Transport Area Working Group
Internet-Draft
Obsoletes: 4170 (if approved)
Intended status: Best Current Practice
Expires: January 12, 2014

J. Saldana
University of Zaragoza
D. Wing
Cisco Systems
J. Fernandez Navajas
University of Zaragoza
Muthu. Perumal
Cisco Systems
F. Pascual Blanco
Telefonica I+D

July 11, 2013

Tunneling Compressed Multiplexed Traffic Flows (TCM-TF) draft-saldana-tsvwg-tcmtf-05

Abstract

Tunneling Compressed and Multiplexed Traffic Flows (TCM-TF) is a method for improving the bandwidth utilization of network segments that carry multiple flows in parallel sharing a common path. The method combines standard protocols for header compression, multiplexing, and tunneling over a network path for the purpose of reducing the bandwidth used when multiple flows are carried over that path. The amount of packets per second can also be reduced.

This document describes the TCM-TF framework and the different options which can be used for each layer (header compression, multiplexing and tunneling).

Status of This Memo

This Internet-Draft is submitted to IETF 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 http://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 12, 2014.

Table of Contents

1. Introduction

This document describes a way to combine existing protocols for header compression, multiplexing and tunneling to save bandwidth for applications that generate long-term flows of small packets.

1.1. Requirements Language

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 [RFC2119].

1.2. Bandwidth efficiency of flows sending small packets

The interactivity demands of some real-time services (VoIP, videoconferencing, telemedicine, video vigilance, online gaming, etc.) require a traffic profile consisting of high rates of small packets, which are necessary in order to transmit frequent updates between the two extremes of the communication. These services also demand small network delays. In addition, some other services also use small packets, although they are not delay-sensitive (e.g., instant messaging, M2M packets sending collected data in sensor networks using wireless or satellite scenarios). For both the delay-sensitive and delay-insensitive applications, their small data payloads incur significant overhead.

When a number of flows based on small packets (small-packet flows) share the same path, bandwidth can be saved by multiplexing packets belonging to different flows. If a transmission queue has not already been formed but multiplexing is desired, it is necessary to add a multiplexing delay, which has to be maintained under some threshold if the service presents tight delay requirements.

1.2.1. Real-time applications using RTP

The first design of the Internet did not include any mechanism capable of guaranteeing an upper bound for delivery delay, taking into account that the first deployed services were e-mail, file transfer, etc., in which delay is not critical. RTP [RTP] was first defined in 1996 in order to permit the delivery of real-time contents. Nowadays, although there are a variety of protocols used for signaling real-time flows (SIP [SIP], H.323, etc.), RTP has become the standard par excellence for the delivery of real-time content.

RTP was designed to work over UDP datagrams. This implies that an IPv4 packet carrying real-time information has to include 40 bytes of headers: 20 for IPv4 header, 8 for UDP, and 12 for RTP. This overhead is significant, taking into account that many real-time services send very small payloads. It becomes even more significant

with IPv6 packets, as the basic IPv6 header is twice the size of the IPv4 header. Table 1 illustrates the overhead problem of VoIP for two different codecs.

IPv4	IPv6
IPv4+UDP+RTP: 40 bytes header	IPv6+UDP+RTP: 60 bytes header
G.711 at 20 ms packetization: 25% header overhead	G.711 at 20 ms packetization: 37.5% header overhead
G.729 at 20 ms packetization: 200% header overhead	G.729 at 20 ms packetization: 300% header overhead

Table 1: Efficiency of different voice codecs

1.2.2. Real-time applications not using RTP

At the same time, there are many real-time applications that do not use RTP. Some of them send UDP (but not RTP) packets, e.g., First Person Shooter (FPS) online games, for which latency is very critical.

There is also another kind of applications which have been reported as real-time using TCP: MMORPGs (Massively Multiplayer Online Role Playing Games), which in some cases have millions of players, thousands of them sharing the same virtual world. They use TCP packets to send the player commands to the server, and also to send to the player's application the characteristics and situation of other gamers' avatars. These games do not have the same interactivity of FPSs, but the quickness and the movements of the players are important, and can decide if they win or lose a fight.

Finally, there is also another fact which has to be taken into account: TCP is getting used for media delivery. For many reasons, such as avoiding firewalls, the standard RTP/UDP/IP protocol stack is substituted in many cases by FLV/HTTP/TCP/IP (FLash Video [FLV]).

1.2.3. Other applications generating small packets

Other applications without delay constraints are also becoming popular (e.g., instant messaging, M2M packets sending collected data in sensor networks using wireless or satellite scenarios). The number of wireless M2M (machine-to-machine) connections is steady growing since a few years, and a share of these is being used for delay-intolerant applications, e.g., industrial SCADA (Supervisory Control And Data Acquisition), power plant monitoring, smart grids, asset tracking.

1.2.4. Optimization of small-packet flows

In order to mitigate the bad network efficiency of flows using small packets, the multiplexing of a number of payloads into a single packet can be considered as a solution. For example, if a single VoIP flow is considered, the number of samples included in a packet can be increased, but at the cost of adding new packetization delays.

When a number of flows share the same path between an origin and a destination, a multiplexer can build a bigger packet in which a number of payloads share a common header. A demultiplexer is necessary at the end of the common path, so as to rebuild the packets as they were originally sent, making multiplexing a transparent process for the extremes of the flow.

In addition, the headers of the original packets can be compressed to save more bandwidth, using some of the already existing header compression standards ([cRTP], [ECRTP], [IPHC], [ROHC]). When different headers are compressed together, tunneling can be used to relieve intermediate routers from the decompression and compression processing.

If only one stream is tunneled and compressed, then little bandwidth savings will be obtained. In contrast, multiplexing is helpful to amortize the overhead of the tunnel header over many payloads. The obtained savings grow with the number of flows optimized together.

1.3. Terminology

This document uses a number of terms to refer to the roles played by the entities using TCM-TF.

o native packet

A packet sent by an application, belonging to a flow that can be optimized by means of TCM-TF.

o native flow

A flow of native packets. It can be considered a "small-packet flow" when the vast majority of the generated packets present a low payload-to-header ratio.

o TCM packet

A packet including a number of multiplexed and header-compressed native ones, and also a tunneling header.

- TCM flow

A flow of TCM packets, including a number of optimized native flows.

- TCM optimizer

The host where TCM optimization is deployed. It corresponds to both the ingress and the egress of the tunnel transporting the compressed and multiplexed packets.

If the optimizer compresses headers, multiplexes packets and creates the tunnel, it behaves as a "TCM-ingress optimizer", or "TCM-IO". It takes native packets or flows and "optimizes" them.

If it extracts packets from the tunnel, demultiplexes packets and decompresses headers, it behaves as a "TCM-egress optimizer", or "TCM-EO". The TCM-egress optimizer takes a TCM flow and "rebuilds" the native packets as they were originally sent.

- TCM-TF session

The relationship between a pair of TCM optimizers exchanging TCM packets.

- policy manager

A network entity which makes the decisions about TCM-TF parameters: multiplexing period to be used, flows to be optimized together, depending on their IP addresses, ports, etc. It is connected with a number of TCM-TF optimizers, and orchestrates the optimization that takes place between them.

1.4. Scenarios of application

Different scenarios of application can be considered for the tunneling, compressing and multiplexing solution:

1.4.1. Residential scenario

If we consider the residential users of a real-time interactive application (e.g., VoIP, an online game generating small packets) in a town or a district, a TCM optimizing module can be included in network devices, in order to group packets with the same destination. Depending on the number of users of the application, the packets could be grouped at different levels in DSL fixed network scenarios, at gateway level in LTE mobile network scenarios or even in other ISP edge routers. TCM-TF may also be applied for fiber residential accesses, and in 2G/3G mobile networks. This would reduce bandwidth

requirements in the provider aggregation network, and in the network connection of the application provider, thus resulting in savings for both actors.

In this scenario, agreements between different companies can be established in order to save bandwidth and to reduce packets per second. For example, a service provider (e.g., an online gaming company) could be allowed to place a TCM optimizer in the aggregation network of an ISP, being able to optimize all the flows of a game or service. Another TCM optimizer would rebuild these packets once they arrive to the network of the provider.

At the same time, the ISP would implement TCM-TF capabilities within its own MPLS network in order to optimize internal network resources: optimizing modules could be embedded in the Label Edge Routers of the network. In that scenario MPLS would be the "tunneling" layer, being the tunnels the paths defined by the MPLS labels and avoiding the use of other tunneling protocols.

Finally, some networks use cRTP [cRTP] on their access links. This gives bandwidth savings on the access link, but as a counterpart it consumes considerable CPU resources on the aggregation router. In these cases, by means of TCM, instead of only saving bandwidth on the access link, it could also be saved across the core, and on the far-end access link, all without the CPU impact on the aggregation router.

1.4.2. Corporate environments

End users can also optimize traffic end-to-end from network borders. As an example, we can consider the case of an enterprise with a number of distributed central offices, in which an appliance could be placed next to the access router, being able to optimize traffic flows with a shared origin and destination. Thus, a number of remote desktop sessions to the same server could be optimized, or a number of VoIP calls between two offices could also require less bandwidth and fewer packets per second.

Another example of an end user collaborating in traffic optimization could be an Internet cafe, which is suitable of having many users of the same application (e.g., a game) sharing the same access link. Internet cafes are very popular in countries with relatively low access speeds in households, where home computer penetration is usually low as well. In many of these countries, bandwidth can become a serious limitation for this kind of business, so TCM-TF savings may become critical for their viability.

Satellite communication links that often manage the bandwidth by limiting the transmission rate, measured in packets per second (pps), to and from the satellite. Applications like VoIP that generate a large number of small packets can easily fill the maximum number of pps slots, limiting the throughput across such links. As an example, a G.729a voice call generates 50 pps at 20 ms packetization time. If the satellite transmission allows 1,500 pps, the number of simultaneous voice calls is limited to 30. This results in poor utilization of the satellite link's bandwidth as well as places a low bound on the number of voice calls that can utilize the link simultaneously. TCM optimization of small packets into one packet for transmission would improve the efficiency. Satellite links would also find it useful to multiplex and compress small TCP packets into one packet. This could be especially interesting for compressing TCP ACKs.

Desktop or application sharing where the traffic from the server to the client typically consists of the delta of screen updates. Also, the standard for remote desktop sharing emerging for WebRTC in the RTCWEB Working Group is: {something}/SCTP/UDP (Stream Control Transmission Protocol [SCTP]). In this scenario, SCTP/UDP could be used in other cases: chatting, file sharing and applications related to WebRTC peers. There could be hundreds of clients at a site talking to a server located at a datacenter over a WAN. Compressing, multiplexing and tunneling this traffic could save WAN bandwidth and potentially improve latency.

1.4.3. Machine to Machine (M2M) scenario

In a M2M/SCADA (Supervisory Control And Data Acquisition) context, TCM optimization can be applied when a satellite link is used for collecting the data of a number of sensors. M2M terminals are normally equipped with sensing devices which can interface to proximity sensor networks through wireless connections. The terminal can send the collected sensing data using a satellite link connecting to a satellite gateway, which in turn will forward the M2M/SCADA data to the to the processing and control center through Internet. The size of typical M2M application transaction depends on the specific service and it may vary from a minimum of 20 bytes (e.g., tracking and metering in private security) to about 1,000 bytes (e.g., video-surveillance). In this context, TCM-TF concepts can be also applied to allow a more efficient use of the available satellite link capacity, matching the requirements demanded by some M2M services. If the case of large sensor deployments is considered, where proximity sensor networks transmit data through different satellite terminals, the use of compression algorithms already available in current satellite systems to reduce the overhead introduced by TCP or UDP and IPv6 protocols is certainly desirable. In addition to this,

tunneling and multiplexing functions available from TCM-TF allows extending compression functionality throughout the rest the network, to eventually reach the processing and control centers.

1.5. Potential beneficiaries of TCM optimization

In conclusion, a standard able to compress headers, multiplex a number of packets and send them together using a tunnel, can benefit various stakeholders:

- network operators can compress traffic flows sharing a common network segment;
- ISPs;
- developers of VoIP systems can include this option in their solutions;
- service providers, who can achieve bandwidth savings in their supporting infrastructures.

Other fact that has to be taken into account is that the technique not only saves bandwidth but also reduces the number of packets per second, which sometimes can be a bottleneck for a satellite link or even for a network router.

1.6. Current Standard

The current standard [TCRTP] defines a way to reduce bandwidth and pps of RTP traffic, by combining three different standard protocols:

- Regarding compression, [ECRTP] is the selected option.
- Multiplexing is accomplished using PPP Multiplexing [PPP-MUX]
- Tunneling is accomplished by using L2TP (Layer 2 Tunneling Protocol [L2TPv3]).

The three layers are combined as shown in the Figure 1:

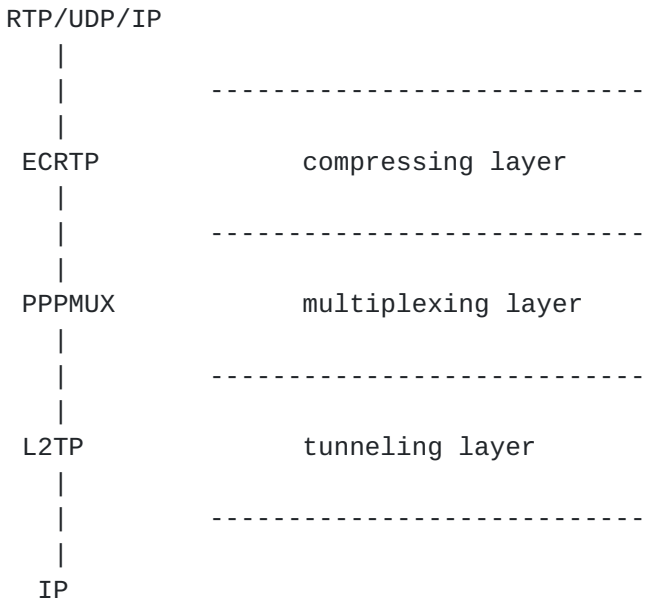

Figure 1

1.7. Improved Standard Proposal

In contrast to the current standard [TCRTP], TCM-TF allows other header compression protocols in addition to RTP/UDP, since services based on small packets also use by bare UDP or TCP, as shown in Figure 2:

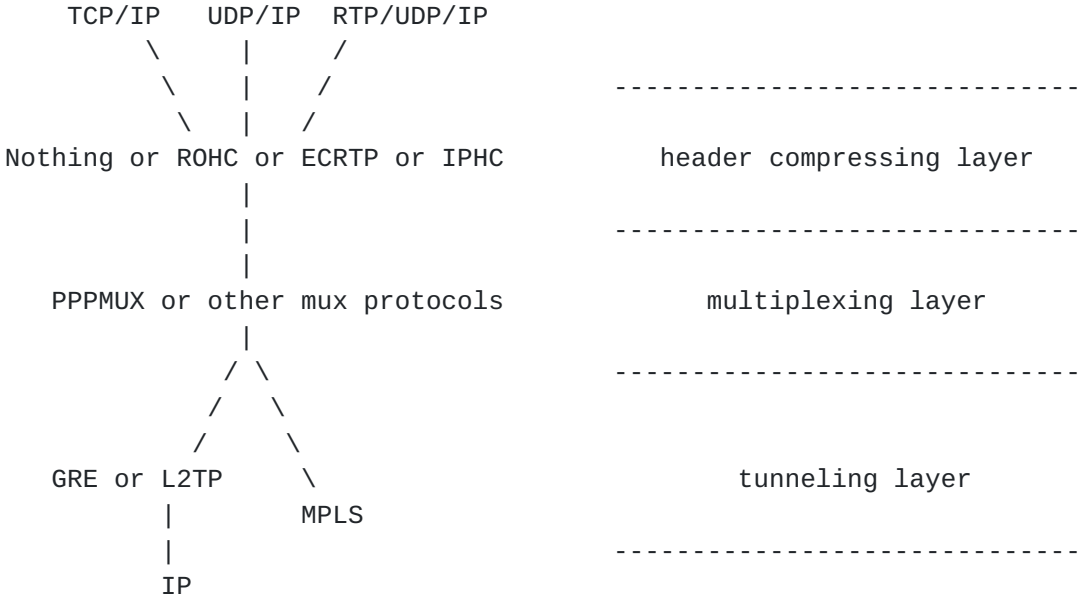

Figure 2

Each of the three layers is considered as independent of the other two, i.e. different combinations of protocols can be implemented according to the new proposal:

- Regarding compression, a number of options can be considered: as different standards are able to compress different headers ([cRTP], [ECRTP], [IPHC], [ROHC]). The one to be used could be selected depending on the traffic to compress and the concrete scenario (packet loss percentage, delay, etc.). It also exists the possibility of having a null header compression, in the case of wanting to avoid traffic compression, taking into account the need of storing a context for every flow and the problems of context desynchronization in certain scenarios. Although non shown in Figure 2, ESP (Encapsulating Security Payload [ESP]) headers can also be compressed.

- Multiplexing is also accomplished using PPP Multiplexing [PPP-MUX]. Nevertheless, other multiplexing protocols can also be considered.

- Tunneling is accomplished by using L2TP (Layer 2 Tunneling Protocol [L2TPv3]) over IP, GRE (Generic Routing Encapsulation [GRE]) over IP, or MPLS (Multiprotocol Label Switching Architecture [MPLS]).

It can be observed that TCRTP [TCRTP] is included as an option in TCM-TF, combining [ECRTP], [PPP-MUX] and [L2TPv3].

Payload compression schemes could also be used, but they are not the aim of this document.

2. Protocol Operation

This section describes how to combine protocols belonging to trhee layers (compressing, multiplexing, and tunneling), in order to save bandwidth for the considered flows.

2.1. Models of implementation

TCM-TF can be implemented in different ways. The most straightforward is to implement it in the devices terminating the flows (these devices can be e.g., voice gateways, or proxies grouping a number of flows):

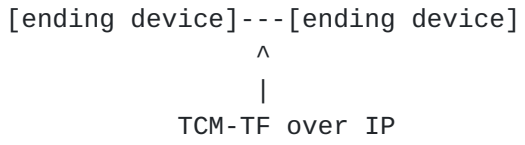

Figure 3

Another way TCM-TF can be implemented is with an external concentration device. This device could be placed at strategic places in the network and could dynamically create and destroy TCM-TF sessions without the participation of the endpoints that generate the flows (Figure 4).

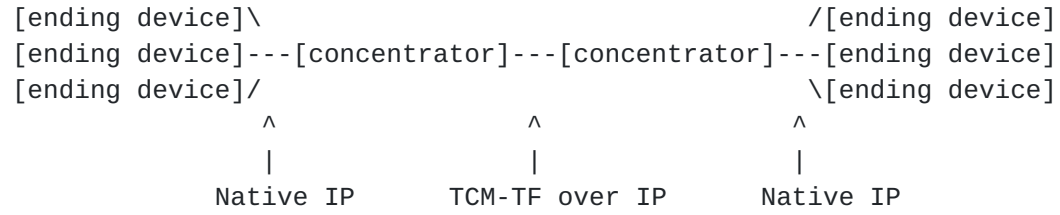

Figure 4

A number of already compressed flows can also be merged in a tunnel using a concentrator in order to increase the number of flows in a tunnel (Figure 5):

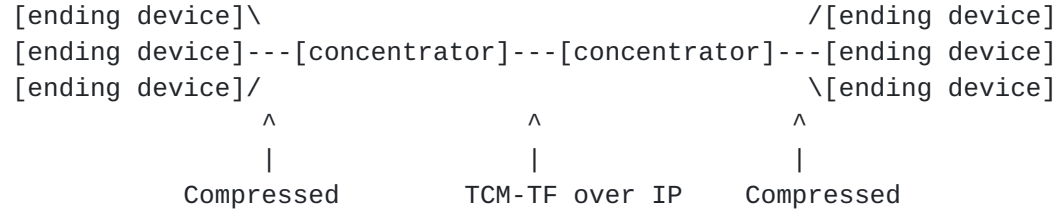

Figure 5

2.2. Choice of the compressing protocol

There are different protocols that can be used for compressing IP flows:

- IPHC (IP Header Compression [IPHC]) permits the compression of TCP /IP, UDP/IP and ESP/IP headers. It has a low implementation complexity. On the other hand, the resynchronization of the context can be slow over long RTT links. It should be used in scenarios presenting very low packet loss percentage.

- cRTP (compressed RTP [cRTP]) works the same way as IPHC, but is also able to compress RTP headers. The link layer transport is

not specified, but typically PPP is used. For cRTP to compress headers, it must be implemented on each PPP link. A lot of context is required to successfully run cRTP, and memory and processing requirements are high, especially if multiple hops must implement cRTP to save bandwidth on each of the hops. At higher line rates, cRTP's processor consumption becomes prohibitively expensive. cRTP is not suitable over long-delay WAN links commonly used when tunneling, as proposed by this document. To avoid the per-hop expense of cRTP, a simplistic solution is to use cRTP with L2TP to achieve end-to-end cRTP. However, cRTP is only suitable for links with low delay and low loss. Thus, if multiple router hops are involved, cRTP's expectation of low delay and low loss can no longer be met. Furthermore, packets can arrive out of order.

- ECRTP (Enhanced Compressed RTP [ECRTP]) is an extension of cRTP [cRTP] that provides tolerance to packet loss and packet reordering between compressor and decompressor. Thus, ECRTP should be used instead of cRTP when possible (e.g., the two TCM optimizers implementing ECRTP).

- ROHC (RObust Header Compression [ROHC]) is able to compress TCP/IP, UDP/IP, ESP/IP and RTP/UDP/IP headers. It is a robust scheme developed for header compression over links with high bit error rate, such as wireless ones. It incorporates mechanisms for quick resynchronization of the context. It includes an improved encoding scheme for compressing the header fields that change dynamically. Its main drawback is that it requires significantly more processing and memory resources than the ones necessary for IPHC or ECRTP.

This standard does not determine which of the existing protocols has to be used for the compressing layer. The decision will depend on the scenario, and will mainly be determined by the packet loss probability, RTT, and the availability of memory and processing resources. The standard is also suitable to include other compressing schemes that may be further developed.

2.2.1. Context Synchronization in ECRTP

When the compressor receives an RTP packet that has an unpredicted change in the RTP header, the compressor should send a COMPRESSED_UDP packet (described in [ECRTP]) to synchronize the ECRTP decompressor state. The COMPRESSED_UDP packet updates the RTP context in the decompressor.

To ensure delivery of updates of context variables, COMPRESSED_UDP packets should be delivered using the robust operation described in [ECRTP].

Because the "twice" algorithm described in [ECRTP] relies on UDP checksums, the IP stack on the RTP transmitter should transmit UDP checksums. If UDP checksums are not used, the ECRTP compressor should use the cRTP Header checksum described in [ECRTP].

2.2.2. Context Synchronization in ROHC

ROHC [ROHC] includes a more complex mechanism in order to maintain context synchronization. It has different operation modes and defines compressor states which change depending on link behavior.

2.3. Multiplexing

Header compressing algorithms require a layer two protocol that allows identifying different protocols. PPP [PPP] is suited for this, although other multiplexing protocols can also be used for this layer of TCM-TF.

When header compression is used inside a tunnel, it reduces the size of the headers of the IP packets carried in the tunnel. However, the tunnel itself has overhead due to its IP header and the tunnel header (the information necessary to identify the tunneled payload).

By multiplexing multiple small payloads in a single tunneled packet, reasonable bandwidth efficiency can be achieved, since the tunnel overhead is shared by multiple packets belonging to the flows active between the source and destination of an L2TP tunnel. The packet size of the flows has to be small in order to permit good bandwidth savings.

If the source and destination of the tunnel are the same as the source and destination of the compressing protocol sessions, then the source and destination must have multiple active small-packet flows to get any benefit from multiplexing.

Because of this, TCM-TF is mostly useful for applications where many small-packet flows run between a pair of hosts. The number of simultaneous sessions required to reduce the header overhead to the desired level depends on the average payload size, and also on the size of the tunnel header. A smaller tunnel header will result in fewer simultaneous sessions being required to produce adequate bandwidth efficiencies.

2.4. Tunneling

Different tunneling schemes can be used for sending end to end the compressed payloads.

2.4.1. Tunneling schemes over IP: L2TP and GRE

L2TP tunnels should be used to tunnel the compressed payloads end to end. L2TP includes methods for tunneling messages used in PPP session establishment, such as NCP (Network Control Protocol). This allows [IPCP-HC] to negotiate ECRTP compression/decompression parameters.

Other tunneling schemes, such as GRE [GRE] may also be used to implement the tunneling layer of TCM-TF.

2.4.2. MPLS tunneling

In some scenarios, mainly in operator's core networks, the use of MPLS is widely deployed as data transport method. The adoption of MPLS as tunneling layer in this proposal intends to natively adapt TCM-TF to those transport networks.

In the same way that layer 3 tunnels, MPLS paths, identified by MPLS labels, established between Label Edge Routers (LSRs), could be used to transport the compressed payloads within an MPLS network. This way, multiplexing layer must be placed over MPLS layer. Note that, in this case, layer 3 tunnel headers do not have to be used, with the consequent data efficiency improvement.

2.5. Encapsulation Formats

The packet format for a packet compressed is:

Figure 6

The packet format of a multiplexed PPP packet as defined by [PPP-MUX] is:

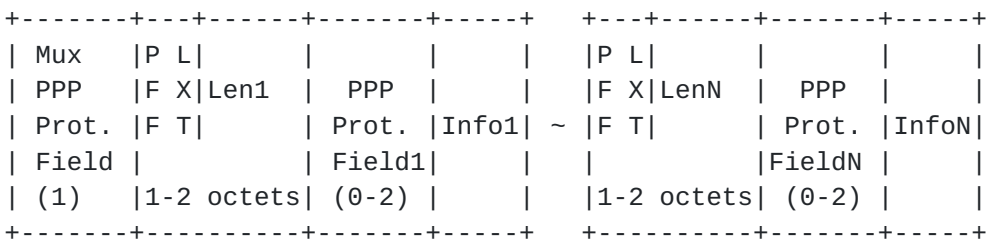

Figure 7

The combined format used for TCM-TF with a single payload is all of the above packets concatenated. Here is an example with one payload, using L2TP or GRE tunneling:

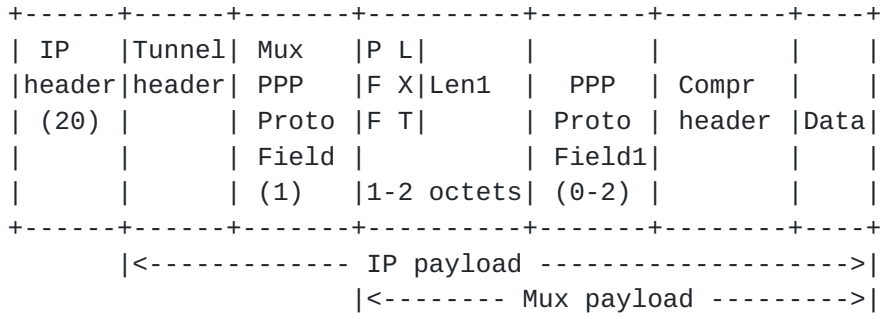

Figure 8

If the tunneling technology is MPLS, then the scheme would be:

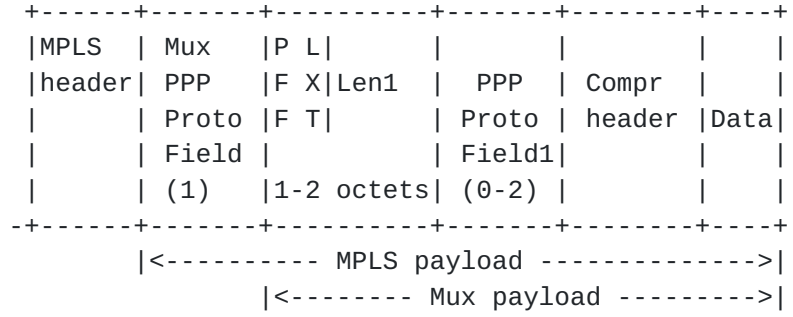

Figure 9

If the tunnel contains multiplexed traffic, multiple "PPPMux payload"s are transmitted in one IP packet.

3. Contributing Authors

Gonzalo Camarillo
Ericsson
Advanced Signalling Research Lab.
FIN-02420 Jorvas
Finland

Email: Gonzalo.Camarillo@ericsson.com

Michael A. Ramalho
Cisco Systems, Inc.
8000 Hawkins Road
Sarasota, FL 34241-9300
US

Phone: +1.732.832.9723
Email: mramalho@cisco.com

Jose Ruiz Mas
University of Zaragoza
Dpt. IEC Ada Byron Building
50018 Zaragoza
Spain

Phone: +34 976762158
Email: jruiz@unizar.es

Diego Lopez Garcia
Telefonica I+D
Ramon de la cruz 84
28006 Madrid
Spain

Phone: +34 913129041
Email: diego@tid.es

David Florez Rodriguez
Telefonica I+D
Ramon de la cruz 84
28006 Madrid
Spain

Phone: +34 91312884

Email: dflorez@tid.es

Manuel Nunez Sanz
Telefonica I+D
Ramon de la cruz 84
28006 Madrid
Spain

Phone: +34 913128821
Email: mns@tid.es

Juan Antonio Castell Lucia
Telefonica I+D
Ramon de la cruz 84
28006 Madrid
Spain

Phone: +34 913129157
Email: jacl@tid.es

Mirko Suznjevic
University of Zagreb
Faculty of Electrical Engineering and Computing, Unska 3
10000 Zagreb
Croatia

Phone: +385 1 6129 755
Email: mirko.suznjevic@fer.hr

4. Acknowledgements

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

The most straightforward option for securing a number of non-secured flows sharing a path is by the use of IPsec [IPsec], when TCM using an IP tunnel is employed. Instead of adding a security header to the packets of each native flow, and then compressing and multiplexing them, a single IPsec tunnel can be used in order to secure all the flows together, thus achieving a higher efficiency. This use of IPsec protects the packets only within the transport network between

tunnel ingress and egress and therefore does not provide end-to-end authentication or encryption. In some cases , e.g., where the end points are trusted, this model may be appropriate.

When a number of already secured flows including ESP [ESP] headers are optimized by means of TCM, and the addition of further security is not necessary, their ESP/IP headers can still be compressed using suitable algorithms [RFC5225], in order to improve the efficiency. This header compression does not change the end-to-end security model.

Future versions of this document will consider whether some TCM-TF mechanisms could be potentially exploited in order to deploy or amplify DoS attacks against network infrastructure. Solutions will be provided if potential attacks are identified.

7. Normative References

[ECRTP] Koren, T., Casner, S., Geevarghese, J., Thompson, B., and P. Ruddy, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering", RFC 3545, 2003.

[ESP] Kent, S., "IP Encapsulating Security Payload ", RFC 4303, 2005.

[FLV] ISO/IEC, "FLV and F4V File Format Specification", 14496-12 MPEG-4 Part 12, 2008.

[GRE] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, 2000.

[I.363.2] ITU-T, "B-ISDN ATM Adaptation layer specification: Type 2 AAL", I. 363.2, 1997.

[IPCP-HC] Engan, M., Casner, S., Bormann, C., and T. Koren, "IP Header Compression over PPP", RFC 3544, 2003.

[IPHC] Degermark, M., Nordgren, B., and S. Pink, "IP Header Compression", RFC 2580, 1999.

[IPsec] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[L2TPv3] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, 2005.

[MPLS] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[PPP-MUX] Pazhyannur, R., Ali, I., and C. Fox, "PPP Multiplexing", RFC 3153, 2001.

[PPP] Simpson, W., "The Point-to-Point Protocol (PPP)", RFC 1661, 1994.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5225] Pelletier, G. and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite ", RFC 5225, April 2008.

[ROHC] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObust Header Compression (ROHC) Framework", RFC 5795, 2010.

[RTP] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, 2003.

[SCTP] Stewart, Ed., R., "Stream Control Transmission Protocol", RFC 4960, 2007.

[SIP] Rosenberg, J., Schulzrinne, H., Camarillo, G., and et. al., "SIP: Session Initiation Protocol", RFC 3261, 2005.

[TCRTP] Thomson, B., Koren, T., and D. Wing, "Tunneling Multiplexed Compressed RTP (TCRTP)", RFC 4170, 2005.

[cRTP] Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, 1999.

Authors' Addresses

Jose Saldana
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza 50018
Spain

Phone: +34 976 762 698
Email: jsaldana@unizar.es

Dan Wing
Cisco Systems
771 Alder Drive
San Jose, CA 95035
US

Phone: +44 7889 488 335
Email: dwing@cisco.com

Julian Fernandez Navajas
University of Zaragoza
Dpt. IEC Ada Byron Building
Zaragoza 50018
Spain

Phone: +34 976 761 963
Email: navajas@unizar.es

Muthu Arul Mozhi Perumal
Cisco Systems
Cessna Business Park
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Phone: +91 9449288768
Email: mperumal@cisco.com

Fernando Pascual Blanco
Telefonica I+D
Ramon de la Cruz 84
Madrid 28006
Spain

Phone: +34 913128779
Email: fpb@tid.es