Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

A. Jain
A. Terzis
Google
H. Flinck
N. Sprecher
S. Arunachalam
Nokia Networks
K. Smith
Vodafone

March 9, 2015

Mobile Throughput Guidance Inband Signaling Protocol
draft-flinck-mobile-throughput-guidance-02.txt

Abstract

The bandwidth available for end user devices in cellular networks can vary by an order of magnitude over a few seconds due to changes in the underlying radio channel conditions, as device mobility and changes in system load as other devices enter and leave the network. Furthermore, packets losses are not always signs of congestion. The Transmission Control Protocol (TCP) can have difficulties adapting to these rapidly varying conditions leading to inefficient use of a cellular network's resources and degraded application performance. Problem statement, requirements and the architecture for a solution is documented in [Req_Arch_MTG_Exposure]

This document proposes a mechanism and protocol elements that allow the cellular network to provide near real-time information on capacity available to the TCP server. This "Throughput Guidance" (TG) information would indicate the throughput estimated to be available at the radio downlink interface (between the Radio Access Network (RAN) and the mobile device (UE)). TCP server can use this TG information to ensure high network 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 from live operator's environment.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The problem statement related to the behavior of the TCP in cellular networks is provdied in [Req_Arch_MTG_Exposure]. That same document specifies the requirements, reference architecture and proposed solution principles for a mobile throughput guidance exposure mechanism that can be used to assist TCP in cellular networks, ensuring high utilization and high service delivery performance.

This document presents a set of considerations and assumptions for the development of a solution. It specifies a protocol that addresses the requirements and the architecture stated in the [Req_Arch_MTG_Exposure]. This document describes also the applicability of the proposed mechanism to video delivery over cellular networks with test results from live production environment.

1.1. Contributing Authors

The editors gratefully acknowledge the following additional contributors: Peter Szilagyi/Nokia, Csaba Vulkan/Nokia, Ram Gopal/Nokia, Guenter Klas/Vodafone and Peter Cosimini/Vodafone.

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
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IV	Initialization Vector
LTE	Long Term Evolution
MTG	Mobile Throughput Guidance
RAN	Radio Access Network
RCTP	RTP Control Protocol
RTT	Round Trip Time
SACK	Selective Acknowledgement
TCP	Transmission Control Protocol
TCP-EDO	TCP Extended Data option
TG	Throughput Guidance
UE	User Equipment

1.4. Definitions

Throughput Guidance Provider:

A functional element in the RAN that signals to the TCP server the information on the (near-real time) throughput estimated to be available at the radio downlink interface

1.5. Assumptions and Considerations for the Solution

This document specifies a solution protocol that is compliant with the requirements and architecture specified in [Req_Arch_MTG_Exposure]. The protocol is used by the cellular network to provide throughput guidance information to the TCP server; this information indicates the throughput estimated to be available at the radio downlink interface for the TCP connection. The protocol allows the information to be provided in near real time in situations where the network conditions are changing frequently or the user is moving.

While the implementation details can vary according to the access technology, the resource allocation is abstracted as the capacity of the "radio link" between the RAN 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 (Mbit/s). From the derived UE's total throughput and with the

UE's TCP flow information, Throughput guidance for the TCP connection can be computed.

The TCP server can use this explicit information to inform several congestion control decisions. For example: (1) selecting the initial congestion 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" change. In other words, with this additional information, TCP neither has to congest the network when probing for available resources (by increasing its congestion window), nor rely on heuristics to decide how much it should reduce its sending rate after a congestion episode.

The same explicit information can also be used to optimize application behavior 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 solution specified in this document also satisfies the following assumptions and considerations:

- The end-to-end traffic is delivered via HTTP.

- The end-to-end traffic is encrypted (through HTTPS), thus HTTP header enrichment cannot be used by intermediate elements between the client and the server.

- TCP is used to deliver the HTTPS traffic.

- The Real-time Transport Protocol (RTP) network protocol is not used for traffic delivery.

The protocol specified in this document assumes that a trustful relationship between the Throughput Guidance Provider and the TCP server has been formed using the means discussed in the Security considerations section.

The solution in this document satisfies the considerations and the assumptions presented above, and proposes an in-band exposure mechanism where the throughput guidance information is added to the TCP headers of the relevant upstream packets. HTTP and TCP are the most prevalent protocols in the Internet, used even by the most popular streaming application. Throughput guidance at TCP level can be shared among multiple applications; it is not limited to any particular application level optimization only but it offers a generic approach that works even if application level end-to-end encryption, e.g HTTPS, is applied.

In particular, the Throughput Guidance Providers adds the throughput guidance information to 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 that would be needed with out of band approaches such as with RCTP that is limited to only certain types of applications. Furthermore, an in-band mechanism can keep up with the rapid changes in the underlying radio link throughput. 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 for the TCP connection at the radio link between the RAN and the UE.

Note that once standardized and implemented, TCP Extended Data option (TCP-EDO) can be used to carry the throughput guidance information as specified in [tcp-edo] and simplify the use of the TCP Option fields by extending the space available for TCP options. Currently the TCP-EDO is still work in progress and not available in production. Therefore, the use of TCP-EDO to carry throughput guidance is left for the later drafts.

2. Protocol

This section describes the protocol mechanism and the information element that needs to be communicated from the RAN to the TCP remote endpoint. We describe the protocol mechanism and message format for throughput guidance. The protocol mechanism is defined in an extensible way to allow additional information to be specified and communicated. The protocol specification is based on the existing experiments and running code. It is recommended to insert the throughput guidance information to the TCP segments that flow from client to server (see reasoning in "Assumptions and Considerations" section). Most of the time, TCP segments are ACK packets from a client to the server and hence packets are unlikely to be fragmented. However, the described protocol solution can deal with fragmentation.

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

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. (Use of TCP-EDO will lift this constraint once available and deployed). 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. 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 the conveyed information in this document. Additional information can be specified in future.

The Throughput Guidance Provider functional element inserts Mobile Throughput Guidance TCP options only if there is enough space in the TCP header. The Throughput Guidance Provider resides on top of a radio network element see [Req_Arch_MTG_Exposure]).

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 fits into a single TCP header (maximum of 40 bytes if no other options are present). As described below, fragmenting information across multiple packets 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. A third mode, authentication-only mode, is equally feasible and may

use TCP Authentication Option (TCP-AO) (see RFC 5935 [RFC5925]). We will describe the authentication-only mode in detail in future version of this draft. Both modes share a common Kind-Length-Value "option header" structure with a flag field separating the two cases.

2.1. Common Kind-Length-Value header

Mobile Throughput Guidance Signaling uses the common TCP options structure as in [RFC793] with experimental identifier as defined in [RFC6994]. To make Mobile Throughput Guidance Signaling extendible to different use cases a common Kind-Length-Value structure is defined below.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Kind | Length | ExID |Flags|     variable length data          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1

Kind:

Code point 253 for Experimental Opition for 16-bit ExID [RFC6994]. The size of this field is 1 byte.

Length:

A 1 byte field, length of the option in bytes as defined in RFC793.

ExID

Two bytes Experimental Identifier according to [RFC6994]. Code point 0x6006.

Flags:

One byte of MTG protocol flag field as defined below.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
| Seq |Frag |P|T|
+-+-+-+-+-+-+-+-+
```

Flag field of common Kind-Length-Value header

Figure 2

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 (i.e. option data) in <type, value> format. The content depends of the transfer mode as defined in the following sections of this document. If the option data is fragmented across multiple headers the first fragment (marked with Frag=001 in the Flags-field) contains "Total Length of Data"-field that is the length of the variable data of MTG in all the fragments. Total Length of Data field is followed the content in <type, value>-format.

As an example for the use of the Flags-field, consider a cipher text of a single block. For it the T-bit is set to one, P-bit is set to

zero, Fragment and Seq-fields are zero in the Flags-field. In case the cipher text option 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 first fragment contains the "Total Length of Data"-field indicating the total length of the data to be fragmented. Last fragment is marked with all Frag-bits set to 1 (Frag= 111 for the last fragment). Therefore, the maximum number of fragments is seven. Details follow in the next sections.

2.2. Plain 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 plain text mode is made of one or more type-value pairs. The type determines the length of the following value.

Table of Type Value pairs of Throughput Guidance option data

Name	Type	Length	Unit of the type
Throughput Guidance	1	2 bytes	Mbits/s

Table 1: MTG type-vale pairs

g The Type 1 element carries the actual throughput estimate in the 16-bit value field 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.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Kind | Length |ExID|Flags |Type1|Value-1| Type2|Value-2|     ...     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    Kind, Length, ExID remains same as described in section 2.1.
    Options data constitutes the Flags and the variable length data.
    Flags: P- and T-bits set to zero
```

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

Figure 3

2.3. Encrypted mode

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

```
                  +-+-+-+-+        +-+-+-+-+-+
+-+-+-+-+-+      | key1  |        |IV(Nonce)|
|key index| -->  +-+-+-+-+        +-+-+-+-+-+
+-+-+-+-+-+      | key 2 |             |
                 +-+-+-+-+  key  +-+-+-V-+-+-+-+
                     ...   ----> |  AES 128-CNT|
                 +-+-+-+-+       +-+-+-+-+-+-+-+
                 | key n |             |
                 +-+-+-+-+             +<------ Plain text
                                       |
                                 +-+-+-V-+-+-+-+
                                 | Cipher text |
                                 +-+-+-+-+-+-+-+
```

Encryption method

Figure 4

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 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 exchanged via an out-of-band key management mechanism such as Diffie-Hellman key exchange; this is 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 to 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 single option space of a single TCP header.

```
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+
       |Kind | Length |ExID| Flags | Key Index |first block of 16 bytes
|      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+
```

Kind, Length, ExID remains same as described in section 2.1.
Options data constitutes the Flags and the variable length data.
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 5

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 is not enough space to accommodate the 16 bytes in the option data, the data is fragmented.

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 |Length|ExID|Flags| Total Length|KeyIndex|1. block |fragmented
block  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+
```

Kind, Length, ExID remains same as described in section 2.1
Options data constitutes the Flags, Total Length, Key Index and the variable length data.
Flags: Type of cipher text T-bit = 1, Frag field = 001 first fragment
Total Length: total number of bytes of option data to be fragmented
Key Index is the index used in encryption

Second fragment if the last one:

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+
     |Kind | Length | ExID |Flags| Key Index | Rest of the fragmented
block      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+
```

Kind, Length, ExID remains same as described in section 2.1
Options data constitutes the Flags, Key Index and the variable length data.
Flags: Type of cipher text T-bit = 1, Frag field = 111 last fragment, otherwise 010.
Total Length: total number of bytes in the fragments
Key Index is the index used in encryption

Cipher text layout extending to two consecutive headers

Figure 6

2.4. 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 | Length | ExID |Flags| Key Index |    Nonce (IV)  16 bytes
|         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+

       Kind, Length, ExID remains same as described in section 2.1
           Options data constitutes the Flags and the variable length data.
           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 7

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.

2.5. 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-ExID-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, ExID, 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, ExID, 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.

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 connection using the Options field in the TCP header (according to the message specification provided in section 2). 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 for that TCP connection.

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.

3.1. Test Results

Nokia Networks and Google tested the video delivery optimization use case in a live production environmentDifferent network load scenarios were taken into consideration. TCP Cubic was used in these tests and the TG information was used by the TCP based video server to adjust TCP congestion window only. The results below are based on data for whole 2 days (23rd and 25th Feb 2015).

All network level metrics showed an average improvement of 30-60%, as detailed below:

- o Reduction of end-to-end TCP RTT by 55-70%
- o TCP retransmissions reduced by 30-45%
- o Mean Client Throughput improved by 20-35%
- o TCP packet loss reduced by 35-50%

The application-level metrics show an average improvement as detailed below:

- o Click-to-play time reduced by 5-20%
- o Average video resolution improvement by 5- 20%
- o Reduction in the number of format changes by 10 - 25%

These user experience improvements results in faster video time to play and are likely to result in longer battery life.

4. Manageability considerations

The application in the RAN SHOULD be configured with a list of destinations to which throughput guidance should be provided. The application in RAN will supply mobile throughput guidance information to more than one TCP server simultaneously based on the list of destinations.

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.

5. Security considerations

Throughput guidance is considered confidential information and it SHOULD be provided in a secure way. The information can be provided as plain text in a secure and closed network (e.g. inside operator 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).

Section 2 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 generates the throughput guidance headers.

As stated in [Req_Arch_MTG_Exposure], the policy configuration of the Throughput Guidance Provider and the server endpoint, as well as the key management and the encryption algorithm are beyond the scope of this protocol definition. The protocol assumes that a trustful relationship has been formed between the Throughput Guidance Provider and the TCP server and that the required security level is already configured by the operator and agreed between the entities (i.e. authentication, encryption or both).

The identity of the Mobile Throughput Guidance provider 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 with.

Furthermore, the throughput guidance information should be treated only as an estimate 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).

The impact of TCP Authentication Option (TCP-AO) with encrypted TCP segment payload [tcp-ao-encrypt] implies that the Throughput Guidance Provider functional element acts as a full back to back TCP proxy. This case is left for later stages as the work [tcp-ao-encrypt] is still at draft stage.

6. 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. ExpID SHOULD be set to 0x6006 (16 bits)

7. Acknowledgements

8. References

8.1. 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.

8.2. 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]
ETSI, "Mobile-Edge Computing - Introductory Technical White Paper", 2014.

[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.

[Req_Arch_MTG_Exposure]
Jain, A., Terzis, A., Sprecher, N., Arunachalam, S., Smith, K., and G. Klas, "Requirements and reference architecture for Mobile Throughput Guidance Exposure", draft-sprecher-mobile-tg-exposure-req-arch-01.txt (work in progress), February 2015.

[tcp-ao-encrypt]
Touch, J., , "A TCP Authentication Option Extension for Payload Encryption", draft-touch-tcp-ao-encrypt-02.txt (work in progress), November 2014.

[tcp-edo] Touch, J., and Eddy, W., "TCP Extended Data Offset Option", draft-ietf-tcpm-tcp-edo-01.txt (work in progress), October 2013.

Appendix A.

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

Hannu Flinck
Nokia Networks
Helsinki
FI

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

Nurit Sprecher
Nokia Networks
Hod HaSharon
IL

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

Swaminathan Arunachalam
Nokia Networks
Irving, TX
US

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

Kevin Smith
Vodafone
One Kingdom Street, Paddington Central
London W2 6BY
UK

Email: kevin.smith@vodafone.com