Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 29, 2015

A. Jain
A. Terzis
Google
N. Sprecher
P. Szilagyi
H. Flinck
Nokia Networks

October 26, 2014

Mobile Throughput Guidance Signaling Protocol
draft-flinck-mobile-throughput-guidance-00.txt

Abstract

The behaviour of the Transmission Control Protocol (TCP), which assumes that network congestion is the primary cause for packet loss and high delay, can lead to inefficient use of a cellular network's resources and degrade application performance. The root cause for this inefficiency is that TCP has difficulty adapting to the rapidly varying network conditions. In cellular networks, the bandwidth available for end devices can vary by an order of magnitude over a few seconds due to changes in the underlying radio channel conditions, as devices move, as well as changes in system load as other devices enter and leave the network.

This document proposes a mechanism and protocol elements that can be used to assist TCP in cellular networks, ensuring high utilization and high service delivery performance.

The document describes the applicability of the proposed mechanism for video delivery over cellular networks; it also presents test results.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 April 29, 2015.

Table of Contents

1. Introduction

The following sub-sections present the problem statement and the solution principles.

1.1. Contributing Authors

The editors gratefully acknowledge the following additional contributors: Swaminathan Arunachalam and Csaba Vulkan.

1.2. Terminology

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].

1.3. Acronyms and Abbreviations

This document uses the following acronyms:

ECGI E-UTRAN Cell Global Identifier format
ECN Explicit Congestion Notification
HMAC Hash-based Message Authentication Code
IP Internet Protocol
IV Initialization Vector
LTE Long Term Evolution
MTG Mobile Throughput Guidance
RAN Radio Access Network
RTT Round Trip Time
SACK Selective Acknowledgement
TCP Transmission Control Protocol
UE User Equipment

1.4. Problem statement

The bandwidth available for end devices in a cellular network can vary by an order of magnitude over a few seconds. This variation is due to changes in the underlying radio channel conditions, as devices move, as well as system load variations driven by the arrival and departure of devices to/from the network. On the other hand, packet losses tend to be sporadic and temporary because retransmissions mechanisms at the physical and link layers repair most packet corruptions.

Transport protocols that derive network capacity through implicit signals, such as packet loss and delay variation, can have difficulty adapting to such rapidly changing network conditions. In turn, this difficulty leads to poorer quality of experience for applications like video playback as well as inefficient use of the cellular network's resources.

1.5. Mechanism Principles

This document proposes that the cellular network provides information on throughput guidance to the TCP server; this information will indicate the throughput estimated to be available at the radio downlink interface. The network SHOULD provide this information in near real time in situations where the network conditions are changing frequently or the user is moving.

While the implementation details will vary according to the access technology, the resource allocation can be abstracted as the capacity of the "radio link" between the network and the UE. For example, in the case of an LTE network, the number of physical resource blocks allocated to a UE, along with the modulation scheme and coding rate used, can be translated into radio link capacity in Megabits per second (Mbps).

The TCP server can use this explicit information to inform several congestion control decisions. For example: (1) selecting the initial window size, (2) deciding the value of the congestion window during the congestion avoidance phase, and (3) adjusting the size of the congestion window when the conditions on the "radio link" deteriorate. In other words, with this additional information, TCP does neither have to congest the network when probing for available resource, nor rely on heuristics to reduce its sending rate after a congestion episode.

The same explicit information can also be used to optimize application behaviour given the available resources. For example, when video is encoded in multiple bitrates, the application server can select the highest encoding rate that the network can deliver.

This document proposes an in-band exposure mechanism where the information elements are added to the TCP headers of the relevant upstream packets. In particular, the throughput guidance information is added into the Options field of the TCP header (see RFC 0793 [RFC0793]) of packets from the TCP client to the TCP server. An in-band mechanism is proposed because it does not require a separate interface, reference value, or correlation mechanism. Furthermore, an in-band mechanism can keep up with the rapid changes in the underlying radio link throughput.

Section 4 describes the definition details and semantics of the Options field of the TCP header.

The proposed scheme is similar to existing mechanisms such as ECN, where an ECN-aware router sets a mark in the IP header in order to signal impending congestion (see [RFC3168]). Note, however, that the

proposed scheme provides explicit information, (termed "Throughput Guidance") about the estimated throughput available at the radio link between the Radio Access Network (RAN) and the UE.

The following issues are not covered: (1)the throughput estimation for the uplink between the UE and the RAN, and (2)the capacity of the network path between the RAN and the server communicating with the UE.

2. Architecture

A Mobile Throughput Guidance Signaling Protocol (MTGSP) is specified to allow a functional entity that resides in the RAN to signal throughput guidance information to the TCP server. The TCP server resides behind the core network of the operator or in the Internet.

As Figure 1 depicts below, the functional element of the Throughput Guidance Provider signals to the TCP server the information on the (near-real time) throughput estimated to be available at the radio downlink interface.

The TCP server MAY use the information to optimize the TCP behaviour. The information MAY also be used to adapt the application behaviour accordingly and to optimize service delivery performance.

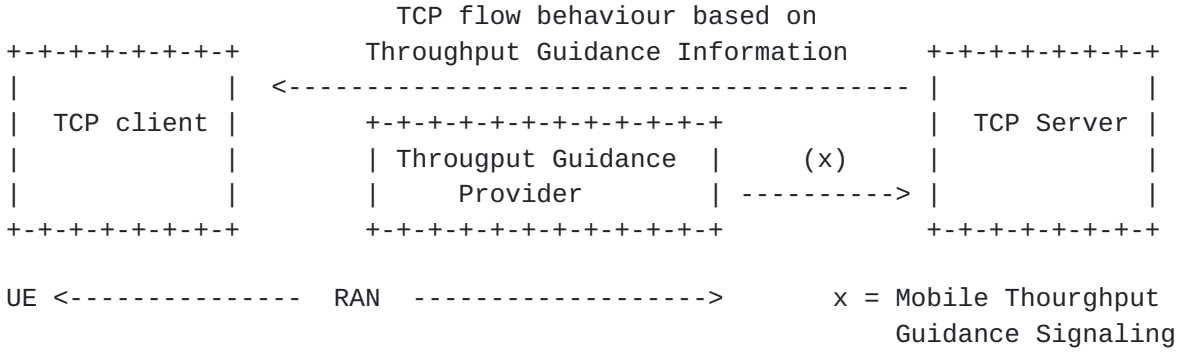


Figure 1

As described above, MTGSP SHALL use the Options field of the TCP header of the same TCP flow to provide throughput guidance information.

The information source and the algorithm used by the Throughput Guidance Provider to calculate the throughput guidance are beyond the scope of this document.

The TCP server MAY use the throughput guidance information to assist TCP in any of the following ways:

- Determine the size of the initial congestion window
- Determine when to exit the slow start phase
- Determine the size of the congestion window during the congestion avoidance phase
- Determine the size of the window after a congestion event

3. Applicability to Video Delivery Optimization

The applicability of the protocol specified in this document to mobile video delivery optimization has been evaluated and tested in different network load scenarios.

In this use case, TCP traffic, for which throughput guidance information is required, passes through a Radio Analytics application which resides in a Mobile-edge Computing (MEC) server (see [MEC_White_Paper]). This Radio Analytics application acts as the Throughput Guidance Provider and sends throughput guidance information for a TCP flow using the Options field in the TCP header (according to the message specification provided below in section 4). The TCP server MAY use this information to assist TCP congestion control decisions as described above. The information MAY also be used to select the application level coding so that it matches the estimated capacity at the radio downlink.

All of these improvements aim to enhance the quality of experience of the end user by reducing the time-to-start of the content as well as video stall occurrences.

With Mobile-edge Computing, the Radio Analytics application can be deployed on top of platforms implemented by different vendors and across multi-operator networks. This means that efficient utilization of the network resources can be expected as well as enhanced quality of experience for the vast majority of the end users.

3.1. Test Results

Nokia Networks and Google tested the video delivery optimization use case in a laboratory environment, simulating (as closely as possible) a live production network. Different network load scenarios were simulated.

All network level metrics showed an average improvement of 40-80%, as detailed below:

- Reduction of end-to-end RTT by 40-60%
- TCP retransmissions reduced by 70-80%
- Buffer bloat reduced by 70-80%

The application-level metrics also improved, as detailed below:

- Click-to-play time reduced by 12-34%
- Stalling occurrences reduced by 46-100%
- Reduction in the number of format changes by 21-27%

4. Protocol

The Mobile Throughput Guidance Signaling message conveys information on the throughput estimated to be available at the down link path of a given TCP connection. The information is sent to the uplink end-point of the connection (e.g., the TCP server). The TCP server MAY use this information to optimize TCP behavior and to adjust application-level behavior to the link conditions.

A good example is a content optimizer or a cache that can adapt the application-level coding to match the indicated downlink radio conditions. As radio link conditions may change rapidly, this guidance information is best carried in-band using TCP options headers rather than through an out-of-band protocol.

Using the TCP options to carry throughput guidance associates the guidance information with an ongoing TCP connection and explicitly avoids separate session identification information. The proposed mechanism neither impacts the TCP state machine nor the congestion control algorithms of the TCP protocol.

The Options field enables information elements to be inserted into each packet with a 40-byte overall limit; this needs to be shared with the standardized and widely-used option elements, such as the

TimeStamp and SACK. The TCP Options field uses a Kind-Length-Value structure that enables TCP implementations to interpret or ignore information elements in the Options field based on the Kind.

In this draft, we define a Kind-Length-Value structure for encoding information about the estimated capacity of a radio access link between the RAN and the UE which is traversed by a TCP connection. Note that the Mobile Throughput Guidance Signaling defines an extendible framework that can convey confidential information to an information receiver. The intention is to define a generic container to convey in-band information within the limited TCP Option space with optional authentication and/or encryption capabilities. Throughput guidance is provided in this document. Additional information can be specified in future documents.

The TCP options for Mobile Throughput Guidance Signaling are added by the Throughput Guidance Provider functional element (which resides on top of a radio network element), when there is enough space in the TCP header.

Confidential information must be delivered in a secure way. The information can be provided as plain text in a secure and closed network. In other cases, the information should be authenticated and encrypted at the TCP-header level (between the Throughput Guidance Provider and the TCP server). An acceptable level of authentication and encryption (according to best common practices) may require more data than can be fitted into a single TCP header (maximum of 40 if no other options are present). As described below, packet fragmentation will be used is such a case.

Two transfer modes are defined to deal with data confidentiality in this document; namely, plain-text mode and authenticated encryption mode. A third mode, authentication-only mode, is equally feasible. However, we do not currently provide the authentication-only mode but will consider it for later updates. Both modes share a common Kind-Length-Value "option header" structure with a flag field separating the two cases.

4.1. Common Kind-Length-Value header

To make Mobile Throughput Guidance Signaling extendible to different use cases a common Kind-Length-Value header is defined below.

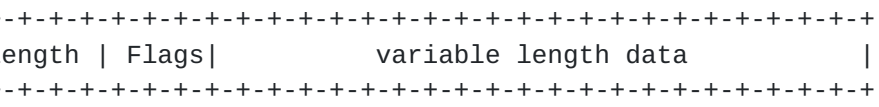


Figure 2

Kind:

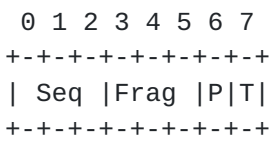
The Option Kind field indicates that the subsequent options are part of Mobile Throughput Guidance Signaling. The size of this field is 1 byte.

Length:

A 1 byte field, indicating the length of the kind and the length fields, as well as the length of the following Options field(s):

Flags:

One byte of MTG protocol flag field as defined below.



Flag field of common Kind-Lenght-Value header

Figure 3

Seq:

Three-bit sequence number that maintains context across different packet types as defined by P- and T-bits below. The scope of the sequence number is to protect against packet reordering, not to provide a globally unique identifier or sequence number. The use of these bits are reserved for possible transfer mode extensions.

Frag:

Three bits that provide information about how to reassemble information if fragmented into multiple packets. If no fragmentation across multiple TCP packet headers is needed, these bits are set to zero. Otherwise, Frag is a counter starting from 1 and incremented by 1 for each subsequent packet of the same type

(see P- and T-bits below). For the last fragment, the Fragment is always 7 (binary 111) to indicate that the information is complete.

P and T bits:

These two bits encode the packet type: Plaintext (P=0, T= 0), Cipher text (P=0, T=1), Nonce (IV) (P=1, T=0) or Authentication (P=1, T=1). For Plaintext, the Fragment bits are always zero.

Variable length data:

The variable length content in type, value format. The content depends of the transfer mode as defined in the following sections of this document.

As an example for the use of Flag-field, consider a cipher text of a single block. For it the T-bit is set one, P-bit is set to zero, Fragment and Seq-fields are zero in the Flag-field. In case the cipher text option that cannot fit into a single TCP packet option, the cipher text is fragmented across multiple TCP headers. The first fragment has value Frag= 001, and the value is incremented for each subsequent fragment. The last fragment is marked with all Frag-bits set to 1 (Frag= 111 for the last fragment). Details follow in the next sections.

4.2. Plane text mode Throughput Guidance Options

The plain text mode can be used in secure and closed networks or with information that has no confidentiality requirement. The plane text mode is made of one or more type, value -pairs. The type defines fixed the length of the following value.

Table of Type Value pairs of Throughput Guidance option data

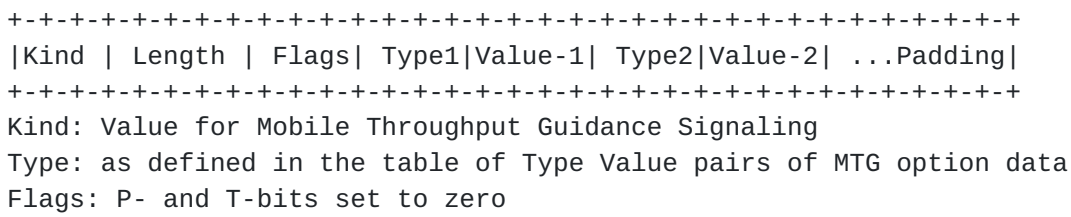
Name	Type	Length	Value Unit
Throughput Guidance	1	2 bytes	kbits/s
Access point ID	4	7 bytes	global identifier (ECGI)

Table 1: MTG type-vale pairs

The Type 1 element carries the actual throughput estimate in the 16-bit value unit. The throughput value is encoded using a fixed-point number representation. The 12 most significant bits are used for the integer value while the bottom 4 bits correspond to the

decimal portion of the throughput value. Throughput is expressed in Megabits per second.

The type-value pair elements are laid out consecutively in the header. At the end padding (i.e., the NO-OP TCP Option header with kind equal to 1, or the End of Option List TCP Option header with kind equal to 0) may be required to align the header size to the multiple of 4 bytes (required by the TCP standard). All bits in the Flag field are set to zero.

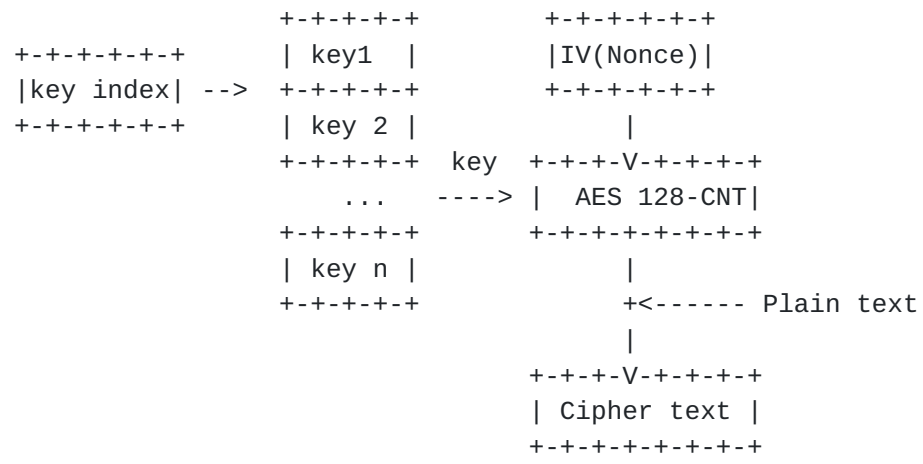


Layout of plain text option data in the TCP header options space.

Figure 4

4.3. Encrypted mode

Encryption requires authentication for integrity protection, as it is insecure to use encryption without it. Thus, the encrypted contains authentication as well. Encryption and authentication must use different keys. The following diagram shows the encryption process.



Encryption method

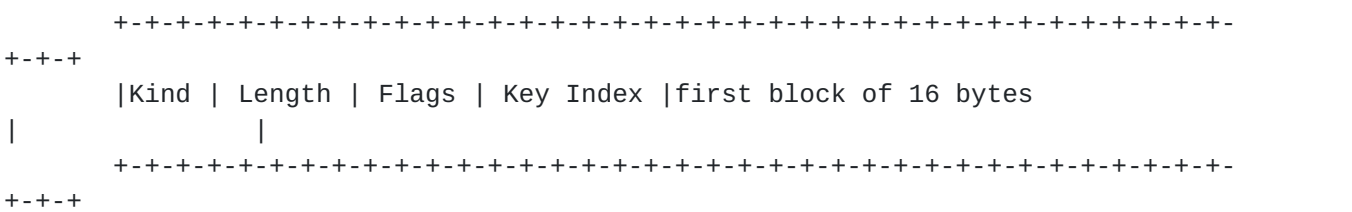
Figure 5

The encryption uses Advanced Encryption Standard (AES), 128 bits (16 bytes) block size, 128 bits (16 bytes) key size, Counter (CTR) block cipher mode. Integrity protection with CTR mode is MUST; this is be provided via HMAC based message authentication (see Authentication section below).

The plaintext contains type-value pair elements of the variable length data. The plaintext is divided into blocks of 16 bytes. A block of plain text MUST not exceed 16 bytes in a single run. Encryption takes a key (16 bytes), an IV or Nonce (16 bytes), the plain-text (at most 16 bytes) and produces a cipher text of 16 bytes. Note: multiple keys, at most 256, may be available (can be negotiated via out-of-band key management; this out of scope of this document) for encryption key index. The keys MUST be different from those used for authentication.

The Nonce is 16 bytes. A unique Nonce is generated for each encrypted block. The same Initialization Vector, IV or Nonce MUST NOT be used with the same encryption key more than once. This is be enforced by the Throughput Guidance Provider; otherwise security scheme will be broken.

The resulting cipher text is in blocks of 16 bytes. The cipher text blocks are packed into the option space together with the used Key Index in a following way if they fit into singe option space of a single TCP header.



Kind: Value for Mobile Throughput Guidance Signaling
Flags: Type of cipher text T-bit set to 1, only one block Frag= 000.
Key Index is the index used in encryption

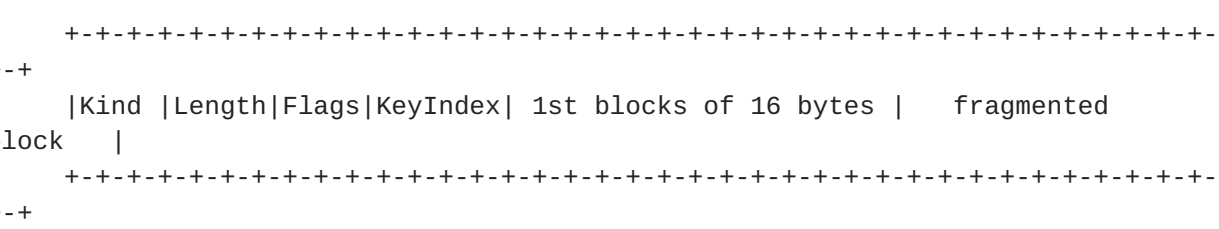
Cipher text layout in the TCP options without fragmentation

Figure 6

The flag field of the common option header indicates that the content is cipher text by having the T bit set to one. Since the ciphered block is not fragmented the Frag-bits of the flag field are set to zero (Frag= 000). (Use of Seq bits is left for later submissions).

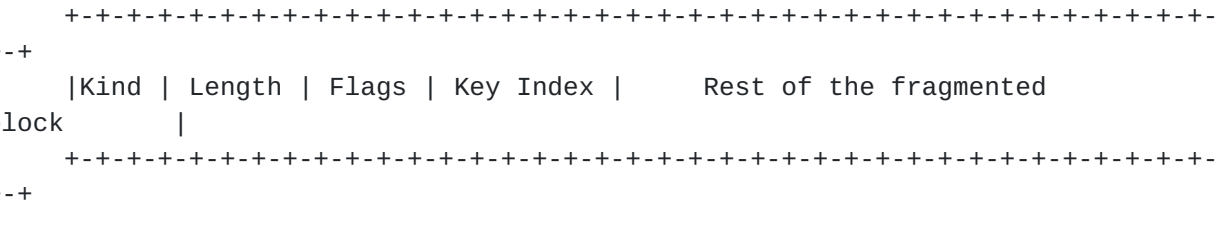
If there are multiple cipher text blocks of 16 bytes, the flag field shows the type of the option being cipher text with the T-bit set to one, and by Frag-field showing the fragment number starting from 001 and incremented by one for each subsequent fragment of a packet of the same type. For the last fragment, the Frag-field is always binary 111 to indicate the last fragment.

First fragment:



Kind: Value for Mobile Throughput Guidance Signaling
Flags: Type of cipher text T-bit = 1, Frag field = 001 first fragment
Key Index is the index used in encryption

Second fragment:



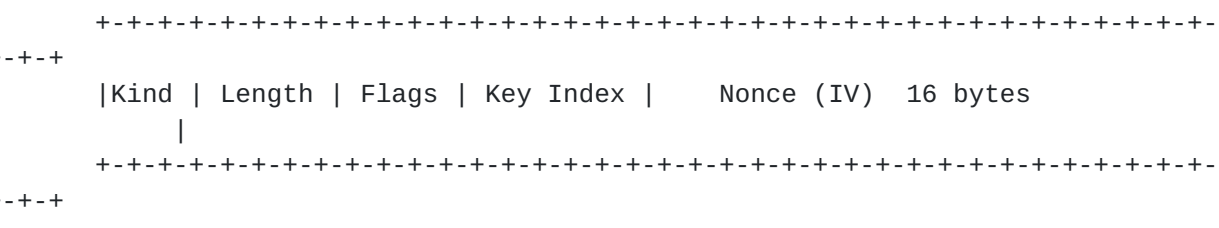
Kind: Value for Mobile Throughput Guidance Signaling
Flags: Type of cipher text T-bit = 1, Frag field = 111 last fragment
Key Index is the index used in encryption

Cipher text layout extending to two consecutive headers

Figure 7

4.4. Nonce (Initialization Vector)

The 16 byte Nonce (or IV) is transmitted along with the cipher text to protect against de-synchronization between the encryption-decryption points.



Kind: Value for Mobile Throughput Guidance Signaling
Flags: Type of IV/Nonce P-bit set to 1, only one block Frag= 000

Key Index is the index used in encryption

Nonce (IV) in a single header

Figure 8

If the Nonce (IV) doesn't fit into the remaining free bytes of the option field it needs to be fragmented using the Frag-field in the same way as cipher text layout is extending across two or more consecutive TCP headers but with the option type field set to indicate Nonce/IV by P-bit set to 1.

4.5. Authentication

The authentication covers the cipher text, the Nonce (IV) and includes additional TCP protocol header fields to protect against replay attacks. The authentication uses HMAC codes (e.g. HMAC-SHA2-224), 128 bits (16 bytes) key size, 224 bits (28 bytes) digest size. Multiple keys (at most 256) for authentication with the same information receiver can be used. The keys MUST be different from those used for encryption. Truncation is possible but at least 160 bits (20 bytes) must be used from the digest to meet the typical security level of mobile networks.

Authentication takes a key, the input (arbitrary length) and produces a 28 byte long digest, which is truncated to 20 bytes (keeping the most significant bytes). The HMAC algorithm and truncation can be negotiated via key management (out of scope of this document).

The authentication covers the TCP sequence number, ACK number, and TimeStamp (TSval, TSecr not the possible 2 bytes of padding) fields of the TCP header as well as the Common Kind-Length-Value header with its data in all cipher text option and IV/Nonce option packets. (The Authentication type options itself cannot be covered by the authentication.)

The order in which the fields are included into the message authentication code is the following. From the TCP header: TCP Seq, ACK, TSval, TSecr. Followed by the following fields from the ciphered text: Kind, Length, Flags, Key Index, cipher text, and from the IV/Nonce type of option packets TCP Seq, ACK, TSval, TSecr (note cipher text and IV/Nonce type of options may be in different TCP packets) followed by Kind, Length, Flags, Key Index, Nonce/IV.

In case the option packets used as input to the HMAC are fragmented into multiple TCP headers, they are processed so that headers with cipher text option are processed first, followed by IV/Nonce option packets.

The options containing the result of the HMAC are marked by setting both P- and T-bits of the flag-field to one. Key Index is set to point to the used authentication key, followed by the resulting authentication code. If the option doesn't fit into the free option

space in the TCP header, it is fragmented across multiple TCP headers in the same way as the cipher text options use the Frag-field.

5. Manageability considerations

There SHOULD be a mechanism to configure the application in the RAN with a list of destinations to which throughput guidance should be provided.

In addition, it SHOULD be possible to configure the frequency (in milliseconds) at which throughput guidance needs to be signaled as well as the required security level and parameters for the encryption and the authentication if supported.

6. Security considerations

The introduction of explicit information from the cellular network that can affect the behavior of a transport connection endpoint introduces a set of security considerations.

First, the identity of the network entity that injects the throughput guidance header must be explicitly known to the endpoint receiving the information. Omitting such information would enable malicious third parties to inject erroneous information.

Fortunately, the issue of malicious disinformation can be easily addressed using well known techniques. First, the network entity responsible for injecting the throughput guidance header can encrypt the header and include a cryptographically secure message authentication code. In this way the transport endpoint that receives the throughput guidance header can check that the information was sent by a legitimate entity and that the information has not been tampered.

Section 4 described how the TCP Header information can be signed and encrypted for security purposes. An out-of-band mechanism is currently used to agree upon the set of keys used to encrypt and authenticate the messages exchanged between the endpoint and the network element that generate the throughput guidance headers.

Furthermore, the throughput guidance information should be treated only as a hint to the congestion control algorithm running at the transport endpoint. The endpoint that receives this information should not assume that it is always correct and accurate. Specifically, endpoints should check the validity of the information received and if they find it erroneous they should discard it and possibly take other corrective actions (e.g., discard all future throughput guidance information from a particular IP prefix).

One way to check if the throughput guidance information overestimates the capacity available on the radio link is to check whether any packet losses or other signs of congestion (e.g., increasing RTT) occur after the guidance is used. Notably, the same mechanism can be used to deal with bottlenecks in other parts of the the end-to-end network path. To check if the throughput guidance underestimates the available network capacity, the source can periodically attempt to send faster and then check for signs of congestion.

7. IANA considerations

In the current version of the document and for field tests, the experimental value 253 is used for the "Throughput Guidance" TCP option kind.

Note that in this case, following RFC 6994 [RFC6994], a two-byte experiment ID field SHOULD follow immediately after the Length field to allow for shared use of the experimental values. The figure below shows how the throughput guidance option header will look in this case. ExpID SHOULD be set to 0x6006 (16 bits).

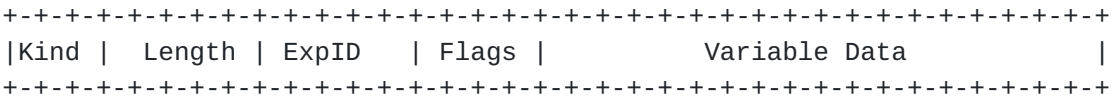


Figure 9

In future versions of the document, a code point should be assigned for the MTGSP Kind field.

8. Acknowledgements

9. References

9.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

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

[RFC6994] Touch, J., "Shared Use of Experimental TCP Options", RFC 6994, August 2013.

9.2. Informative References

[I-D.narten-iana-considerations-rfc2434bis]
Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.

[MEC_White_Paper]
"Mobile-Edge Computing - Introductory Technical White Paper", September 2014, <http://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC4413] West, M. and S. McCann, "TCP/IP Field Behavior", RFC 4413, March 2006.

Authors' Addresses

Ankur Jain
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Phone: +1-925-526-5879
Email: jankur@google.com

Andreas Terzis
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Phone: +1-650-214-5270
Email: aterzis@google.com

Nurit Sprecher
Nokia Networks
Hod HaSharon
IL

Phone: +97297751229
Email: nurit.sprecher@nsn.com

Peter Szilagyi
Nokia Networks
Budapest
Hungary

Phone: +36209777797
Email: peter.1.szilagyi@nsn.com

Hannu Flinck
Nokia Networks
Helsinki
FI

Phone: +358504839522
Email: hannu.flinck@nsn.com

Swaminathan Arunachalam
Nokia Networks
Irving
US

Phone: +19723303204
Email: swaminathan.arunachalam@nsn.com

Csaba Vulkan
Nokia Networks
Budapest
Hungary

Phone: +36209777797
Email: csaba.vulkan@nsn.com