Internet Draft
Expires: August 2003

Spencer Dawkins
Cyneta Networks
Carl E. Williams
MCSR Labs
Alper E. Yegin
DoCoMo USA Labs

Framework and Requirements for TRIGTRAN

draft-dawkins-trigtran-framework-00.txt

Status of this Memo

Conventions used in this document:
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [ ].

Abstract

IETF-standardized unicast transport protocols have been designed to allow two end points to maintain communications by individually reacting to loss or degraded packet arrival times. Historically, those protocols have assumed loss is congestive and have reacted by decreasing the packet transmission rate to ease congestion. There are a number of cases, however, where these assumptions are incorrect, and one or more path segments present losses due to intermittent connectivity, a high uncorrected error rate, or the need for access path changes ("hand-off"s). Previous work [PILC] has addressed these conditions using end-to-end mechanisms. This draft examines the use of an on-path signaling mechanisms capable of providing advisory notifications for use in modifying the behavior of the transport in order to better respond to actual network conditions. This draft serves to create discussion in this area

as there are many ways to skin the cat. We are interested in hearing about them through open discussion.

List of Abbreviations

TRIGTRAN	Triggers for the Transport
AR	Access Router

1. Introduction

IETF transport protocol development has been based on the assumption that two communicating endpoints know more about characteristics of the paths between these endpoints than any single device within the network. Because IP datagrams can be forwarded over a variety of paths between two endpoints, a device within the network might have detailed knowledge of one path, but typically does not have detailed knowledge of all possible paths.

The scope of this work will focus on a framework for providing information to the transport via triggers of connection path characteristics. In particular, it is possible that a wireless access device might provide information about the path in a useful way because

(a) the wireless access device has detailed knowledge of a sub-network link, and

(b) it can still communicate with one endpoint when a problematic sub-network link stops working, or starts working, or changes its characteristics in some interesting way.

The goal here is that changes in path characteristics, especially in reachability, can be explicitly signaled expeditiously, while still relying on transport acknowledgements and timeouts.

If this goal is accepted, it may be broadened to include other sub-network events, if these sub-network events are generic in nature and accepted by the IETF community as a whole.

To further this goal this document will provide a basis of understanding of the following:

- The nature of generic "transport triggers"

- Possible uses of "transport triggers"

- Mechanisms for signaling transport triggers to accessible transport endpoints

- The architectural impact of this addition to the transport layer

Although the need for this change is more obvious in a wireless environment, we're also soliciting input from the rest of the Internet community in these areas:

- Whether there are "transport triggers" applicable to many sub-network types, beyond link up/link down

- Whether the use of "transport triggers" is worth the effort of modifying existing transport protocols to make use of this information

Why TRIGTRAN Isn't Fast Handoff

Transport triggers are similar to, but distinct from, similar discussions on triggers in MOBILEIP and in IRTF's Routing Research Group on micro-mobility. The primary difference is the low latency and tight coupling required for fast handoff. It is anticipated that the resulting model for defining transport triggers will provide a framework for future trigger discussion that are required for IP handoff protocols.

Why TRIGTRAN Isn't Wireless-only or TCP-only

Although TRIGTRAN is initially focusing on TCP connections over wireless sub-network links, we note that SCTP transports often have multiple wireline paths between two SCTP hosts for reliability. We don't want to do anything in TRIGTRAN that would prevent the use of TRIGTRAN as a notification mechanism for SCTP switchover - so please keep us honest!

This document describes a general framework and provides for a requirement list for the TRIGTRAN architecture in terms of notification events and protocol considerations. In the next section the authors provide a write-up on TRIGTRAN justification. This content may well end up in the problem space draft but the authors would like to include this discussion here for purposes of the BOF that is planned at IETF San Francisco.

2. Justification for TRIGTRAN

The variety of devices accessing the Internet, and the variety of access links they are using, continues to increase. At least some of these links exhibit characteristics that cause some Internet protocols, especially TCP [RFC793], to perform poorly.

Among these characteristics are:

1. Intermittent connectivity
2. Access path changes ("hand-offs")
3. High uncorrected error rate

For example, TCP congestion control [RFC2581] performs well over paths that lose traffic primarily because of congestion and buffer exhaustion, but performs poorly when TCP connections traverse links with high uncorrected error rates. Sending TCPs spend an inordinate amount of time waiting for acknowledgements that will not arrive, and then, although these losses are not due to congestion-related buffer exhaustion, the sending TCP transmits with a substantially reduced congestion window as it probes the network to determine its "safe" traffic level.

The root cause here is that TCP sees only one (implicit) signal about path conditions - packet loss - and can interpret this signal in only one way. The most conservative assumption is that packet loss is due to congestion, and for most of TCP's history, this conservative assumption was correct and sufficient. When transports traverse paths that include intermittent connectivity or other non-congestion "challenges", additional detection mechanisms are required.

TRIGTRAN ("Triggers for Transport") is a follow-up to the body of work done in the IETF's Performance of Link Characteristics(PILC) working group [PILC]. In PILC we were able to examine the specific TCP mechanisms that are problematic in environments with "challenged" links, and develop Best Current Practice specifications describing what can be done to mitigate these problems without introducing intermediate devices into the connection. PILC established the limits of "end to end" mechanisms with "challenged" links. With TRIGTRAN we are looking at advisory explicit notification ("hints") being initiated from an edge router to endpoint transport implementations across the Internet.

3. Strawman Framework

TRIGTRAN is focusing specifically on the case of problematic access links, because so many problematic links fall into this category (although not all problematic links are access links), and because this is the simplest useful case. More complex topologies are outside the scope of TRIGTRAN, at least for now.

In a nutshell, the minimal TRIGTRAN architecture looks like:

```
+------+ +-----------------+ (Internet    +------+
| Host | | TRIGTRAN Router |    goes      | Host |
+------+ +-----------------+     here)    +------+
        X                                    X
Sub-network Event ------------------>  Notifies Transport
      Here                                  Here
```

The critical feature here is that the host receiving a TRIGTRAN trigger is across an arbitrary network topology from the TRIGTRAN edge router sending the trigger. The host receiving the trigger then takes some transport-level action (sending a packet, retransmitting a packet, waiting for some period of time to transmit a packet, etc).

The transports would figure out "most events" eventually, given enough time (i.e., round trip times). For instance, TCP is good at figuring at bandwidth changes, but not as good at detecting a remote link transitioning to the "up" state after a retransmission timeout. Eventually, a backed-off RTO timer will fire, and the now-accessible receiver will acknowledge the next (successful) retransmission, but the sender and receiver will be waiting when they could be communicating.

TRIGTRAN can give the host receiving triggers hints that it might reattempt transmission, without waiting a complete RTO interval. TRIGTRAN is intended to provide network-based hints that clue the transport in more quickly (where "quickly" is measured in RTTs, not in milliseconds).

TRIGTRAN triggers are advisory in nature - they do not replace transport-level mechanisms (in the case of TCP, the receiver's ACK stream). Indeed, the TRIGTRAN architecture is a continuum of an existing body of work based on the principle that more and more often the network can clue a transport in on what is going on. Previous examples of "network-based clues" include ICMP Source Quench and Explicit Congestion Notification (ECN). These methods are a way for the transport to obtain more clues from the network but without relying exclusively on that information to function properly.

4. TRIGTRAN Protocol Principles/Considerations

* Transports can request trigger coverage from any adjacent access router, although only TRIGTRAN-aware access routers will provide trigger coverage. The host making this request is called the "TRIGTRAN Initiator".

* Correspondent hosts will request desired trigger notifications explicitly (they will not be sent to a correspondent host without prior arrangement).

* Trigger coverage requests and notification requests will be piggybacked on existing traffic ("setting a bit", not injecting new packets). The notifications themselves will be injected, of course.

* A TRIGTRAN-capable access router will inject trigger notifications. The exact structure of the notification is TBD.

* Triggers are per-host-pair over a specified interface - if a TRIGTRAN Initiator requests trigger coverage for any packets destined for a correspondent host, and the correspondent host expresses interest in receiving triggers, the TRIGTRAN-capable access router will send a single notification to the correspondent host.

* No reliability mechanism for triggers is defined. If a single trigger is lost for an event of interest to a transport, the transport will respond to the event using end-to-end mechanisms.

* TRIGTRAN registrations can be installed in one round trip (from the point of view of the TRIGTRAN Initiator).

* TRIGTRAN registrations install "soft state". TRIGTRAN Initiators must repeat coverage requests periodically, and correspondent hosts requesting trigger notifications must repeat this request periodically. The periodicity for these requests is TBD, but should be on the order of five minutes. The TRIGTRAN-capable access router will expire these requests after three of these time periods have elapsed.

* TRIGTRAN should work even if TRIGTRAN-capable access routers serve both hosts. Of course, each TRIGTRAN-capable access router will send triggers to the "correspondent host" adjacent to the other access router.

* TRIGTRAN is not a substitute for end-to-end mechanisms. TRIGTRAN triggers must be advising the correspondent host on something that it will figure out eventually without triggers.

* TRIGTRAN is per-transport-protocol. With two different transports running over some link, if both transports have requested trigger coverage, two separate triggers will be sent for a particular event.

* TRIGTRAN operations are not defined for an IP multicast address.

* Protocol notification message must contain enough information to identify per-host-pair.

* Trigger notifications are injected when a specified event is detected by the link-layer implementation on a TRIGTRAN-capable router for a specified link.

* A correspondent node may ignore notifications even though it may have requested trigger coverage for a TRIGTRAN Initiator.

* "Soft-state" for a per-host-pair should exist only at the adjacent TRIGTRAN-capable router only.

* When TRIGTRAN notifications and end-to-end mechanisms are in conflict the latter will take precedence over notifications.

* Triggers should be link-layer independent.

* Each TRIGTRAN notification will carry information for one event only. The correspondent node should be able to determine by an appropriate identifier field what event has taken place.

5. Trigger Events/notification

Presented in this section is an enumeration of the various triggers that encompass the framework. Motivation and suggested responses are provided for each trigger notification. This is a preliminary list of notifications and their associated suggested responses.

These triggers were identified during our work in PILC as things transports would WANT to know, but that are difficult to discover using end-to-end signaling. For instance, "Connectivity Interrupted" can't be signaled end-to-end, by definition.

Trigger: Connectivity Interrupted

Motivation:

When a link goes down TCP RTO exponential backoff occurs. The sender will eventually "give up", assuming that the receiving TCP (and perhaps the receiving host) will not recover.

Suggested Response:

The correspondent transport may choose to perform normal RTO processing for a longer period of time (in Solaris TCP, this would be a longer tcp_ip_abort_interval).

Note that a TCP that continues to receive ACKs should ignore this trigger.

Trigger: Connectivity Restored

Motivation:

When a link returns to working state, an other-end TCP may have experienced RTO, and may be waiting to attempt retransmission. Since TCP backs off exponentially (up to 64 seconds between retransmission attempts, in common implementations), the receiver will be waiting unnecessarily.

Suggested Response:

Attempt single-packet probe immediately, if successful,
resume perform normal operation.

Note that this attempt should be made only once per Connectivity Interrupted incident (clear when end-to-end ACKs have been received during retransmission).

Trigger: Packets Discarded by subnetwork, not lost due to congestion

Motivation:

In some wireless handoff scenarios, a subnetwork may explicitly discard packets at the "old" base station. In these cases, the application will either Fast Retransmit/Fast Recover or RTO/Slow Start (depending on whether additional ACKs are received for packets delivered by the new base station). These losses will reduce the congestion window, although they are not caused by congestion.

Suggested Response:

Retransmit without performing congestion avoidance. Note that this attempt should be made only once per loss event (in the document draft-allman-tcp-sack-13.txt, additional notifications would be ignored until the "scoreboard" data structure is emptied).

6. Security assessment and considerations for the TRIGTRAN framework

TRIGTRAN mechanisms provide explicit notifications from access routers to endpoint transport implementations that may be across the Internet.

The critical feature here is that the host receiving a TRIGTRAN trigger is across an arbitrary network topology from the access router sending the trigger, and the host receiving the trigger has no previous trust relationship with the access router. The host receiving the trigger will take some transport-level action (sending a packet, retransmitting a packet, waiting for some period of time to transmit a packet, etc.).

The transports would "figure out an event" eventually, given enough time. TRIGTRAN is intended to provide network-based hints that clue the transport in more quickly (where "quickly" is measured in RTTs, not in milliseconds). Since "link down" will probably be one of the triggers, end-to-end mechanisms cannot be used to send explicit notifications (since one of the ends isn't accessible).

A security assessment for TRIGTRAN amounts to evaluating what impact a forged trigger can have on a host that uses the hints to deal with the respective event. For example, we don't want TRIGTRAN to provide a mechanism for denial of service attacks, etc. (this should be obvious, but let's make it explicit).

TRIGTRAN triggers are advisory in nature - they do not replace transport-level mechanisms (in the case of TCP, the receiver's ACK stream). If a correspondent host gets a forged "Connectivity Interrupted" trigger, but continues to receive ACKs from the actually-reachable TRIGTRAN Initiator, the reasonable action is to ignore the trigger, not the ACKs. If a correspondent host gets a forged "Connectivity Restored" trigger, but does not receive ACKs from the actually-unreachable receiver, the transport would take its normal action for an unresponsive receiver (in the case of TCP, this would be RTO, retransmission, and slow start). The correspondent host can use existing transport-level mechanisms to determine the validity of the trigger. Because TRIGTRAN triggers are advisory the correspondent host isn't required to act as if the events are real. Thus, we don't think a security association is required between the TRIGTRAN router and the correspondent host receiving triggers. If one is present, fine, but it's not required.

The alternative, requiring the host to establish trust relationships with arbitrary routers in other administrative domains in order to receive triggers, seems to be overkill in this situation. If TRIGTRAN triggers overrode end-to-end mechanisms, a trust relationship would clearly be required.

We note that, in the absence of trust relationships between TRIGTRAN Initiators and TRIGTRAN routers, it's possible for forged packets to fill up the TRIGTRAN router's "soft state" notification table. If we are true to our self-imposed restriction that all triggers would be advisory in nature, a denial-of-service attack would have the effect of disabling TRIGTRAN, and normal end-to-end mechanisms would prevail - as they do today.

Our self-imposed limit to access routers for our initial work may help here - the access router would have some ability to "ingress-filter" trigger coverage requests, as edge routers filter on IP address prefixes today.

7. Why TRIGTRAN is Not Doomed

At the IETF 55 TRIGTRAN BoF, Sally Floyd presented a number of questions for TRIGTRAN. One of the most relevant was "ICMP Source Quench failed. P-MTU Discovery failed. Why will TRIGTRAN be different?"

This question needs to be answered. Our crack at an answer follows.

7.1 Why TRIGTRAN Is Not Doomed (Source Quench)

RFC 792 describes the Internet Control Management Protocol (ICMP). ICMP includes a message type called "Source Quench" (Type 4). Source Quench was intended to provide "back pressure"

when a gateway discards an incoming IP datagram because no buffers are available. The message is sent to the Source IP Address carried in the discarded IP datagram. Conceptually, a host receiving an ICMP Source Quench message would slow down its sending rate until it stopped receiving ICMP Source Quench messages, and then gradually increase its sending rate.

The original specification did not provide quantitative guidance on HOW MUCH to slow down. RFC 1016 proposed a formula Source Quench Induced Delay ("SQuID"), but this RFC was published six years after RFC 792, and defined itself as a "crazy idea". A better characterization might have been "embryonic", reflecting an unsophisticated awareness of congestion - TCP didn't include Slow Start/Congestion Avoidance until a couple of years later.

The original specification allowed gateways to generate an ICMP Source Quench for every datagram dropped, but did not require gateways to send them at all. RFC 1009 ("Requirements for Internet Gateways") required gateways to include the capability to send rate-limited ICMP Source Quench messages, but when it was updated as RFC 1812 ("Requirements for IP Version 4 Routers"), this requirement was dropped in favor of deprecating ICMP Source Quench ("Gateways SHOULD NOT"). The reasons given for this about-face included:

- ICMP Source Quench affected only packets sent from the host generating the "over the top" packet, so did not provide a fair mechanism for hosts sharing overcommitted network paths, and

- ICMP Source Quench added (reverse-direction) packets to the network during congestion events, and used router memory and processing power to construct and send ICMP Source Quenches during congestion events.

The overwhelming problem with ICMP Source Quench wasn't that it required gateways to send a "trigger" to hosts - it had problems with unfairness and inefficiency. Since the specifications omitted critical details and didn't require this functionality, hosts were forced to add end-to-end congestion avoidance at the transport layer, anyway.

This experience isn't terrifically relevant to TRIGTRAN, because standards-based recommendations have vacillated from MAY to SHOULD NOT - hardly an overwhelming motivator to deployment!

7.2 Why TRIGTRAN is Not Doomed (Path Maximum Transmission Unit Discovery)

RFC 791 ("Internet Protocol") allows IP packets of up to 65,535 octets, but subnetworks typically don't support frames with a payload this large. A host can know the Maximum Transmission Unit size for its local subnetwork, but can't be sure that a path across multiple subnetworks will support a larger MTU without IP fragmentation. RFC 1122 ("Requirements for Internet Hosts -- Communication Layers") specified that IP implementations "SHOULD" use an MTU of 576 or less to communicate with hosts on a different network, unless the implementation "knew" that the path supported larger MTUs.

RFC 1063 ("IP MTU Discovery Options") and its successor RFC **1191 ("Path MTU Discovery") described a mechanism to gain this** knowledge. An IP implementation wishing to use large MTUs sent a packet of the desired size into the network with the "Don't Fragment" bit set. If the packet encountered a subnetwork that didn't support the desired MTU size, the gateway for that link discarded the packet, and reported an ICMP ICMP Destination Unreachable error with a code meaning "fragmentation needed and DF set", (also described as "Datagram Too Big"), and an indication of the bottleneck MTU size, stored in a previously-unused ICMP header field. To avoid requiring a "flag day" for P-MTU discovery, hosts were required to accept "Datagram Too Big" errors that didn't include the bottleneck MTU size, and to either fall back to 576 bytes or search with a new, lower P-MTU "guess", thus accommodating gateways that hadn't been updated.

As P-MTU Discovery was deployed, another "discovery" happened. As described in RFC 1435 ("IESG Advice from Experience with Path MTU Discovery"), "some vendors have added to their routers the ability to disable ICMP messages generated by the router". The effect was that of a "black hole" - the router dropped the "too big" datagram, but did not send any notification to the sending host that this was happening. Further deployment experience led to RFC 2923 ("TCP Problems with Path MTU Discovery"), pointing out that "black holing" was still happening, and that routers were configured this way for a variety of reasons, including a misguided attempt to shut down ICMP messages crossing firewalled administrative domains. Procedures for hosts encountering "black holes" were described, but guessing too high on a "black-holed" path still leads to delays of several seconds as the host TCP implementation uses timeouts to detect failure and "falls back" to 576 bytes.

The Internet experience with P-MTU Discovery is more relevant to TRIGTRAN if TRIGTRAN considers ICMP messages as a trigger signal mechanism, although the "black holing" problem is less severe because TRIGTRAN triggers are advisory in nature. End-to-end mechanisms will lead to the same result after some delay.

The history of P-MTU Discovery is useful as a reminder that TRIGTRAN triggers must traverse firewalls, no matter what mechanism is defined to transport them.

8. Summary

While the draft is initially focusing on wireless links, other link types (i.e. modems) are of importance and will be taken into account. Moving forward with this work an eye on non-wireless links will also be taken into account.

There are many ways to skin the cat and we are interested in hearing about alternatives. The draft was put together as the output from the IETF TRIGTRAN BOF in Atlanta 2002. This is a preliminary writeup whose purpose is to facilitate the discussion of a second BOF in San Francisco IETF in March 2003. The preliminary thinking in this draft would serve to create additional discussion highlighting issues and hopefully asking the right questions.

9. Acknowledgements

Thanks to Ted Hardie for coming up with a good abstract and other comments on the draft. Thanks to Sally Floyd for numerous comments and questions raised at the IETF Atlanta BOF on TRIGTRAN that structured much of the text for this draft as well as her insightful review of this draft. Thanks to Mark Allman, Aaron Falk, and John Wroclawski for their inputs on this draft and contributions on the mailing list on this subject matter. Also, thanks to those who have contributed to the IETF Atlanta BOF discussion and comments on the TRIGTRAN mailing list. Special thanks to Allison Mankin for her vision and leadership on dealing with this problem space.

10. References

[BAR-BOF] Notes from L2 Triggers meeting (PILC mailing list posting), Aaron Falk and Carl E. Williams, April 2002, available from http://pilc.grc.nasa.gov/list/archive/1837.html.

Several of the following drafts describe lower-latency triggers intended for Mobile IP fast handoff. TRIGTRAN reuses a number of concepts from this work.

[CORSON] A Triggered Interface (work in progress), Scott Corson, May 2002, draft-corson-triggered-00.txt

[WILLIAMS]Problem Statement for Link-layer Triggers work (work in progress), Carl Williams, Alper E. Yegin, and James Kempf, June 2002, draft-williams-l2-probstmt-00.txt

[YEGIN] Link-layer Triggers Protocol (work in progress), Alper Yegin, June 2002, draft-yegin-l2-triggers-00.txt

[GURI] Layer-2 API for Paging (expired work in progress), Sridhar Gurivireddy, Behcet Sarikaya, Vinod Choyi, and Xiafeng Xu, October 2001, draft-guri-seamoby-paging-triggers-00.txt

[MANYFOLKS] Supporting Optimized Handover for IP Mobility : Requirements for Underlying Systems (work in progress), Alper Yegin (editor), June 2002, draft-manyfolks-l2-mobilereq-02.txt

[PILC] Performance Implications of Link Characteristics IETF Working Group http://www.ietf.org/html.charters/pilc-charter.html

[RFC896] Congestion Control in IP/TCP Internetworks, IETF RFC 896, January 1984, John Nagle.

[RFC792] Internet Control Message Protocol, IETF RFC 792, September 1981, Jon Postel.

[RFC1016] Something a Host Could do with Source Quench: The Source Quench Introduced Delay (SQuID), IETF RFC 1016, July 1987, Jon Postel.

[RFC1009] Requirements for Internet Gateways, IETF RFC 1009, R.Braden and Jon Postel, June 1987.

[RFC1812] Requirements for IP version 4 Routers, IETF RFC 1812, Editor: Fred Baker, 1995.

[RFC791] Internet Protocol, IETF RFC 791, September 1981, Editor: Jon Postel.

[RFC1122] Requirements for Internet Hosts ? Communications Layers, IETF 1122, October 1989, Editor: R. Braden.

[RFC1063] IP MTU Discovery Options, IETF RFC 1063, July 1988, J. Mongul, C. Kent, C. Partridge, K. McCloghrie.

[RFC1191] PATH MTU Discovery, IETF RFC 1191, November 1990, J. Mogul, S. Deering.

[RFC1435] IESG Advice from Experience with Path MTU Discovery, March 1993, FTP Software.

[RFC2923] TCP Problems with PATH MTU Discovery, September 2000, K. Lahey.

11. Contact Information

Spencer Dawkins
Cyneta Networks
1201 North Bowser Road Phone: 1-469-385-2484
Suite 100
Richardson, Texas 75081
USA Email: sdawkins@cynetanetworks.com

Carl E. Williams
MCSR Labs
3790 El Camino Real
Palo Alto, CA 94306 Phone: 1-650-279-5903
USA Email: carlw@mcsr-labs.org

Alper E. Yegin
DoCoMo USA Labs
181 Metro Drive, Suite 300 Phone: +1 408 451 4743
San Jose, CA 95110 Fax: +1 408 451 1090
USA Email: alper@docomolabs-usa.com