

For example, LOOPS uses packet number space independent from that of the transport layer. Acknowledgment should be generated from ON receiver to ON sender for packet loss detection and local measurement. To reduce overhead, negative ACK over each path segment is a good choice here. A Timestamp echo mechanism, analogous to TCP’s Timestamp option, should be employed in-band in LOOPS extension to measure the local RTT and variation for an overlay segment. Local in-network recovery is performed. The measurement over segment is expected to give a hint on whether the lost packet of locally recovered one was caused by congestion. Such a hint could be further feedback, using like by ECN Congestion Experienced (CE) markings, to the end host sender. It directs the end host sender if congestion window adjustment is necessary. LOOPS normally works on the overlay segment which aggregates the same type of traffic, for instance TCP traffic or finer granularity like TCP throughput sensitive traffic. LOOPS does not look into the inner packet (when an encapsulation scheme is used). Elements to be considered in LOOPS are discussed briefly here.

6.1. 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 if end-to-end retransmission at the sender was not triggered when the local recovery is successful.
End-to-end retransmission is normally triggered by a NACK as in RTCP or multiple duplicate ACKs as 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 local retransmission is used, more care is required.

For packet loss in RTP streaming, local 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 it sends back information that the sender would interpret as a NACK for the lost packet. 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 avoid the sender’s end-to-end retransmission being triggered prematurely:

- The egress overlay node can buffer the out-of-order packets for a while, giving a limited time for a packet being 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 locally recovered packets can be specially marked and this marking is spin back to end host sender. Then RTT measurement should not use that packet.

The buffer management is nontrivial in this case. 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. In some extreme case the lost packet is not recovered successfully locally, the sender may invoke end-to-end fast retransmit slower than it would be in classic TCP.

- 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 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 similar 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. On the other hand, if time based loss detection is not supported at the sender, end to end retransmission will be invoked as usual. It wastes some bandwidth.

6.1.1. OE to OE Measurement, Recovery, and Multipathing

When multiple segments are stitched, another type of local recovery can be is performed between OE (Overlay Edge) to OE. 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-segment based recovery.

If there is more than one overlay path between two OEs, multipathing can split and recombine the traffic. Measurements such as RTT and loss rate between OEs have to be specific to each path. 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. FEC should not increase redundancy over the path where a congestion is found. The egress OE should be able to remove the duplicated packets when multipathing is available.

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 segment recovery or how to fine tune the parameter settings. When the segment delay ratio is small, the segment retransmission is more effective. Such approach requires nested LOOPS function. This draft does not focus on the nest LOOPS now. More details will be discussed later if comments showing interests in it are received.
6.2. 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.

The 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. The loss that is locally recovered can be caused by a non-persistent congestion such as a random loss or a microburst, both of which ideally would not let the sender invoke the congestion control mechanism. But then, loss can also possibly caused by a real persistent congestion which should let the sender aware of it and reduces its sending rate.

When a local recovery takes effect, we consider the following two cases. Firstly, the classic TCP sender does not see enough number of duplicate ACKs to trigger fast retransmit. This may be due to the local recovery procedures, which hides the out-of-order packet from receiver using mechanisms like reordering buffer at egress node. 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 the desirable throughput improvement when the recovered packet is lost due to non-persistent congestion. It solves the throughput problem mentioned in Section 3.3 and Section 4. However, it also brings the risk that the sender is not able to detect a real persistent congestion in time, and then overshooting may occur. Eventually a severe congestion that is not recoverable by a local recovery mechanism will be detected by sender. 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 and its variation trend.
Such measurement can help to determine if a lost packet by congestion. 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.

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 will be undone at the sender. This results in similar pros and cons as described earlier. Pros are preventing the unnecessary window reduction and improving the throughput when the loss is caused by non-congestive 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.

In summary, local recovery can improve Flow Completion Time (FCT) by eliminating tail loss in small flows. As it may change loss event to out-of-order event in most cases to TCP sender, if TCP sender uses loss based congestion control, there is no much throughput improvement. We suggest ECN and spurious retransmission to be enabled when local recovery is in use, it would give the desirable throughput performance, i.e. when loss is caused by congestion, reduce congestion window; otherwise keep sender’s sending rate. We do not suggest to use spurious retransmission alone together with local recovery as it may cause the TCP sender falsely undo window reduction when congestion occurs. If only ECN is enabled or neither ECN nor spurious retransmission is enabled, the throughput with local recovery in use is no much difference from that of the tradition TCP.

6.3. 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 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, as in SRv6 [I-D.ietf-6man-segment-routing-header]. In either way, LOOPS information is required to be embedded in some form to support the data plane measurement and feedback. Retransmission or FEC based loss recovery can be either per ON-hop or OE to OE based.

LOOPS alone has no setup requirement on control plane. Some overlay protocols, e.g., CAPWAP [RFC5415], has session setup phase, it can be used to exchange the information such as dynamic FEC parameters.

6.4. 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 techniques [RFC7149] or other approaches and is not a part of LOOPS. LOOPS is a set of functions to be implemented on Overlay Nodes, that will be involved in forwarding packets in a long haul overlay network. LOOPS targets the following features.

1. Local recovery: Retransmission, FEC, or combination thereof 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: Ingress/Egress overlay nodes measure the local segment RTT, loss and/or throughput to immediately get the overlay segment status.

3. Signal to end-to-end congestion control: Strategy to set ECN CE marking or simply not to recover the packet to signal the end host sender about if and/or how to adjust the sending rate is required.

7. Security Considerations

LOOPS does not require access to the traffic payload in clear, 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 and intercepted by malicious nodes. Means to ensure that only legitimate nodes are involved should be considered.
Denial of service attack can be launched from an ON. A rogue ON might be able to spoof packets 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.

8. IANA Considerations

No IANA action is required.

9. Acknowledgements

Thanks to etosat mailing list about the discussion about the SatCom and LOOPS use case.

10. Informative References


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

Mohamed Boucadair
Orange

Email: mohamed.boucadair@orange.com

Jianglong Wang
China Telecom

Email: wangjl11.bri@chinatelecom.cn