            Tunneling Compressing and Multiplexing (TCM) Traffic Flows. Reference
                                      Model
                          draft-saldana-tsvwg-tcmtf-09

Abstract

   Tunneling, Compressing and Multiplexing (TCM) is a method for
   improving the bandwidth utilization of network segments that carry
   multiple small-packet flows in parallel sharing a common path. The
   method combines different protocols for header compression,
   multiplexing, and tunneling over a network path for the purpose of
   reducing the bandwidth. The amount of packets per second can also be
   reduced.

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

Status of This Memo

Table of Contents

1. Introduction

      This document describes a way to combine different 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 surveillance, 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 low 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 or
      IoT scenarios 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, their traffic can be optimized by multiplexing
      packets belonging to different flows. As a consequence, bandwidth
      can be saved and the amount of packets per second can be reduced. If
      a number of small packets are waiting in the buffer, they can be
      multiplexed and transmitted together. In addition, if a transmission
      queue has not already been formed but multiplexing is desired, it is
      necessary to add a delay in order to gather a number of packets.
      This delay has to be maintained under some threshold if the service
      presents tight delay requirements. It is a believed fact that this
      delay and jitter can be of the same order of magnitude or less than
      other common sources of delay and jitter currently present on the
      Internet without causing harm to flows that employ congestion control
      based on delay.

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 [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 G.711 at 20 ms packetization: 25% header overhead G.729 at 20 ms packetization: 200% header overhead	IPv6+UDP+RTP: 60 bytes header G.711 at 20 ms packetization: 37.5% 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 [First-person], for which latency
is very critical. The quickness and the movements of the players are
important, and can decide the result of the game. In addition to
latency, these applications may be sensitive to jitter and, to a
lesser extent, to packet loss [Gamers], since they implement
mechanisms for packet loss concealment.

1.2.3. Other applications generating small packets

Other applications without delay constraints are also becoming
popular some examples are instant messaging, M2M packets sending
collected data in sensor networks using wireless or satellite
scenarios, IoT traffic generated in Constrained RESTful Environments,
where UDP packets are employed [RFC7252]. 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 the moments or places where network capacity gets scarce,
allocating more bandwidth is a possible solution, but it implies a
recurring cost. However, including optimization techniques between a
pair of network nodes (able to reduce bandwidth and packets per
second) when/where required is a one-time investment.

In scenarios including a bottleneck with a single Layer-3 hop, header
compression standard algorithms [cRTP], [ECRTP], [IPHC], [ROHC] can
be used for reducing the overhead of each flow, at the cost of
additional processing.

However, if header compression is to be deployed in a network path
including several Layer-3 hops, tunneling can be used at the same
time in order to allow the header-compressed packets to travel end-
to-end, thus avoiding the need to compress and decompress at each
intermediate node. In these cases, compressed packets belonging to
different flows can be multiplexed together, in order to share the
tunnel overhead. In this case, a small multiplexing delay will be
necessary as a counterpart, in order to join a number of packets to
be sent together. This delay has to be maintained under a threshold
in order to grant the delay requirements.

A series of recommendations about delay limits have been summarized
in [I-D.suznjevic-tsvwg-mtd-tcmtf], in order to maintain this
additional delay and jitter in the same order of magnitude than other
sources of jitter currently present on the Internet.

A demultiplexer and a decompressor are necessary at the end of the
common path, so as to rebuild the packets as they were originally
sent, making traffic optimization a transparent process for the
origin and destination of the flow.

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
[VoIP_opt], [FPS_opt].

All in all, the combined use of header compression and multipexing
provides a trade-off: bandwidth can be exchanged by processing
capacity (mainly required for header compression and decompression)
and a small additional delay (required for gathering a number of
packets to be multiplexed together).

1.2.5. Energy consumption considerations

   As an additional benefit, the reduction of the sent information, and
   especially the reduction of the amount of packets per second to be
   managed by the intermediate routers, can be translated into a
   reduction of the overall energy consumption of network equipment.
   According to [Efficiency] internal packet processing engines and
   switching fabric require 60% and 18% of the power consumption of
   high-end routers respectively. Thus, reducing the number of packets
   to be managed and switched will reduce the overall energy
   consumption. The measurements deployed in [Power] on commercial
   routers corroborate this: a study using different packet sizes was
   presented, and the tests with big packets showed a reduction of the
   energy consumption, since a certain amount of energy is associated to
   header processing tasks, and not only to the sending of the packet
   itself.

   All in all, another tradeoff appears: on the one hand, energy
   consumption is increased in the two extremes due to header
   compression processing; on the other hand, energy consumption is
   reduced in the intermediate nodes because of the reduction of the
   number of packets transmitted. Thi tradeoff should be explored more
   deeply.

1.3. Terminology

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

   o native packet

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

   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.

   o TCM flow

A flow of TCM packets, each one including a number of multiplexed
header-compressed packets.

o   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.

o   TCM session

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

o   policy manager

A network entity which makes the decisions about TCM optimization
parameters (e.g., 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 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. They can be
   classified according to the domains involved in the optimization:

1.4.1. Multidomain scenario

   In this scenario, the TCM tunnel goes all the way from one network
   edge (the place where users are attached to the ISP) to another, and
   therefore it can cross several domains. As shown in Figure 1, the
   optimization is performed before the packets leave the domain of an
   ISP; the traffic crosses the Internet tunnelized, and the packets are
   rebuilt in the second domain.

```
           _   _  _  _                                      _   _
         ( `        ) _                  _   _           ( `      )_ _
         (  +------+   )`)            ( `        )_          (  +------+  `)
  -->(_  -|TCM-IO|---  _) --->  (          )       `)    ----->(_-|TCM-EO|--_)-->
         (  +------+   _)            (_      (_  .   _) _)       (     +------+  _)
         (_  _      _  _)                                      (_   _  (   _)   _)

              ISP 1                    Internet                  ISP 2

              <--------------------TCM-------------------->
```
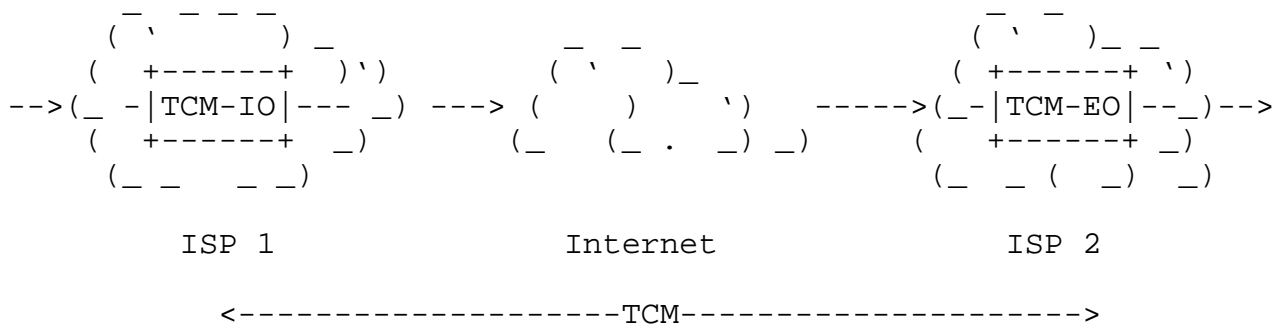
                              Figure 1

   Note that this is not from border to border (where ISPs connect to
   the Internet, which could be covered with specialized links) but from
   an ISP to another (e.g., managing all traffic from individual users
   arriving at a Game Provider, regardless users' location).

   Some examples of this could be:

   o An ISP may place a TCM optimizer in its aggregation network, in
      order to tunnel all the packets belonging to a certain service,
      sending them to the application provider, who will rebuild the
      packets before forwarding them to the application server. This
      will result in savings for both actors.

   o A service provider (e.g., an online gaming company) can be allowed
      to place a TCM optimizer in the aggregation network of an ISP,
      being able to optimize all the flows of a service (e.g., VoIP, an
      online game). Another TCM optimizer will rebuild these packets
      once they arrive to the network of the provider.

1.4.2. Single domain

   In this case, TCM is only activated inside an ISP, from the edge to
   border, inside the network operator. The geographical scope and
   network depth of TCM activation could be on demand, according to
   traffic conditions.

   If we consider the residential users of real-time interactive
   applications (e.g., VoIP, online games generating small packets) in a
   town or a district, a TCM optimizing module can be included in some
   network devices, in order to group packets with the same destination.
   As shown in Figure 2, depending on the number of users of the
   application, the packets can 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 may also be applied

for fiber residential accesses, and in mobile networks. This would
reduce bandwidth requirements in the provider aggregation network.


```
           +------+
 N users -|TCM-IO|\
           +------+ \
                    \
                     \          _  _                         _  _
           +------+   \-->  ( `    )_        +------+     ( `    )_
 M users -|TCM-IO|------> (      )    `)  --|TCM-EO|--> (      )    `)
           +------+   / ->(_ _ (_ .  _) _) +------+    (_ _ (_ .  _) _)
                     /
           +------+ /          ISP                        Internet
 P users -|TCM-IO|/
           +------+

           <--------------TCM--------------->
```
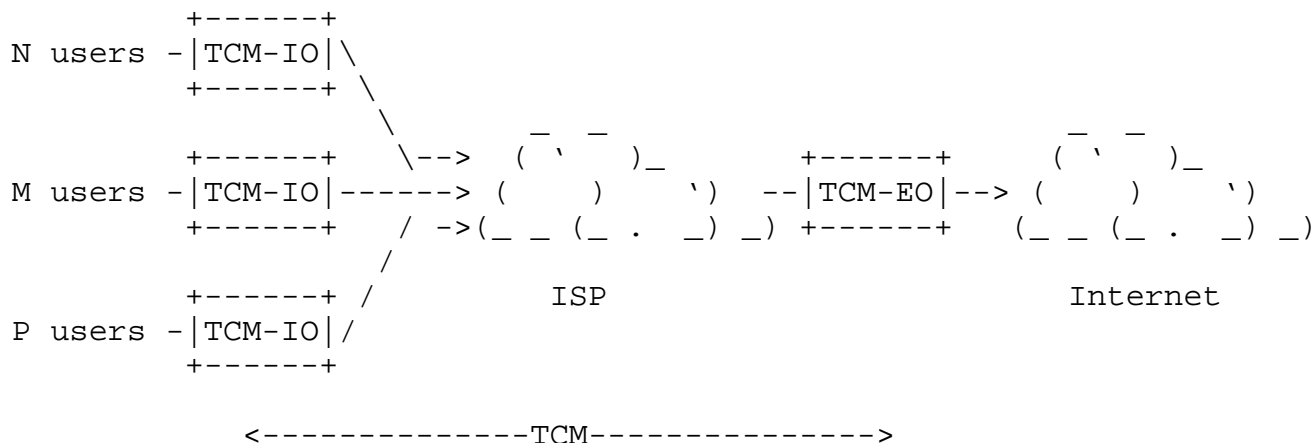
                             Figure 2

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

Finally, some networks use cRTP [cRTP] in order to obtain bandwidth
savings on the access link, but as a counterpart considerable CPU
resources are required 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 ISP network, thus avoiding the impact
on the CPU of the aggregation router.

1.4.3. Private solutions

End users can also optimize traffic end-to-end from network borders.
TCM is used to connect private networks geographically apart (e.g.,
corporation headquarters and subsidiaries), without the ISP being
aware (or having to manage) those flows, as shown in Figure 3, where
two different locations are connected through a tunnel traversing the
Internet or another network.

```
    _   _                              _   _                              _   _
  ( `   )_        +------+        ( `   )_        +------+        ( `   )_
 (       )   `)  --|TCM-IO|-->(       )   `)  --|TCM-EO|-->(       )   `)
(_   (_ .  _) _) +------+  (_   (_ .  _) _) +------+  (_   (_ .  _)_)

    Location 1                  ISP/Internet               Location 2

              <-------------TCM----------->
```
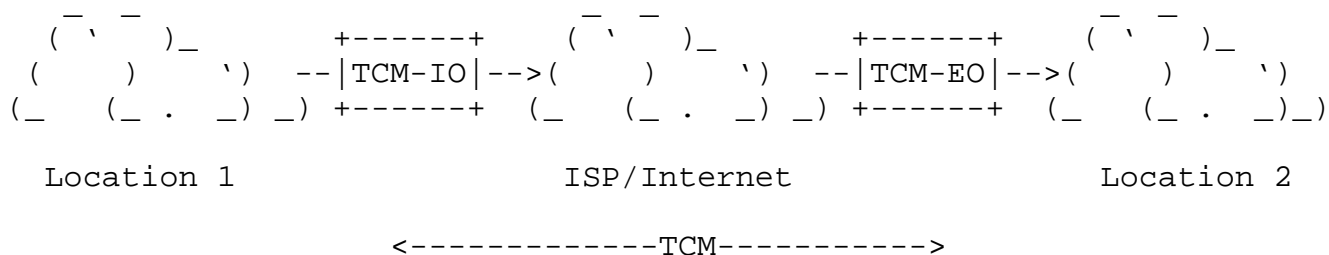
                              Figure 3

   Some examples of these scenarios:

   o The case of an enterprise with a number of distributed central
      offices, in which an appliance can 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 can be optimized, or a number of VoIP calls between
      two offices will also require less bandwidth and fewer packets per
      second. In many cases, a tunnel is already included for security
      reasons, so the additional overhead of TCM is lower.

   o An Internet cafe, which is suitable of having many users of the
      same application (e.g., VoIP, online games) 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
      businesses, so TCM savings may become interesting for their
      viability.

   o Community Networks [topology_CNs] (typically deployed in rural
      areas or in developing countries), in which a number of people in
      the same geographical place share their connections in a
      cooperative way. The structure of these networks is not designed
      from the beginning, but they grow organically as new users join.
      As a result, a number and a number of wireless hops are usually
      required in order to reach a router connected to the Internet.

   o 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 will improve the efficiency.

o 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 the Internet. The size of a 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 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 UDP and IPv6 protocols is certainly desirable. In addition to this, tunneling and multiplexing functions available from TCM allows extending compression functionality throughout the rest the network, to eventually reach the processing and control centers.

o 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 can be used in other cases: chatting, file sharing and applications related to WebRTC peers. There can 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.

1.4.4. Mixed scenarios

   Different combinations of the previous scenarios can be considered. Agreements between different companies can be established in order to save bandwidth and to reduce packets per second. As an example, Figure 4 shows a game provider that wants to TCM-optimize its connections by establishing associations between different TCM-IO/EOs placed near the game server and several TCM-IO/EOs placed in the

networks of different ISPs (agreements between the game provider and
each ISP will be necessary). In every ISP, the TCM-IO/EO would be
placed in the most adequate point (actually several TCM-IO/EOs could
exist per ISP) in order to aggregate enough number of users.

```
              _ _
N users  ( `   )_
+---+   (       )   `)
|TCM|->(_ (_ .  _)
+---+       ISP 1   \
          _ _        \      _ _          _              _ _
M users  ( `   )_      \   ( `   )      ( ` )          ( `   )
+---+   (       )   `) \ (      )   `)  (   )  `)  +---+  (      )   `)
|TCM|->(_     (_ ._)---- (_ (_ .  _) ->(_ (_ . _)->|TCM|->(_     (_ . _)
+---+       ISP 2      /   Internet      ISP 4     +---+  Game Provider
          _ _        /                     ^
O users  ( `   )_   /                      |
+---+   (       ) `) /                    +---+
|TCM|->(_     (_ ._)           P users->|TCM|
+---+       ISP 3                        +---+
```
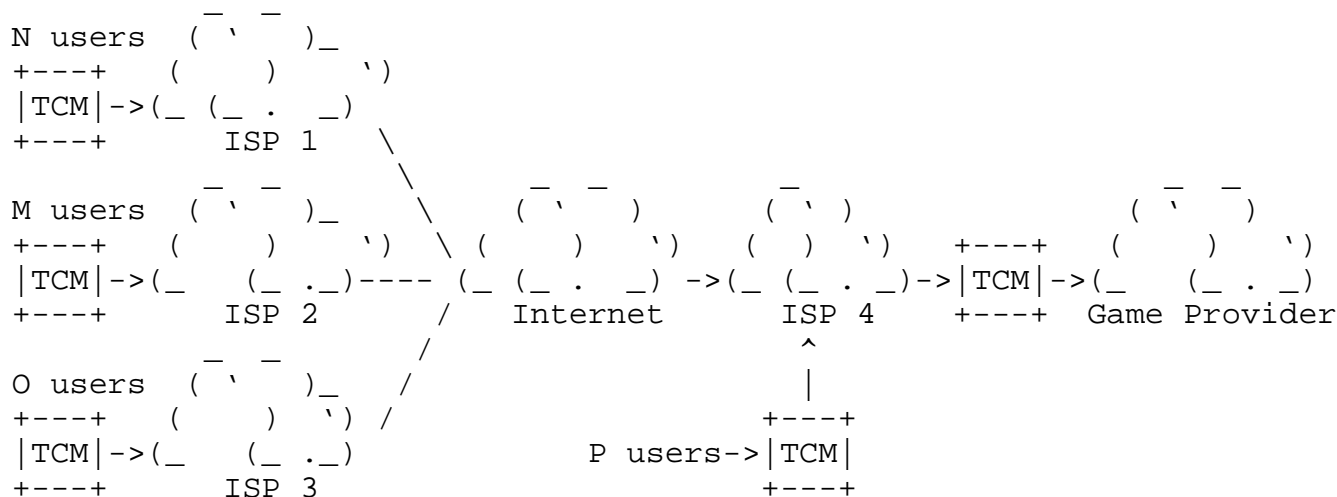
                              Figure 4

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:

   o network operators can compress traffic flows sharing a common
      network segment;

   o ISPs;

   o developers of VoIP systems can include this option in their
      solutions;

   o service providers, who can achieve bandwidth savings in their
      supporting infrastructures;

   o users of Community Networks, who may be able to save significant
      bandwidth amounts, and to reduce the number of packets per second
      in their networks.

   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 [Online].

1.6. Current Standard for VoIP

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

   o Regarding compression, [ECRTP] is the selected option.

   o Multiplexing is accomplished using PPP Multiplexing [PPP-MUX]

   o Tunneling is accomplished by using L2TP (Layer 2 Tunneling
      Protocol [L2TPv3]).

   The three layers are combined as shown in the Figure 5:

```
            RTP/UDP/IP
                |
                |           ---------------------------
                |
              ECRTP                compressing layer
                |
                |           ---------------------------
                |
              PPPMUX               multiplexing layer
                |
                |           ---------------------------
                |
              L2TP                  tunneling layer
                |
                |           ---------------------------
                |
               IP
```

                            Figure 5

1.7. Current Proposal

   In contrast to the current standard [TCRTP], TCM allows other header
   compression protocols in addition to RTP/UDP, since services based on
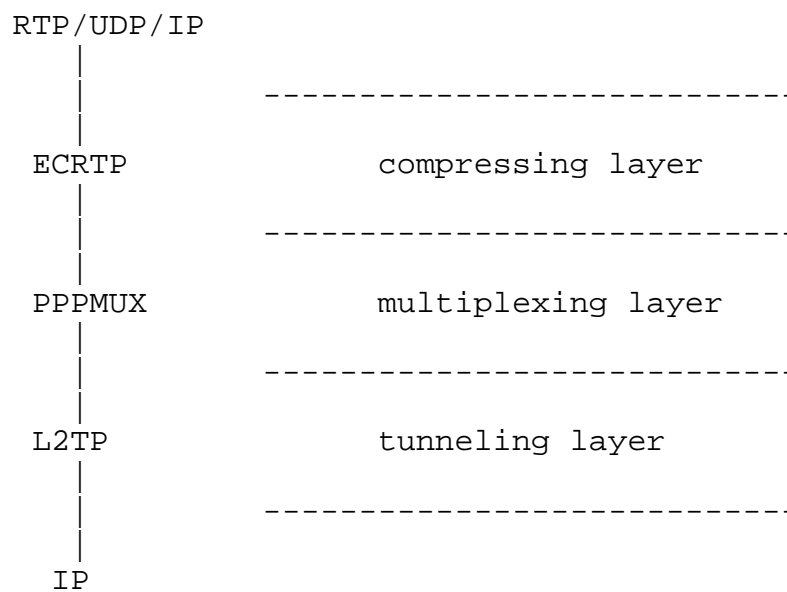   small packets also use by bare UDP, as shown in Figure 6:

```
      UDP/IP          RTP/UDP/IP
          \              /
           \            /              -----------------------------
            \          /
Nothing or ROHC or ECRTP or IPHC          header compressing layer
                 |
                 |                      -----------------------------
                 |
    PPPMux or other mux protocols            multiplexing layer
                 |
               / \                      -----------------------------
              /   \
             /     \
    GRE or L2TP     \                         tunneling layer
         |          MPLS
         |                             -----------------------------
         IP
```

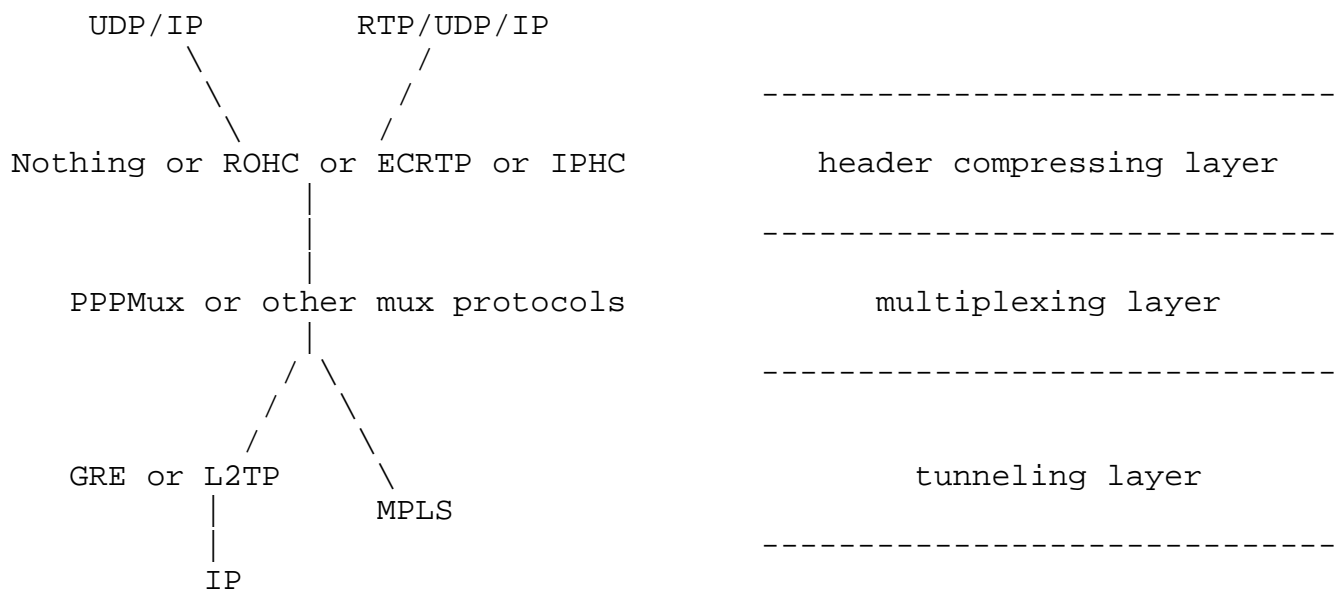   Figure 6

    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:

    o 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 can be
       selected depending on the protocols used by 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 not shown in Figure 6, ESP (Encapsulating
       Security Payload [ESP]) headers can also be compressed.

    o Multiplexing can be accomplished using PPP Multiplexing (PPPMux)
       [PPP-MUX]. However, PPPMux introduces an additional compelxity,
       since it requires the use of PPP, and a protocol for tunneling
       layer 2 frames. For this reason, other multiplexing protocols can
       also be considered, as the one proposed in
       [I-D.saldana-tsvwg-simplemux].

    o 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, combining [ECRTP], [PPP-MUX] and [L2TPv3], so backwards
compatibility with TCRTP is provided. If a TCM optimizer implements
ECRTP, PPPMux and L2TPv3, compatibility with RFC4170 MUST be granted.

If a single link is being optimized a tunnel is unnecessary. In that
case, both optimizers MAY perform header compression between them.
Multiplexing may still be useful, since it reduces packets per
second, which is interesting in some environments (e.g., satellite).
Another reason for that is the desire of reducing energy consumption.
Although no tunnel is employed, this can still be considered as TCM
optimization, so TCM signaling protocols will be employed here in
order to negotiate the compression and multiplexing parameters to be
employed.

Payload compression schemes may 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 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):

```
    [ending device]---[ending device]
                   ^
                   |
              TCM over IP
```
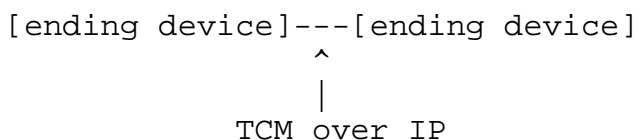
                          Figure 7

Another way TCM can be implemented is with an external optimizer.
This device can be placed at strategic places in the network and can
dynamically create and destroy TCM sessions without the participation
of the endpoints that generate the flows (Figure 8).

```
[ending device]\                                       /[ending device]
                \                                     /
[ending device]----[optimizer]-----[optimizer]-----[ending device]
                /                                     \
[ending device]/                                       \[ending device]
              ^                      ^                      ^
              |                      |                      |
          Native IP           TCM over IP           Native IP
```
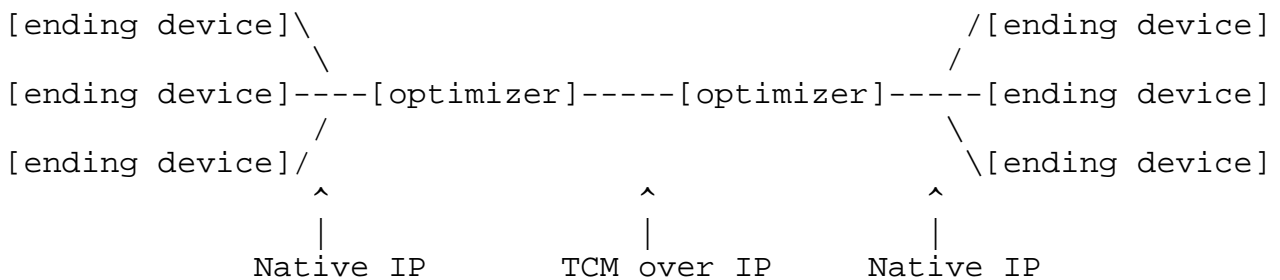
                               Figure 8

A number of already compressed flows can also be merged in a tunnel
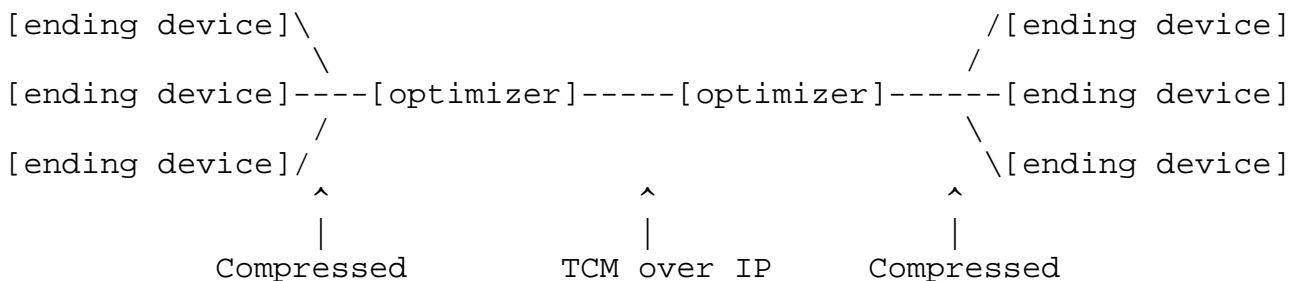using an optimizer in order to increase the number of flows in a
tunnel (Figure 9):

```
[ending device]\                                       /[ending device]
                \                                     /
[ending device]----[optimizer]-----[optimizer]------[ending device]
                /                                     \
[ending device]/                                       \[ending device]
              ^                      ^                      ^
              |                      |                      |
          Compressed          TCM over IP           Compressed
```

                               Figure 9

2.2. Choice of the compressing protocol

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

o IPHC (IP Header Compression [IPHC]) permits the compression of
   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.

o 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.

o 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).

o ROHC (RObust Header Compression [ROHC]) is able to compress 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.

The present document does not determine which of the existing
protocols has to be used for the compressing layer. The decision
will depend on the scenarioand the service being optimized. It will
also be determined by the packet loss probability, RTT, jitter, 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. For example, Simplemux [I-D.saldana-tsvwg-simplemux]
   can be employed as a light multiplexing protocol which is able to
   carry packets belonging to different protocols.

   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 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.

   When multiplexing, a limit in the packet size has to be established
   in order to avoid problems related to MTU. This document does not
   establish any rule about this, but it is strongly recommended that
   some method as Packetization Layer Path MTU Discovery is used before
   multiplexing packets[RFC4821].

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.

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 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:

```
            +------------+----------------------+
            |            |                      |
            |   Compr    |                      |
            |   Header   |        Data          |
            |            |                      |
            |            |                      |
            +------------+----------------------+
```
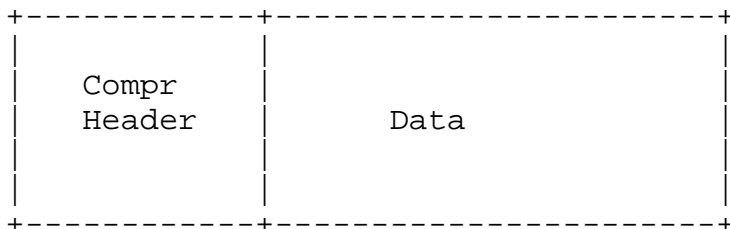
                             Figure 10

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

```
+-------+---+------+-------+-----+   +---+------+-------+-----+
| Mux   |P L|      |       |     |   |P L|      |       |     |
| PPP   |F X|Len1  | PPP   |     |   |F X|LenN  | PPP   |     |
| Prot. |F T|      | Prot. |Info1| ~ |F T|      | Prot. |InfoN|
| Field |   |      | Field1|     |   |   |      |FieldN |     |
| (1)   |1-2 octets| (0-2) |     |   |1-2 octets| (0-2) |     |
+-------+---------+-------+-----+   +---------+-------+-----+
```
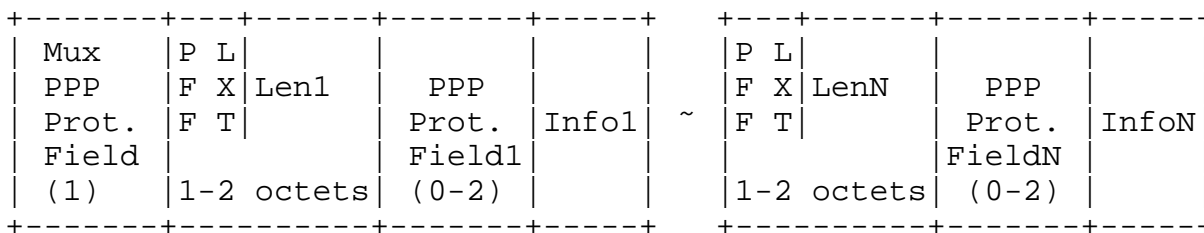
                            Figure 11

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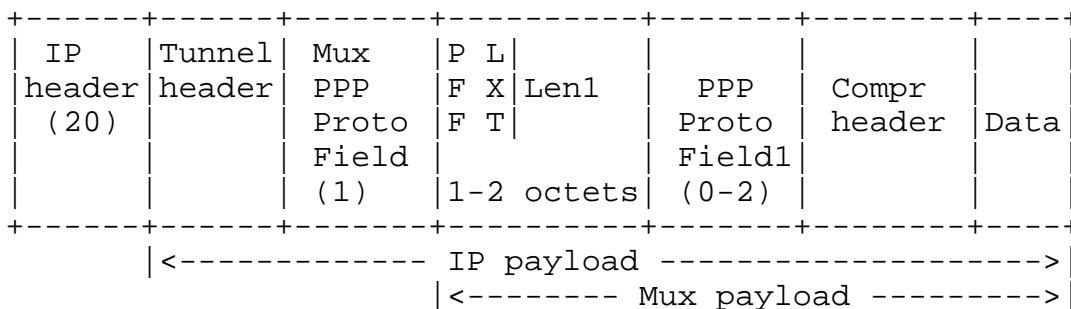   using L2TP or GRE tunneling:

```
+------+------+-------+---------+-------+--------+----+
| IP   |Tunnel| Mux   |P L|     |       |        |    |
|header|header| PPP   |F X|Len1 | PPP   | Compr  |    |
| (20) |      | Proto |F T|     | Proto | header |Data|
|      |      | Field |   |     | Field1|        |    |
|      |      | (1)   |1-2 octets| (0-2) |        |    |
+------+------+-------+---------+-------+--------+----+
       |<------------- IP payload -------------------->|
                      |<-------- Mux payload --------->|
```

                            Figure 12

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

```
+------+-------+---------+-------+--------+----+
|MPLS  | Mux   |P L|     |       |        |    |
|header| PPP   |F X|Len1 | PPP   | Compr  |    |
|      | Proto |F T|     | Proto | header |Data|
|      | Field |   |     | Field1|        |    |
|      | (1)   |1-2 octets| (0-2) |        |    |
-+------+-------+---------+-------+--------+----+
       |<---------- MPLS payload -------------->|
              |<-------- Mux payload --------->|
```
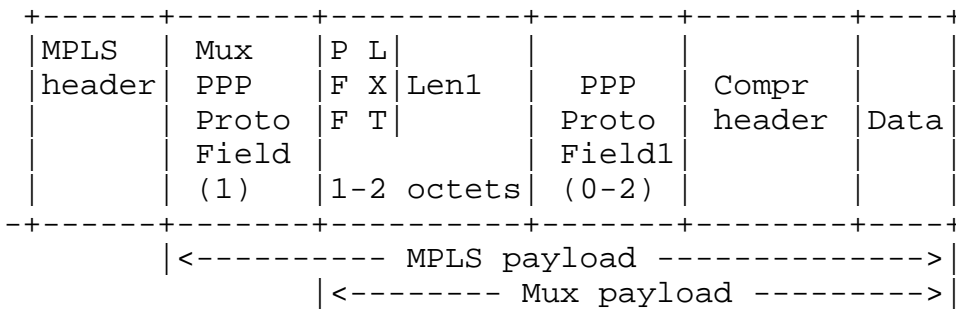
                            Figure 13

   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.
6310 Watercrest Way, Unit 203
Lakewood Ranch, FL 34202
USA

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.

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.

The resilience of TCM to denial of service, and the use of TCM to
deny service to other parts of the network infrastructure, is for
future study.

7. References

7.1. Normative References

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

   [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.

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

   [H.323]   International Telecommunication Union, "Recommendation
               H.323", Packet based multimedia communication systems
               H.323, July 2003.

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

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

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

   [RFC4821]   Mathis, M. and J. Heffner, "Packetization Layer Path MTU
                  Discovery", RFC 4821, March 2007.

   [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.

   [RFC7252]   Shelby, Z., Hartke, K., and C. Bormann, "The Constrained
                  Application Protocol (CoAP)", RFC 7252, June 2014.

   [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.

7.2. Informative References

   [Efficiency]
                  Bolla, R., Bruschi, R., Davoli, F., and F. Cucchietti,
                  "Energy Efficiency in the Future Internet: A Survey of
                  Existing Approaches and Trends in Energy-Aware Fixed
                  Network Infrastructures", IEEE Communications Surveys and
                  Tutorials vol.13, no.2, pp.223,244, 2011.

   [First-person]
                  Ratti, S., Hariri, B., and S. Shirmohammadi, "A Survey of
                  First-Person Shooter Gaming Traffic on the Internet", IEEE
                  Internet Computing vol 14, no. 5, pp. 60-69, 2010.

   [FPS_opt]   Saldana, J., Fernandez-Navajas, J., Ruiz-Mas, J., Aznar,
               J., Viruete, E., and L. Casadesus, "First Person Shooters:
               Can a Smarter Network Save Bandwidth without Annoying the
               Players?", IEEE Communications Magazine vol. 49, no.11,
               pp. 190-198, 2011.

   [Gamers]   Oliveira, M. and T. Henderson, "What online gamers really
               think of the Internet?", NetGames '03 Proceedings of the
               2nd workshop on Network and system support for games, ACM
               New York, NY, USA Pages 185-193, 2003.

   [I-D.saldana-tsvwg-simplemux]
               Saldana, J., "Simplemux. A generic multiplexing protocol",
               draft-saldana-tsvwg-simplemux-02 (work in progress),
               January 2015.

   [I-D.suznjevic-tsvwg-mtd-tcmtf]
               Suznjevic, M. and J. Saldana, "Delay Limits and
               Multiplexing Policies to be employed with Tunneling
               Compressed Multiplexed Traffic Flows", draft-suznjevic-
               tsvwg-mtd-tcmtf-03 (work in progress), June 2014.

   [Online]   Feng, WC., Chang, F., Feng, W., and J. Walpole, "A traffic
               characterization of popular on-line games", IEEE/ACM
               Transactions on Networking 13.3 Pages 488-500, 2005.

   [Power]   Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang,
               D., and S. Wright, "Power Awareness in Network Design and
               Routing", INFOCOM 2008. The 27th Conference on Computer
               Communications. IEEE pp.457,465, 2008.

   [topology_CNs]
               Vega, D., Cerda-Alabern, L., Navarro, L., and R. Meseguer,
               "Topology patterns of a community network: Guifi. net.",
               Proceedings Wireless and Mobile Computing, Networking and
               Communications (WiMob), 2012 IEEE 8th International
               Conference on (pp. 612-619) , 2012.

   [VoIP_opt]
               Saldana, J., Fernandez-Navajas, J., Ruiz-Mas, J., Murillo,
               J., Viruete, E., and J. Aznar, "Evaluating the Influence
               of Multiplexing Schemes and Buffer Implementation on
               Perceived VoIP Conversation Quality", Computer Networks
               (Elsevier) Volume 6, Issue 11, pp 2920 - 2939. Nov. 30,
               2012.

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
    Ericsson
    Ferns Icon
    Doddanekundi, Mahadevapura
    Bangalore, Karnataka 560037
    India

    Email: muthu.arul@gmail.com

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

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