Network Working Group
Internet Draft
Intended status: Experimental
Expires: August 20,2024

J. Zhu
M. Zhang
Intel

February 20, 2024

A UDP-based GMA (Generic Multi-Access) Protocol

draft-zhu-intarea-gma-control-05

Abstract

A device can simultaneously connect to multiple access networks, e.g., Wi-Fi, LTE, 5G, DSL, and SATCOM (Satellite Communications). It is desirable to seamlessly combine multiple connections over these networks below the transport layer (L4) to improve quality of experience for applications that do not have built-in multi-path capabilities. This document presents a new convergence protocol for managing data traffic across multiple network paths. The solution has been developed by the authors based on their experiences in multiple standards bodies including IETF and 3GPP, is not an Internet Standard and does not represent the consensus opinion of the IETF. This document will enable other developers to build interoperable implementations to experiment with the protocol.

Status of this Memo

Copyright Notice

Table of Contents

1 Introduction

A device can simultaneously connect to multiple networks, e.g., Wi-Fi, LTE, 5G, DSL, and SATCOM (Satellite Communications). It is desirable to seamlessly combine multiple connections over these networks below the transport layer (L4) to improve quality of experience for applications that do not have built-in multi-path capabilities.

A new Multi-Access Management Service (MAMS) framework has been recently specified in [MAMS] to support various multi-access solutions [ATSSS] [LWIPEP] [GRE1] [GRE2]. As shown in Figure 1, its user-plane protocol stack consists of two layers: convergence and adaptation. The convergence layer is responsible for multi-access operations, including multi-link (path) aggregation, splitting/reordering, lossless switching/retransmission, etc. It operates on top of the adaptation layer. From the perspective of a transmitter, a user payload (e.g., IP packet) is processed by the convergence layer first, and then by the adaptation layer before being transported over a delivery connection; from the receiver's perspective, an IP packet received over a delivery connection is processed by the adaptation layer first, and then by the convergence layer.

```
          +-----------------------------------------------------+
          |   User Payload, e.g., IP Protocol Data Unit (PDU)   |
          +-----------------------------------------------------+
       +-------------------------------------------------------------+
       |  +-----------------------------------------------------+  |
       |  | Multi-Access (MX) Convergence Layer                 |  |
       |  +-----------------------------------------------------+  |
       |  +-----------------------------------------------------+  |
       |  | MX Adaptation    | MX Adaptation    | MX Adaptation    |  |
       |  | Layer            | Layer            | Layer            |  |
       |  +-----------------+-----------------+-----------------+  |
       |  | Access #1 IP     | Access #2 IP     | Access #3 IP     |  |
       |  +-------------------------------------------------------+  |
       |                            MAMS User-Plane Protocol Stack |
       +-------------------------------------------------------------+
```

Figure 1: MAMS User-Plane Protocol Stack [MAMS]

A new Generic Multi-Access (GMA) encapsulation protocol [GMAE] has been specified to encode additional control information, e.g., Timestamp, Sequence Number, to support multi-access operations at the convergence layer. This document presents a UDP-based GMA control protocol for the convergence layer, and enhancements to the GMA encapsulation protocol. The GMA protocol only operates between endpoints that have been configured to use GMA through MAMS management messages [MAMS] or other management methods, which is out of the scope of this document.

From the perspective of applications, the GMA protocol is a multi-path tunneling protocol operating below the network layer (L3), and therefore can support any legacy single-path transport protocol, e.g. TCP, UDP, QUIC, etc. From the perspective of a underlay access network, it is a light-weight transport protocol designed specifically for multi-path operation, removing unnecessary complexity and overhead (e.g., end-to-end encryption, congestion control, reliable transmission, etc.) as seen in a modern transport protocol [QUIC]. Moreover, it can be easily extended to support advanced multi-path operations, e.g., network coding, network-based traffic steering, in-band QoS monitoring, etc.

The solution described in this document has been developed by the authors based on their experiences in multiple standard bodies including the IETF and 3GPP. However, it is not an Internet Standard and does not represent the consensus opinion of the IETF. This document presents the protocol specification to enable experimentation as described in Section 1.1 and to facilitate other interoperable implementations.

1.1 Scope of Experiment

The protocol described in this document is an experiment. One objective of the experiment is to determine whether the protocol meets the 3GPP multi-access requirements [ATSSS2] [Dual3GPP], can be safely used, and has support for deployment. Particularly, the proposed GMA protocol addresses the following issues of using QUIC for ATSSS:

- o Encapsulation Overhead: the GMA encapsulation protocol uses a 2-bytes Flag field to control all optional header fields instead of the TLV (Type-Length-Value) based approach. As a result, the minimum encapsulation overhead is 2 bytes, and the maximum is 16 bytes.
- o Multiple Encryptions: the GMA encapsulation protocol does not mandate encryption to avoid unnecessary encryption overhead when a delivery connection is secure and trusted.
- o Congestion Control in Congestion Control: the GMA control protocol does not mandate congestion control. All incoming packets (from higher layer) MAY be sent out without any delay due to congestion control.

In addition, the GMA protocol makes reliable delivery optional, assuming it has beenaddressed by the application or transport layer. Hence, it does not require Acknowledgement (ACK) for data packets, and can avoid any delay due to retransmission or ACK overhead on the reverse path.

The GMA protocol supports both out-of-band and in-band path quality measurements (e.g. one-way-delay, loss, etc.) and congestion detection. A (out-of-band) control message, e.g. probe, with acknolwedgement can be used to actively measure round trip time and reliability of a connection. While the GMA header fields, e.g. sequence number, timestamp, etc., in the GMA header of a received data packet can be used for in-band measurement. Another objective of the experiment is to evaluate state-of-the-art congestion detection algorithms [GCC] [MPIP] [DCTCP] for multi-path traffic management.

It is expected that this protocol experiment can be conducted on the Internet since GMA packets are encapsulated with UDP. Thus, experimentation is conducted between consenting end systems that have been mutually configured to participate in the experiment. An open-source based GMA software implementation [GMA] has been provided for this experiment. The authors will continually assess the progress of this experiment and encourage other implementers to contact them to report the status of their implementations and their experiences with the GMA protocol.

2 Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3 Use Case

As shown in Figure 2, a client device (e.g., Smartphone, Laptop, etc.) may connect to the Internet via both Wi-Fi and LTE connections, operating as a delivery connection. In addition, a virtual (e.g. IPv4, IPv6, or Ethernet) connection is established between client and multi-access gateway. The virtual connection is the anchor, providing the IP address and connectivity for end-to-end Internet access, and delivery connections provide multiple paths between client and gateway to support multi-access traffic management.

```
         +------- Virtual (anchor) Connection ------+
         |                                          |
        +-+---+                                +---+-+
        | | |A|--- LTE (delivery) Connection --|C| | |
Apps ---|X|U|-|                                |-|S|Z|--- Internet
        | | |B|-- Wi-Fi (delivery) Connection--|D| | |
        +-+---+                                +---+-+
        Client                                  Gateway
```

```
 o A: the adaptation layer endpoint of the LTE connection in the
      client
 o B: the adaptation layer endpoint of the Wi-Fi connection in the
      client
 o C: the adaptation layer endpoint of the LTE connection in the
      multi-access gateway
 o D: the adaptation layer endpoint of the Wi-Fi connection in the
      multi-access gateway
 o U: the convergence layer endpoint in the client
 o S: the convergence layer endpoint in the multi-access gateway
 o X: the virtual connection endpoint in the client
 o Z: the virtual connection endpoint in the multi-access gateway
```

Figure 2: GMA-based Multi-Access Traffic Management

For example, the virtual connection could be a Multi-Access Protocol Data Unit (MA-PDU) connection as specified in 3GPP [ATSSS]. Per-packet aggregation allows the MA-PDU connection to use the combined bandwidth of multiple delivery connections. Moreover, packets may be duplicated over multiple connections to achieve high reliability and low latency, where duplicated packets are eliminated by the receiving side. Such multi-access traffic management requires additional control message exchange between client and gateway.

UDP is used as the adaptation layer protocol in this document. Figure 3a and 3b show UDP-based GMA user-plane and control-plane protocol, respectively. The "UDP Tunnelling" ports at client are chosen from the dynamic port range, and at gateway are configured and provided to clients through MAMS messages, e.g., MX UP Setup Config [MAMS].

```
         +-----------------------------------------------------+
         |        Virtual Connection (IP, Ethernet, etc.)      |
         +-----------------------------------------------------+
         |        UDP-based GMA Encapsulation                  |
         +-----------------------------------------------------+
         |     UDP         |       UDP       |      UDP        |
         +-----------------+-----------------+-----------------+
         | Access #1 IP    | Access #2 IP    | Access #3 IP    |
         +-----------------------------------------------------+
```

Figure 3a: UDP-based GMA User-Plane Protocol Stack

```
         +-----------------------------------------------------+
         |             GMA Control Messages                    |
         +-----------------------------------------------------+
         |        UDP-based GMA Encapsulation                  |
         +-----------------------------------------------------+
         |     UDP         |       UDP       |      UDP        |
         +-----------------+-----------------+-----------------+
         | Access #1 IP    | Access #2 IP    | Access #3 IP    |
         +-----------------------------------------------------+
```

Figure 3b: UDP-based GMA Control-Plane Protocol Stack

4 UDP-based GMA Encapsulation Protocol

```
         +----------------------------------------------------+
         | IP hdr | UDP hdr  | GMA Header | Payload (GMA SDU) |
         +----------------------------------------------------+
```

Figure 4: UDP-based GMA PDU Format

Figure 4 shows the UDP-based GMA encapsulation format [GMAE]. The GMA header consists of the mandatory "Flags" field (the first two bytes), defined as follows:

- o Client ID Present (bit 0): If the bit is set to 1, the Client ID field is present.

- o Payload Type (PT) (bit 1): If the bit is set to 1, the GMA PDU carries a GMA control message or an encrypted GMA SDU (data). Otherwise (default), it carries an unencrypted GMA SDU (data).
- o Flow ID Present (bit 2): If the bit is set to 1, the Flow ID field is present.
- o Per-Packet Priority (PPP) Present (bit 3): If the bit is set to 1, the PPP field is present.
- o Packet Group Identification (PGI) Present (bit 4): If the bit is set to 1, the PCI field is present.
- o Delivery SN Present (bit 5): If the bit is set to 1, the Delivery SN field is present.
- o Flow SN Present (bit 6): If the bit is set to 1, the Flow SN field is present.
- o Timestamp Present (bit 7): If the bit is set to 1, the Timestamp field is present.
- o Reserved (bit 8-15): set to "0" and ignored on receipt.

Bit 0 is the most significant bit (MSB), and bit 15 is the least significant bit (LSB).

The receiver SHOULD first decode the Flags field to determine the length of the GMA header, and then decode optional fields accordingly. The GMA header MAY consist of the following optional fields:

- o Client ID (2 Byte): an unsigned integer to identify the virtual connection. A client may establish multiple virtual connections, e.g. MA-PDU, with the gateway, each of which SHOULD be provided with a unique "Client ID".
- o PT (1 Byte)
  - + Bit 0: the Key Phase bit to indicate which key is used to protect the GMA payload.
  - + Bit 1~7: the GMA control message type, set to "0" if the payload is an encrypted GMA SDU
- o Flow ID (1 Byte): an unsigned integer to identify the IP flow
  - + 0: unknown flows
  - + 1~20: reserved for flows using the redundancy mode, with which a flow may be duplicated over the available delivery connections.
  - + 21~50: reserved for flows using the splitting mode, with which a flow may be split over the available delivery connections.
  - + 51~100: reserved for flows using the steering mode, with which a flow is steered to only one of the available delivery connections.
  - + Others: reserved for future use
- o Per-Packet Priority (1 Byte): an unsigned integer to identify the relative priority of the GMA SDU in the flow (smaller value means higher priority).

- Packet Group ID (1 Byte): an unsigned integer to identify the group of GMA SDUs. If one GMA SDU in the group is dropped, other GMA SDUs in the same group SHOULD also be dropped. For example, all GMA SDUs from a video frame MAY be classified into a same group.
- Delivery SN (1 Byte): an auto-incremented unsigned integer to indicate the GMA PDU transmission order on a delivery connection. Delivery SN is used for a flow using the splitting mode to measure packet loss of each delivery connection and generated per delivery connection per flow. It SHOULD be ignored or not used for flows with the reududnancy or splitting mode
- Flow SN (3 Bytes): an auto-incremented unsigned integer to indicate the GMA SDU (IP packet) order of a flow. Flow SN is used for reordering, and generated per flow.
- Timestamp (4 Bytes): an unsigned integer to indicate the value of the timestamp clock at the transmitter in the unit of 100 microseconds when a GMA PDU is transmitted.

The use of Key Phase bit is similar to what is specified in QUIC [QUICTLS], allowing a recipient to detect a change in keying material without needing to receive the first packet that triggered the change. The Key Phase bit is initially set to 0 and toggled to signal each subsequent key update. The Key Phase bit SHALL be ignored if the payload is not encrypted or authenticated.

Figure 5 shows the GMA header format with all the fields present, and the order of the GMA control fields SHALL follow the bit order in the Flags field. Note that the bits in the Flags field are ordered with the first bit transmitted being bit 0 (MSB). All fields are transmitted in regular network byte order and appear in the order of their corresponding flag bits. If a flag bit is not set, the corresponding optional field is absent.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Flags       |   reserved    |             Client ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PT       |  Flow ID      |    PPP         |       PGI     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Delivery SN  |           Flow SN                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Timestamp                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: GMA Header Format with all Optional Fields Present

Some GMA header fields, e.g. Client ID, Flow ID, and PPP are designed to support fine granular packet classification. Notice that GMA header fields (unlike IP header field) won't change regardless of how a GMA PDU is delivered, since they are encapsulated as part of UDP payload. Therefore, an intermediate node, e.g. router, Access Point, Base Station, etc., can perform active queue management (AQM) based on these GMA header fields directly.

Other GMA header fields, e.g. Delivery SN, Flow SN, and Timestamp, are designed to support multi-path traffic management. For example, Flow SN allows reordering at the receiver when a flow is split over multiple connections. In the meantime, Delivery SN is needed for packet loss measurement per delivery connection especially if a flow uses the splitting mode, and Timestamp allows in-band one-way-delay (OWD) measurement, which can then be used to detect congestion and buffer overflow at intermediate nodes. Moreover, Delivery SN and Flow SN can be used to support the Fast Packet Loss Detection mechanism as described in [MPSN] for minimizing multi-path reordering delay.

GMA payload MAY be protected by a symmetric key cipher, e.g. AES256-GCM. A GMA receiver (e.g. gateway) uses the Client ID field to determine the corresponding key for decryption. Moreover, the GMA payload is encrypted and the GMA header is authenticated but not encrypted.

GMA SDU (data) SHOULD be protected by the symmetric key only if the delivery connection is "untrusted" and subject to malicious attacks. If the encrypted GMA payload carries GMA SDU (data), the PT field MUST be present in the GMA header and the GMA control message type field MUST be set to "0". In other words, an encrypted GMA data SDU is encapsulated as a special control message.

```
+--------------------------------------------------------------+
|   GMA Header    |     GMA Payload          | GCM Tag | IV   |
+--------------------------------------------------------------+
|<-authenticated->|<-------encrypted -------->|
```

Figure 6: AES256-GCM Encrypted GMA Message

Figure 6 shows the format of an AES256-GCM encrypted GMA message, where IV (initialization vector) is 12 bytes long and GCM Tag is

16 bytes long. The GMA header is used as additional authenticated data (AAD).

5 GMA Control Messages

The GMA header of a GMA control message consists of Client ID, Payload Type, Flow SN, and Timestamp. All GMA control messages share the same Flow SN space. Table 1 lists all the GMA control messages specified in this document and their value of "Type" in the GMA header.

Notice that Coded GMA SDU (CGS) message (5.16) SHOULD be protected by the symmetric key only if the delivery connection is untrusted. All other GMA control message SHOULD be protected regardless.

Table 1: GMA Control Messages

Type	GMA Control Message	Section
0	Encrypted GMA SDU (data)	N/A
1	Probe	5.1
2	ACK	5.2
3	Traffic Splitting Update (TSU)	5.3
4	Traffic Splitting Ack (TSA)	5.4
5	Delivery Connection Reconfiguration (DCR)	5.5
6	Key Update	5.6
7	Traffic Steering Command (TSC)	5.7
8	Traffic Splitting Command (TSP)	5.8
9	QoS Testing Request (QTR)	5.9
10	QoS Testing Response (QTP)	5.10
11	QoS Testing Notification (QTN)	5.11

12	QoS Violation Notification (QVN)	5.12
13	Packet Loss Report (PLR)	5.13
14	First Sequence Number (FSN)	5.14
15	Coding Configuration Request (CCR)	5.15
16	Coded GMA SDU (CGS)	5.16
17	Connection Priority Reconfiguration (CPR)	5.17

5.1 Probe Message

A client (or gateway) MAY send out a Probe message for initial connection establishment, path quality estimation, keep-alive, timestamp synchronization, and link measurement report. In response, the gateway (or client) SHOULD send back the ACK message if it is required in the corresponding Probe message.

A control messages may be retransmitted if it is not acknowledged within a timeout period. It is left to implementation to configure the retransmission timer and the maximum number of retransmission attempts. Flow SN SHOULD be adjusted incrementally regardless of whether a control message is new or retransmitted. A delivery connection is established after a successful handshake of Probe and ACK, and terminated if any control message can't be successfully acknolwedged after retransmissions.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| LS Bitmap     |  Probing Flag |       N       |     CID       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    K          |     TLV based Link Information Elements       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Probe Message Format

A Probe message consists of the following mandatory fields:

- o Link Status (LS) Bitmap (1 Byte): to indicate the status (0: not connected; 1: connected) of the i-th delivery connection, where connections are ordered according to their CID, bit #7 (LSB) corresponds to the 1st delivery connection and bit #0 (MSB) corresponds to the 8th delivery connection.

- o Probing Flag (1 Byte)
  - + Bit #0: a bit flag to indicate if timestamp needs to be reset (1) or not (0)
  - + Bit #1: a bit flag to indicate if the ACK message is expected (1) or not (0)
  - + Bit #2: a bit flag to indicate if the receiving side SHOULD update the UDP tunnel end-point (1) or not (0) based on the Probe message
  - + Bit #3: a bit flag to indicate if the client is synchronized (1) or not (0) with the gateway in time.
  - + Bit #4: a bit flag to indicate if Link Information Elements (IE) are present (1) or not (0).
  - + Bit #5~7: reserved

A Probe message with the Bit #0 flag set to "1" is also called Probe-Sync. A client will send out a Probe-Sync message to reset timestamp when its local timestamp timer exceeds a pre-defined threshold, e.g., 0x7FFF0000 and prevent it from overflowing due to the limited size (4 Bytes). Once receiving a Probe-Sync message, the gateway will reset the timestamp timer to "0" for the client and respond with an ACK message. The "Request Type" field in the ACK message is set to 0, indicating the corresponding Probe message is Probe-Sync.

The client SHOULD reset its timestamp timer to "0" after the Probe-Sync message is successfully acknowledged. As a result, the timestamp field in a GMA PDU indicates the time between the last successful Probe-Sync message exchange and the transmission of the GMA PDU.

If the Bit #1 flag is not set, the receiving endpoint SHOULD NOT send back the ACK message.

If the Bit #2 flag is not set, the probe message is used to test the reachability of alternative path of the delivery connection, and therefore the receiving endpoint SHOULD NOT update the UDP tunnel end-point accordingly.

If the Bit #3 flag is set, indicating the client is already synchronized with the gateway in time, they SHOULD use their local clock directly as the timestamp clock without going through the above "Probe-Sync" procedure. How to maintain time synchronization between two GMA endpoints is out of the scope of this document.

If the Bit #4 flag is set, the Probe message consists of the following optional fields:

- o N (1 Byte): the number of delivery connections whose Link IEs are included in the message.

For each connection, include the following fields:

- o Connection ID (1 Byte) to identify the delivery connection
- o K (1 Byte): the number of Link IEs for the connection
- o TLV(Type-Lenth-Value) encoded Link IEs (variable): a GMA client MAY use the Probe message to report its link quality, e.g., signal strength and other information, e.g., Wi-Fi channel number, as shown in Table 2.

Probe may also be used to measure path quality for a specific flow. In this case, the Probe message and its corresponding ACK message SHOULD carry the same QoS classification marking, e.g. DSCP, as a data packet of the flow. In addition, the "Flow ID" field SHOULD be included in the GMA header of the flow-specific Probe and ACK message to identify the flow.

Table 2: GMA Link Information Elements

Name	Type	Length	Value
Wi-Fi RSSI	0	1	-255dBm ~ 0dBm
Wi-Fi Band	1	1	0:2.4GHz, 1: 5GHz, 2:6GHz
Wi-Fi Channel	2	1	0~255
Wi-Fi BSSID	3	6	Wi-Fi AP MAC address
Wi-Fi Bandwidth	4	1	0 ~ 255 x 10Mbps
Wi-Fi Type	5	1	0: IEEE 802.11 a/b/g/n 1: Wi-Fi 6 2: Wi-Fi 7
Cellular RSRQ	30	1	-255dB ~ 0dB
Cellular RSRP	31	1	-255dBm ~ 0dBm

Cellular RSSI	32	1	-255dBm ~ 0dBm
GSM Cell ID	33	4	0 ~ 2^32 - 1
Cellular Type	34	1	0: 3G 1: 4G LTE 2: 5G NR

5.2   Acknowledgement (ACK) Message

An ACK message is used to confirm the successful reception of a control message unless it is not required or a specific acknolwedge message, e.g. TSA (5.4), is required. The source IP address and UDP port of the control message SHOULD be used as its desintation IP address and UDP port.

The Flow SN field in the GMA header of the ACK message is set to the Flow SN of the corresponding control message. The ACK message consists of the following fields:

- o Request Type (1 Byte): the corresponding control message type, e.g. Probe, etc.
- o Padding (variable)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Request Type  |     Padding                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Ack Message Format

5.3   Traffic Splitting Update (TSU) Message

A TSU message is used by the receiving endpoint of a data flow to update traffic splitting or steering configuration at the transmitting endpoint. Unlike a probe, the gateway SHOULD always update the UDP tunnel end-point for a client based on a received TSU message from the client.

A TSU message consists of the following fields:

- o Link Status Bitmap (1 Byte): the same as specified in 5.1
- o K (1 Byte): the number of TLV-encoded TSU IEs

- o TSU IEs (variable)

A TSU IE consists of the following fields:

- o Type (1 Byte)
  - + 0: the traffic steering configuration
  - + 1: the traffic splitting configuration
  - + 2: the minimum OWD (One-Way-Delay) measurement report
  - + Others: reserved
- o Length (1 Byte): the TSU IE length
- o Flow ID (1 Byte): an unsigned integer to identify the flow.

If Type is "0", the TSU IE consists of the following traffic steering configuration parameters:

- o C (1 Byte): the CID of the delivery connection that the flow should be using.

For a flow with the redundancy mode, the traffic steering configuration IE MAY consist of multiple CID fields to indicate which delivery connections will be used to send duplicated packets of the flow.

If Type is "1", the TSU IE consists of the following traffic splitting configuration parameters:

- o L (1 Byte): the total number of packets per traffic splitting cycle, e.g., L = 32, and each packet is assigned an index from 0 to L-1.
- o S1[i] (N Bytes): the index of the first packet sent over the i-th delivery connection per traffic splitting cycle, where connections are ordered according to their Connection ID and i = 1, 2,..., N.
- o S2[i] (N Bytes): the index of the last packet sent over the i-th delivery connection per traffic splitting cycle, where connections are ordered according to their Connection ID and i = 1, 2,..., N.

For example, with two delivery connections, i.e., N=2, the configuration of S1[1] = S2[1] = 0, S1[2] = S2[2] = 1 and L = 2 indicates sending one packet of every two packets over the first connection, and the other one over the second connection.

If Type is "2", the TSU IE consists of the following minimum OWDmeasurement report parameters:

o D[i] (N Bytes): an unsigned integer (0 ~ 254) to indicate the minimum OWD measurement (in milliseconds) of the i-th delivery connection.
 + 255: reserved

Notice that the GMA endpoints (client and gateway) may not be synchronized in time, and therefore the absolute value of minimum OWD (d(i)) is not useful. Instead, the difference between the minimum OWD of a connection and the maximum is reported, i.e.,

```
                  D(i) = max(d(i) | i = 1 ~ N) - d(i)
```

It indicates how much delay should be added by the GMA transmitting endpoint to equalize minimum OWD across delivery connections and mitigate the impact of reordering.

Figure 9 shows the TSU message format for two flows with the splitting mode and the steering mode, respectively. In addition, the minimum OWD measurement report IE is included for the splitting flow.

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  LS Bitmap    |       K       |   IE Type(=0) | IE Length(=4) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flow ID   |       C       |   IE Type(=1) | IE Length(=10)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flow ID   |       L       |     S1[1]     |     S1[2]     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     S2[1]     |     S2[2]     |   IE Type(=2) | IE Length(=5) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Flow ID   |      D[1]     |     D[2]      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: TSU Message Format (K = 2, N = 2)

5.4 Traffic Splitting Acknowledgement (TSA) Message

A TSA message is used to confirm the successful reception of a TSU message. The Flow SN of the TSA message is set to the Flow SN of the corresponding TSU message. A TSU message consists of the following fields:

o K (1 Byte): the number of TSA IEs
o TSA IEs

A TSA IE consists of the following fields:

- o Type (1 Byte)
  - + 0: the Start SN configuration
  - + 1: the OWD adjustment configuration
  - + Others: reserved
- o Length (1 Byte): the TSA IE length

If Type is "0", a TSA IE consists of the following fields for each flow configured in the TSU message:

- o Flow ID (1 Byte): an unsigned integer to identify the flow.
- o StartSN (3 Bytes): the Flow SN of the first GMA SDU using the configuration provided by the corresponding TSU message.

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      K        |   IE Type(=0) | IE Length(=10)|    Flow ID    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 StartSN                       |    Flow ID    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 StartSN                       |  IE Type(=1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  IE Length(=5)|    Flow ID    |     D[1]      |     D[2]      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: TSA Message Format (K=2, N=2)

If Type is "1", a TSA IE consists of the following fields for the flow reporting its Minimum OWD in the TSU message:

- o Flow ID (1 Byte): an unsigned integer to identify the flow.
- o D[i] (N Bytes): a signed integer (-128~127) to indicate the delay adjustment in milliseconds for the i-th delivery connection based on the minimum OWD measurement in the TSU message.

Figure 11 shows the GMA traffic splitting reconfiguration procedure for downlink traffic, where the client (receiver) performs path quality measurement based on received packets and reconfigures traffic splitting parameters at the gateway (transmitter). Once update is needed, the client will send out a TSU message carrying the new traffic splitting configuration parameters to the gateway. The gateway will then send back the TSA message and reconfigure traffic splitting accordingly.

```
          client                                             gateway
              |                                                 |
              |<------------------ GMA SDU #1 ------------------|
              |<------------------ GMA SDU #2 ------------------|
  +--------------------------+                                  |
  | path quality measurement |                                  |
  +--------------------------+                                  |
              |------------------ TSU ------------------------->|
              |<------------------- TSA(StartSN=3)--------------|
              |<------------------ GMA SDU #3 ------------------|
              |<------------------ GMA SDU #4 ------------------|
```

Figure 11: Downlink Traffic Splitting Reconfiguration Procedure

5.5 Delivery Connection Reconfiguration (DCR) Message

The gateway MAY send out a DCR message to enable or disable a delivery connection for a client. In response, the client SHOULD stop sending any (control or data) packet to a disabled connection and set the corresponding bit in the Link Status Bitmap field in Probe and TSU to "0". If a previously disabled delivery connection is enabled by the DCR message, the client SHOULD send out a Probe message to check whether the gateway is reachable via the delivery connection.

A DCR message consists of the following fields:

- o Connection Status Bitmap (1 Byte): to indicate the status (0: disabled; 1: enabled) of the i-th delivery connection, where connections are ordered according to their Connection ID the same way as in Link Status Bitmap (5.1).

5.6 Key Update Message

The gateway MAY send out a Key Update message to change the symmetric key for a client. In response, the client SHOULD start using the new key immediately. The gateway SHOULD start using the new key after receiving the ACK message or a GMA control message with the toggled Key Phase bit.

A Key Update message consists of the following fields:

- o Key Type (1 Byte)
  - + 0: AES256-GCM
  - + Others: reserved

If Key Type == 0

- o Key Value (32 Bytes): the AES256-GCM key

5.7 Traffic Steering Command (TSC) Message

The gateway MAY send out a TSC message to configure network-based traffic steering. A TSC message consists of the following fields:

- o Flag (1 Byte)
  - + 0: disable network-based traffic steering (default)
  - + 1: enable network-based traffic steering
  - + Others: reserved
- o N1 (1 Byte): the number of downlink flows configured in the TSC message.
- o N2 (1 Byte): the number of uplink flows configured in the TSC message.

If Flag == 1 and N1 > 0, the following control parameters are included for each downlink flow:

- o Flow ID (1 Byte): an unsigned integer to identify the flow.
- o CID (1 Byte): the CID of the delivery connection that the downlink flow will be using.

For uplink flow, the TSC message is only used to enable or disable network-based traffic steering, and the TSU message is used for configuration. Therefore, only "Flow ID" fields are included in the TSC message.

5.8 Traffic Splitting Command (TSP) Message

The gateway MAY send out a TSP message to configure network-based traffic splitting for downlink traffic. Uplink traffic splitting is always controlled by the gateway using the TSU message. A TSP message consists of the following fields:

- o Flag (1 Byte)
  - + 0: disable network-based traffic splitting (default)
  - + 1: enable network-based traffic splitting
  - + Others: reserved
- o Number of Flows (1 Byte): the number of downlink flows that are configured in the TSP message

If Flag == 1, the following control parameters are included for each flow:

- o Flow ID (1 Byte): an unsigned integer to identify the flow

- o L (1 Byte): the total number of packets per traffic splitting cycle, e.g. L = 32, and each packet is assigned an index from 0 to L-1.
- o S1[i] (N Bytes): the index of the first packet sent over the i-th delivery connection per traffic splitting cycle, where connections are ordered according to their Connection ID and i = 1, 2, ..., N.
- o S2[i] (N Bytes): the index of the last packet sent over the i-th delivery connection per traffic splitting cycle, where connections are ordered according to their Connection ID and i = 1, 2, ..., N.

5.9 QoS Testing Request (QTR) Message

A client MAY send out a QTR message to request QoS testing for a flow. It consists of the following fields:

- o Flow ID (1 Byte): an unsigned integer to identify the flow for QoS testing.
- o Traffic Direction (1 Byte): an unsigned integer to indicate the direction of flow (0: downlink, 1: uplink, 2: both)
- o CID (1 Byte): the CID of the delivery connection that the QoS testing will be performed.
- o Test Duration (2 Byte): an unsigned integer to indicate the testing duration in ms.

5.10 QoS Testing Response (QTP) Message

A QTP message is used by the receiving endpoint of QoS testing to indicate if the testing is successful or not. It consists of the following fields:

- o Flow ID (1 Byte): an unsigned integer to identify the flow
- o CID (1 Byte): the CID of the delivery connection that the QoS testing has been performed
- o Status: an unsigned integer to indicate the result of QoS testing (0: success; 1: failure)

5.11 QoS Testing Notification (QTN) Message

The gateway MAY send out a QTN message to start QoS testing for a flow. It consists of the following fields:

- o Flow ID (1 Byte): an unsigned integer to identify the flow for QoS testing.
- o Traffic Direction (1 Byte): an unsigned integer to indicate the direction of flow (0: downlink, 1: uplink, 2: both)

- o CID (1 Byte): the CID of the delivery connection that the QoS testing will be performed.
- o Test Duration (2 Byte): an unsigned integer to indicate the testing duration in ms.

5.12 QoS Violation Notification (QVN) Message

The gateway MAY send out a QVN message to indicate that QoS violation has been detected or is expected for a flow. It consists of the following fields:

- o N1 (1 Byte): Number of uplink flows with QoS violation
- o N2 (1 Byte): Number of downlink flows with QoS violation
- o Uplink Flow ID (1 Byte x N1): an unsigned integer to identify uplink flow with QoS violation.
- o Downlink Flow ID (1 Byte x N2): an unsigned integer to identify downlink flow with QoS violation.

5.13 Packet Loss Report (PLR) Message

A PLR message is used by the receiving endpoint to report lost GMA SDUs for example during handover. In response, the transmitter may retransmit lost GMA SDUs accordingly. A PLR message consists of the following fields:

- o Number of Flows (1 Byte): the number of flows

For each flow, the following control parameters are included:

- o Flow ID (1 Byte): an unsigned integer to identify the flow
- o ACK number (3 Bytes): the next (in-order) sequence number (SN) that the sender of the PLR message is expecting
- o Number of Loss Bursts (1 Byte)
  For each loss burst, include the following
  - + Flow SN of the first lost GMA SDU in a burst (3 Bytes)
  - + Number of consecutive lost SDUs in the burst (1 Byte)

5.14 First Sequence Number (FSN) Message

A GMA transmitter MAY send out the FSN messages to indicate the oldest SDU in its buffer if a lost SDU is not found in the buffer after receiving the PLR message. In response, the GMA receiver SHOULD NOT report any packet loss with Flow SN < FSN. A FSN message consists of the following fields:

- o Number of Flows (1 Byte): the number of flows

For each flow, the following control parameters are included:

- Flow ID (1 Byte): an unsigned integer to identify the flow
- First Sequence Number (3 Bytes): the sequence number (SN) of the oldest SDU in the (retransmission) buffer.

5.15 Coding Configuration Request (CCR) Message

A CCR message is used to support packet loss recovery through systematic network coding, e.g. XOR [CTCP]. XOR and Reed-Solomon are supported in this document. Other network coding techniques, e.g., Random Linear Network Code (RLNC) [RLNC], Raptor Code [RC], etc., may be added in the future. A CCR message consists of the following fields:

- Flag (1 Byte):
  - 0: disable network-based network coding for a downlink flow
  - 1: enable network-based network coding for a downlink flow
  - 2: network coding configuration for a uplink flow
  - 3: network coding configuration for a downlink flow
  - Others: reserved
- Flow ID (1 Byte): an unsigned integer to identify the flow

The Flag field in a CCR message from a client is always set to "3".

If Flag == 1, 2 or 3, include the following fields:

- Coding Type (1 Byte)
  - 0: None
  - 1: XOR
  - 2: (Systematic) Reed-Solomon [RS]
  - Others: reserved
- N (1 Bytes): the number of consecutive (uncoded) GMA SDUs used to generate the coded GMA SDU

If Coding Type = (Systematic) Reed-Solomon, include the following:

- M (1 Byte): the number of coded (parity) GMA SDUs generated for every N consecutive uncoded GMA SDUs.
- L (1 Byte): the symbol size for the RS code finite field, i.e., the maximum codeword length (N + M) is given by 2^L-1.

If Coding Type == XOR, one coded GMA SDU will be generated by applying XOR across every N uncoded GMA SDU, and no additional parameter will be included in the CCR message.

5.16 Coded GMA SDU (CGS) Message

A CGS message is used to encapsulate a coded GMA SDU, generated by applying the specified network coding method to multiple consecutive

(uncoded) GMA SDUs. The Flow SN field (as shown in Figure 5) MUST NOT be included in the GMA header of a CGS message, as it carries a GMA data SDU. A CGS message SHOULD be encrypted only if the delivery connection is untrusted.

A CGS message consists of the following fields:

- o Flow ID (1 Byte): an unsigned integer to identify the flow.
- o Flag (1 Byte)
  - + Bit #0: to indicate if the CGS message uses the same coding configuration as its previous CGS message or not. This bit is flipped whenever a new configuration is used.
  - + Bit #1: to indicate if the FC field is present or not.
  - + Bit #2~7: reserved
- o Fragmentation Control (FC) (1 Byte): to provide necessary information for re-assembly.
  - + Bit #0: a More Fragment (MF) flag to indicate if the fragment is the last one (0) or not (1).
  - + Bit #1~#7: Fragment Offset (in units of fragments) to specify the offset of a particular fragment relative to the beginning of the SDU.
- o Flow SN (3 Bytes): the Flow SN of the first (uncoded) GMA SDU used to generate the coded GMA SDU, updated every N GMA SDUs
- o C-SN (1 Bytes): the sequence number (0 ~ M-1) of the coded GMA SDU carried by the CGS message, reset to "0" every N uncoded GMA SDUs.
- o Coded GMA SDU (variable): if the Coded GMA SDU is too long, it can be fragmented and transported by multiple CGS messages.

5.17 Connection Priority Reconfiguration (CPR) Message

The gateway MAY send out a CPR message to configure the priority of a delivery connection for a client or a flow. It consists of the following fields:

- o Client Connection Priority Bitmap (1 Byte): to indicate the default priority (0: low; 1: high) of the i-th delivery connection, where connections are ordered the same way as in Link Status Bitmap (5.1).
- o N1 (1 Byte): to indicate the number of downlink flows that are configured with a flow-specific connection priority bitmap.
- o N2 (1 Byte): to indicate the number of uplink flows that are configured with a flow-specific connection priority bitmap.

For each downlink flow, include the following fields:

- o Downlink Flow ID (1 Byte)

- o Flow Connection Priority Bitmap (1 Byte): to indicate the priority of the i-th delivery connection for the downlink flow.

For each uplink flow, include the following fields:

- o Uplink Flow ID (1 Byte)
- o Flow Connection Priority Bitmap (1 Byte): to indicate the priority of the i-th delivery connection for the uplink flow.

There are only two priority levels: high and low. Client SHOULD only use a low-priority connection for its data traffic if all high-priority connections are disconnected or disabled. The client SHOULD use the Client Connection Priority Bitmap (CCPB) for a flow if the flow is not configured with a Flow Connection Priority Bitmap (FCPB).

6 Basic GMA Control Procedures

GMA control sequence consists of the following three phases:

- o Phase 1 (Initialization): client and gateway exchange MAMS messages [MAMS] to configure the GMA-based multi-access traffic management.
- o Phase 2 (GMA Operation): client and gateway exchange GMA control messages as defined in this document to manage traffic steering/splitting/duplicating across multiple connections.
- o Phase 3 (Termination): client and gateway exchange MAMS messages to terminate the GMA operation.

6.1 Initialization

A GMA client may trigger the initialization procedure once detecting any one of the delivery connections, e.g. Wi-Fi, LTE, etc., becomes available. Figure 12 shows the MAMS message exchange sequence to activate the GMA operation. Please refer to [MAMS] for more details about MAMS messages.

```
      Client                                     multi-access Gateway
       |                                                    |
       |------- MX Discover Message ----------------------->|
       |                                                    |
       |<----------------------------- MX System Info ------|
       |                                                    |
       |------------------------------ MX Capability REQ -->|
       |<----- MX Capability RSP ---------------------------|
       |------------------------------- MX Capability ACK -->|
       |                                                     |
       |<-------------------- MX UP Setup Config -----------|
       |-------- MX UP Setup Confirmation ----------------->|
       |                                                     |
```

Figure 12: MAMS-based Initialization Procedure

To support the virtual (anchor) connection specified in this document, the MX Capability REQ message SHOULD include the following additional information:

- Last IP address: the virtual IP address used in the last MAMS session
- Last MAMS session ID: the unique session id of the last MAMS session

Moreover, the MX Capability REQ/RSP message SHOULD indicate the following GMA capabilities for downlink and uplink, respectively:

- Maximum number of flows with the redundancy mode
- Maximum number of flows with the splitting mode
- Maximum number of flows with the steering mode
- Network-based traffic steering (7.1)
- QoS-aware traffic steering (7.2)
- GMA-based retransmission (7.3)
- Network coding (7.4)
- Dynamic Connection Management (7.5)
- Dynamic OWD Equalization (7.6)
- Network coding method (XOR or Reed-Solomon)

The MX UP Setup Config message SHOULD include the following additional information:

- Client ID: see Figure 5
- Client IP address: the client IP address of the virtual anchor connection.
- Gateway IP address: the gateway IP address the virtual anchor connection
- DNS server: the DNS server IP address of the virtual anchor connection
- Subnet mask: the subnet mask of the virtual anchor connection
- MAMS port: the TCP port number at the multi-access Gateway for exchange MAMS messages over the virtual anchor connection
- Key: the symmetric encryption (e.g. AES256-GCM) key to protect GMA payload

- o Untrusted CID List: the list of "untrusted" delivery connections where GMA data SDU MUST be protected by the symmetric encryption key.
- o Best-Effort CID List: the list of "best-effort" delivery connections where QoS-aware traffic steering procedure (7.2) SHOULD be used for moving a QoS flow to the connection.

6.2 GMA Operation

After completing the initialization phase successfully, a client will start the GMA operation phase by sending out probes to decide if a delivery connection can be used for data transfer.

After successful probing, client will activate the virtual anchor connection based on the information in the MX UP Setup Config message and start (GMA-based) multi-access traffic management.

If the client is already synchronized with the gateway in time, it will use its local clock as the timestamp clock. Otherwise, the client will perform the timestamp synchronization procedure by sending out the Probe-Sync message. Afterwards, the client SHOULD send out the Probe-Sync message once a while to reset the timestamp clock.

During the GMA operation, the client SHOULD continuously perform path quality measurements (e.g. one-way delay, loss, etc.) based on probing as well as received data packets, and manage traffic across all available connections accordingly. How and when to trigger probing as well as how to perform path quality measurements are left to implementation. Moreover, it is up to implementation which delivery connection is used to send control messages, e.g. TSU, etc. However, the ACK message SHOULD use the same delivery connection as its corresponding control message.

For a downlink flow, if network-based traffic steering (7.1) is enabled, the gateway SHOULD control how to steer or split the flow through the TSC or TSP message; otherwise, the client SHOULD control it through the TSU message.

For an uplink flow using the steering mode, if network-based traffic steering (7.1) is enabled, the gateway SHOULD control how to steer traffic through the TSC message (5.7); otherwise, the client SHOULD control it without any control message.

For an uplink flow using the splitting mode, the gateway SHOULD control it through the TSU message.

```
     Client                                     multi-access Gateway
       |                                                    |
  +--------------+                                          |
  | Link x is up |                                          |
  +--------------+                                          |
       |---------------------------- Probe (over Link x) -->|
       |<----- ACK (over Link x) ---------------------------|
       |                                                    |
  +----------------------------------------+                |
  | activate the virtual anchor connection |                |
  | and start the GMA operation            |                |
  +----------------------------------------+                |
       |                                                    |
       |                                                    |
       |---------------------------- Probe-Sync ----------->|
       |                                   +----------------+
       |                                   |reset timestamp |
       |                                   +----------------+
       |<----- ACK -----------------------------------------|
  +---------------+                                         |
  |reset timestamp|                                         |
  +---------------+                                         |
       |                                                    |
  +----------------------------------------+                |
  | perform path quality measurement based |                |
  | on probes and data packets, and decide |                |
  | to steer traffic over Link x           |                |
  +----------------------------------------+                |
       |------------------------------ TSU (over Link x)--->|
       |<----- TSA (over Link x)----------------------------|
```

Figure 13: GMA-based Multi-Access Traffic Management Procedure

6.3 Termination

A client may trigger the termination procedure to stop the GMA operation at any time. Figure 14 shows the MAMS message exchange sequence to terminate the GMA operation.

```
    Client                                     multi-access Gateway
       |                                                    |
       |------- MX Termination Request--------------------->|
       |                                                    |
       |<------------------------ MX Termination Response---|
```

Figure 14: MAMS-based Termination Procedure

7 Advanced GMA Control Procedure

7.1 Network-based Traffic Steering (Splitting)

Figure 15 and 16 show the network-based traffic steering and splitting procedure, respectively. It is initiated by the gateway sending out the TSC (5.7) or TSP (5.8) message.

If the Flag field in the TSC (TSP) message is set to "0", network-based control is disabled and the client SHOULD decide how to steer (split) a flow based on its local information.

If the Flag field in the TSC (TSP) message is set to "1", network-based control is enabled and the traffic steering (splitting) configuration in the TSC (TSP) message SHOULD be used.

For a downlink flow, the client SHOULD send out a TSU message to confirm the updated traffic steering (splitting) configuration.

For an uplink flow, the gateway SHOULD use the TSU message to update the traffic steering (splitting) configuration after enabling network-based control with the TSC (TSP) message.

```
      Client                                     multi-access Gateway

       |                                                    |
       |<---------- (downlink) flow #1 over link x ---------|
       |                +-------------------------------------------+
       |                |collect measurement from network and decide|
       |                |to steer traffic over link y               |
       |                +-------------------------------------------+
       |<---------------- TSC (steer flow #1 to link y)-----|
       |------- ACK --------------------------------------->|
       |                                                    |
       |------- TSU --------------------------------------->|
       |<-----------------------------------------TSA ------|
       |                                                    |
       |<-------------- flow #1 downlink over link y -------|
       |                                                    |
```

Figure 15: Network-based Downlink Traffic Steering Procedure

```
    Client                                   multi-access Gateway
       |                                                    |
       |<---------- (downlink) flow #1 over link x & y------|
       |                +-------------------------------------------+
       |                |collect measurement from network and decide|
       |                |to update traffic splitting ratio          |
       |                +-------------------------------------------+
       |<-------TSP (updated splitting ratio for flow #1)---|
       |------- ACK --------------------------------------->|
       |                                                    |
       |------- TSU --------------------------------------->|
       |<-----------------------------------------TSA ------|
       |                                                    |
       |<--------- flow #1 (updated splitting ratio)--------|
       |                                                    |
```

Figure 16: Network-based Downlink Traffic Splitting Procedure

7.2 QoS-aware Traffic Steering

Figure 17 shows the QoS-aware traffic steering procedure for steering a flow with QoS requirements, e.g. maximum delay, maximum loss rate, etc., to a best-effort connection, e.g. Wi-Fi.

At the very beginning, a flow (e.g. uplink flow #1) is delivered over the connection (e.g. Cellular) that can guarantee its QoS requirements. Once a best-effort connection (e.g. Wi-Fi) becomes available, the client SHOULD send out the QTR message to request QoS testing for the flow. In response, the gateway SHOULD decide when to start the testing and send out the QTN message. If the Network-based Traffic Steering (7.1) is enabled for the flow, the gateway will initiate the procedure by sending out the unsolicited QTN message directly.

During the QoS testing, the transmitting endpoint SHOULD duplicate the flow over the testing connection. All duplicated packets SHOULD be discarded by the receiving endpoint and used only for testing. In the meantime, they SHOULD be marked with low priority to minimize their impact to other flows that have already been steered to the connection.

If the QoS testing fails, the receiving endpoint of the QoS testing will send out a QTP message to notify the transmitter of the testing result. Otherwise, it will send out a TSU message to steer the flow to the best-effort connection. Afterwards, it will continue monitoring the QoS performance of the flow. Once detecting any QoS violation, it will send out a TSU message to steer the flow back to the QoS-guaranteed connection.

If network-based traffic steering is enabled, the gateway MAY steer a QoS flow from a best-effort connection back to the QoS-guaranteed connection anytime. However, when the gateway decides to steer a QoS flow to a best-effort connection, it SHOULD first send out the QTN message to initiate QoS testing and steer the flow only if the QoS testing succeeds.

```
     Client                                               Gateway
       |                                                    |
       |-------- (uplink) flow #1 over link x ------------->|
  +--------------+                                          |
  | link y is up |                                          |
  +--------------+                                          |
       |                                                    |
  +-----skipped if network-based traffic steering is enabled -------+
  |    |------- QTR (req testing flow #1 over link y)------>|        |
  |    |<------------------- ACK ---------------------------|        |
  +------------------------------------------------------------------+
       |                                                    |
       |<------ QTN (start testing flow #1 over link y)-----|
       |--------------------- ACK ------------------------->|
       |                                                    |
       |----duplicating flow #1 over link y --------------->|
       |                                                    |
       |                 +-------------------------------------------+
       |                 |collect measurement and drop all duplicated|
       |                 |packets received from link y               |
       |                 +-------------------------------------------+
       |                                                    |
       |<-------------------- TSU --------------------------|
       |--------------------- ACK ------------------------->|
       |                                                    |
       |-------- flow #1 over link y ---------------------->|
       |                                                    |
       |                +--------------------------------------------+
       |                |collect measurement and detect QoS violation|
       |                +--------------------------------------------+
       |<-------------------- TSU -------------------------|
       |--------------------- ACK ------------------------>|
       |                                                   |
       |-------- flow #1 over link x --------------------->|
       |                                                   |
```

Figure 17: QoS-aware Traffic Steering Procedure

It is pretty straightforward to measure packet loss rate using the Flow SN field in the GMA header and detect QoS violation accordingly. Next, we will introduce a simple method to detect QoS violation based on OWD measurement.

Define the following notations:

- d0: the (true) OWD of a received data packet
- d1: the OWD measurement of a received data packet
- d2: the OWD measurement of the received ACK message for the last probe
- d3: the (true) OWD of the probe message on the reverse path
- r: the round trip time (RTT) measurement of the last probe
- D: the maximum OWD threshold for QoS violation detection.

If client and gateway are not synchronized in time, we can't measure OWD directly. Moreover, we can't measure RTT of a data packet either because the data packet does not have acknowledgement. Thus, we propose to use the RTT measurement of the last probe as the reference to estimate the RTT of a received data packet, and use it as the OWD upper-bound, i.e.,

$$d0 = r - d3 + d1 - d2 < r + d1 - d2$$

Then, we detect OWD QoS violation as follows:

$$r + d1 - d2 > D$$

We MAY send the flow-specific probe message with high priority to reduce d3 and minimize its impact.

Notice that the above QoS-aware traffic steering procedure SHOULD be used only if the QoS requirement of a flow can be guaranteed by at least one delivery connection. Otherwise, the flow SHOULD be configured with the redundancy mode, and the GMA receiver SHOULD measure and detect QoS violation based on data packets received from each delivery connection and determine which delivery connections will be used to send duplicated packets of the flow. At the very beginning, a flow MAY be duplicated over all the available connections. Afterwards, if a connection is found sufficient for meeting the QoS requirement by itself, the GMA receiver MAY steer the flow to the single connection and stop duplication to improve efficiency. If detecting any QoS violation, it will reconfigure the flow to start duplicating over multiple connections again. The TSU message (5.3) with the traffic steering configuration IE (Type = 0) SHOULD be used for reconfiguration.

7.3 GMA-based Retransmission

Figure 7 shows the GMA-based retransmission procedure in an example. The first lost packet is found and retransmitted. However, the second lost packet is not found, and the FSN message is sent out to notify the client.

```
          Client                                        Gateway
              |                                            |
              |<------------------ GMA SDU (data packets)--|
              |                                            |
  +---------------------+                                  |
  |Packet Loss detected |                                  |
  +---------------------+                                  |
              |                                            |
              |----- PLR Message ------------------------->|
              |                              +---------------------+
              |                              |Lost packet found    |
              |                              +---------------------+
              |<-------------retransmit(lost)MX SDUs ------|
              |<------------------ GMA SDU (data packets)--|
              |                                            |
  +----------------------+                                 |
  |Packet Loss detected  |                                 |
  +----------------------+                                 |
              |                                            |
              |----- PLR Message ------------------------->|
              |                              +---------------------+
              |                              |Lost packet not found|
              |                              +---------------------+
              |<-------------FSN message ------------------|
```

Figure 18: GMA-based Retransmission Procedure

7.4 Network Coding

Network coding for an uplink flow is always configured by the gateway using the CCR message with the Flag field set to "2".

For a downlink flow, it may be configured by gateway (network-based) or client. Figure 19 shows the client-based procedure, where the client detects packet loss and sends out a CCR message with the Flag field set to "3" to activate network coding along with all the required parameters. In this example, XOR is configured as the coding method with N = 2. In response, gateway starts sending one CGS message carrying the coded GMA SDU for every two (uncoded) GMA SDUs. Afterwards, client MAY send out a CCR message to deactivate network coding for the flow.

Figure 20 shows the network-based procedure. Wherein, the gateway will send out a CCR message with the Flag field set to "1" to provide all the configuration parameters. Notice that network coding MAY be used for a flow regardless of its operation modes: splitting, steering, or duplicating.

```
          Client                                          Gateway
              |                                            |
              |<------------------ GMA SDUs(data packets)--|
              |                                            |
  +----------------------+                                 |
  |Packet Loss detected |                                  |
  +----------------------+                                 |
              |                                            |
              |------CCR Message (Flag = 3, XOR, N=2)----->|
              |<------------------- ACK Message -----------|
              |                                            |
              |<------------------ GMA SDU #1 -------------|
              |      lost<-------- GMA SDU #2 -------------|
              |<-- CGS Message (GMA SDU #1 XOR GMA SDU #2)-|
  +----------------------+                                 |
  | GMA SDU #2 recovered |                                 |
  +----------------------+                                 |
              |                                            |
              |------CCR Message (None) ------------------>|
              |<------------------- ACK Message -----------|
              |<------------------ GMA SDU #3 -------------|
              |<------------------ GMA SDU #4 -------------|
              |<------------------ GMA SDU #5 -------------|
```

Figure 19: Client-based Network Coding Procedure for Downlink

```
          Client                                      Gateway
              |                                            |
              |<------------------ GMA SDUs(data packets)--|
              |                                            |
              |       +-------------------------------------------+
              |       |collect measurement from network and decide|
              |       |to activate network coding                 |
              |       +-------------------------------------------+
              |                                            |
              |<-----CCR Message (Flag = 1, XOR, N=2)------|
              |-------------------- ACK Message ---------->|
```

Figure 20: Network-based Network Coding Procedure for Downlink

7.5 Dynamic Connection Management

The gateway MAY use a DCR message (5.5) to disable or enable one or multiple delivery connections. For example, if the gateway detects or predicts significant performance degradation of a network, it may proactively disable the connection for clients that are connected to the network. The gateway gains knowledge of which network a client is connected to via Link IEs in a Probe message.

On the other hand, the gateway MAY use a CPR message (5.17) to provide guidance for a client to steer traffic especially when network-based traffic steering (7.1) is disabled or a flow is configured with the redundancy mode. The key difference between DCR and CPR are:

o A DCR message provide the configuration for all traffic while a CPR message applies only to data traffic and may provide a flow-specific configuration.

o A disabled connection (by DCR) MUST NOT be used for any traffic until it is enabled by another DCR message. Unlikely, a low priority connection (by CPR) MAY be used for data traffic if all high priority connections are lost.

For example, if a client has 4 concurrent delivery connections, the gateway may configure two of them as high priority and the other two as low priority for its flow with the redundancy mode. As a result, the flow SHOULD be duplicated over the two high priority connections when they are available. Only if both high priority connections are lost, the flow will be duplicated over the two low priority connections. If only one high-priority connection remains, the flow will be sent over the remaining high-priority connection until the gateway configures more connections as high priority.

When network-based traffic steering is disabled, a client will decide how to steer a flow over all available connections. However, the gateway may detect or predict that a network is experiencing performance degradation so that the QoS requirements of a flow can't be met. In this scenario, the gateway MAY use CPR to prevent a client from using "bad" connections for the flow.

Figure 21 shows an example of CPR-based dynamic connection management. At the very beginning, a flow is split over two connections: x and y. Once the gateway detects performance degradation of x, it sends out a CPR message with x set to low priority and y set to high priority. In response, the client will steer the flow to y. Afterwards, y is lost and the client steers the flow to x.

```
       Client                                                 Gateway
       |                                                    |
       |<--------- flow #1 split over x & y --------------->|
       |            +-------------------------------------------+
       |            |collect measurement from network and detect|
       |            |performance degradation of x              |
       |            +-------------------------------------------+
       |<------- CPR (x: low priority and y: high-priority)-|
       |--------------------- ACK ------------------------->|
       |------- TSU (steer flow #1 to y)------------------->|
       |<--------------------- ACK -------------------------|
       |<--------- flow #1 over y ------------------------->|
       |                                                    |
  +-----------+                                             |
  | y is lost |                                             |
  +-----------+                                             |
       |------- TSU (steer flow #1 to x)------------------->|
       |<-------------------- ACK --------------------------|
       |<--------- flow #1 over x ------------------------->|
       |                                                    |
```

Figure 21: CPR-based Dynamic Connection Management Example

7.6 Dynamic One-Way-Delay (OWD) Equalization

A GMA transmitting endpoint MAY add an extra "delay" to each of the packets before sending it to a delivery connection for mitigating the impact of reordering due to OWD difference among the delivery connections, aka "Delay Equalization" in [MPSN].

When enabled, the GMA receiving endpoint SHOULD measure and report minimum OWD measurement based on data packets of the flow periodically, e.g., every 12 seconds, or immediately when receiving a data packet of the flow with its OWD (d) meeting the following criteria:

$$d < t - c$$

Where t is the last minimum OWD estimation and c is a configurable constant (margin), e.g. 10ms.

In response, the GMA transmitter SHOULD update the delay (T(i)) added to the i-th connection using the following two-step procedure:

```
o step 1: T(i) = T(i) + D(i), where i = 1 ~ N
o step 2: T(i) = T(i) - min(T(i) | i = 1 ~ N)
```

Wherein, step 1 is for equalizing the minimum OWD for all the connections, and step 2 is for minimizing the delay added to each connection. The GMA transmitter MAY apply the above procedure only to the connections that are actively being used to send data packets of a flow and set T(i) = 0 for others. In this case, N indicates the total number of active connections. Moreover, the GMA receiver MAY request to reset the delay for a connection by setting its D(i) to "255" in a TSU message. In response, the GMA transmitter SHOULD simply set T(i)=0 for the i-th connection, and exclude it from the procedure above.

Moreover, the GMA transmitting endpoint SHOULD use the OWD adjustment configuration IE in the TSA message (5.4) to indicate how much one-way delay has been added or reduced for a connection following the above procedure. In response, the GMA receiving endpoint SHOULD adjust its minimum OWD estimation accordingly, i.e. t'=t + q, where the notations are defined as follows:

- t: the minimum OWD estimation before receiving the TSA message
- q: the OWD adjustment in the TSA message
- t': the minimum OWD estimation after receiving the TSA message

8 Security Considerations

A method is provided to protect GMA control messages with a symmetric key (e.g. AES256). It can also be used to protect GMA data packets if a delivery connection is "untrusted".

9 IANA Considerations

This document makes no requests of IANA.

10 Contributing Authors

The editors gratefully acknowledge the following additional contributors in alphabetical order: Wei Mao/Intel, Hosein Nikopour/Intel.

11 References

11.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174 May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[GRE1] Dommety, G., "Key and Sequence Number Extensions to GRE", <https://www.rfc-editor.org/info/rfc2890>.

[QUIC] RFC 9000, "QUIC: A UDP-Based Mutiplexed and Secure Transport", <https://www.rfc-editor.org/rfc/rfc9000.txt>

11.2 Informative References

[MAMS] RFC 8743, "Multi-Access Management Services (MAMS)" <https://tools.ietf.org/rfc/rfc8743.txt>

[LWIPEP] 3GPP TS 36.361, "Evolved Universal Terrestrial Radio Access (E-UTRA); LTE-WLAN Radio Level Integration Using Ipsec Tunnel (LWIP) encapsulation; Protocol specification"

[ATSSS] 3GPP TR 23.793, Study on access traffic steering, switch and splitting support in the 5G system architecture.

[GRE2] RFC 8157, Huawei's GRE Tunnel Bonding Protocol, May 2017

[ATSSS2] M. Boucadair, et al. 3GPP Access Traffic Steering Switching and Splitting (ATSSS) - Overview for IETF Participants, < https://datatracker.ietf.org/doc/html/draft-bonaventure-quic-atsss-overview-00>

[GMAE] J. Zhu, et al. RFC 9188 Generic Multi-Access (GMA) Encapsulation Protocol <https://www.rfc-editor.org/rfc/rfc9188.txt>

[GCC] S. Holmer, et al. A Google Congestion Control Algorithm for Real-Time Communication, https://www.ietf.org/archive/id/draft-ietf-rmcat-gcc-02.txt

[MPIP] L. Sun, et al. Multipath IP Routing on End Devices: Motivation, Design, and Performance,

[QUICTLS] M. Thomson and S. Turner, Using TLS to Secure QUIC, https://www.rfc-editor.org/rfc/rfc9001.txt

[GMA] https://github.com/IntelLabs/gma

[CTCP] Simone Ferlin, et al., MPTCP meets FEC: Supporting Latency-Sensitive Applications over Heterogeneous Networks, IEEE Transactions on Networking, Oct 2018

[RLNC] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi and B. Leong, "A random linear network coding approach to multicast," IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4413-4430, 2006.

[RC] A. Shokrollahi, "Raptor codes," in IEEE Transactions on Information Theory, vol. 52, no. 6, pp. 2551-2567, June 2006, doi: 10.1109/TIT.2006.874390.

[RS] I. Reed and G. Solomon, "Polynomial codes over certain finite fields," Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 2, pp. 300-304, 1960.

[DCTCP] RFC 8257, "Data Center TCP (DCTCP): TCP Congestion Control for Data Centers", <https://datatracker.ietf.org/doc/html/rfc8257>

[Dual3GPP] 3GPP TR 22.841, "Study on Upper Layer Traffic Steer, Switch and Split over Dual 3GPP Access", 2023-12

[MPSN] M. Amend, D. Von Hugo, Multipath Sequence Maintenance, <https://www.ietf.org/archive/id/draft-amend-iccrg-multipath-reordering-03.txt>

Authors' Addresses

Jing Zhu

Intel

Email: jing.z.zhu@intel.com

Menglei Zhang

Intel

Email: menglei.zhang@intel.com