Network Working Group
Internet Draft
Intended status: Experimental
Expires: October 11,2022

J. Zhu
M. Zhang
Intel

April 11, 2022

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

draft-zhu-intarea-gma-control-01

Abstract

A device can be simultaneously connected to multiple networks, e.g., Wi-Fi, LTE, 5G, and DSL. It is desirable to seamlessly combine the connectivity 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 control protocol to manage traffic steering, splitting, and duplicating across multiple connections. The solution has been developed by the authors based on their experiences in multiple standards bodies including the 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 in order to experiment with the protocol.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on October 11, 2022.

Copyright Notice

Table of Contents

1. Introduction

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

Figure 1 shows the Multi-Access Management Service (MAMS) user-plane protocol stack [MAMS], which has been used in today's multi-access solutions [ATSSS] [LWIPEP] [GRE1] [GRE2]. It 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 in the protocol stack. 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.

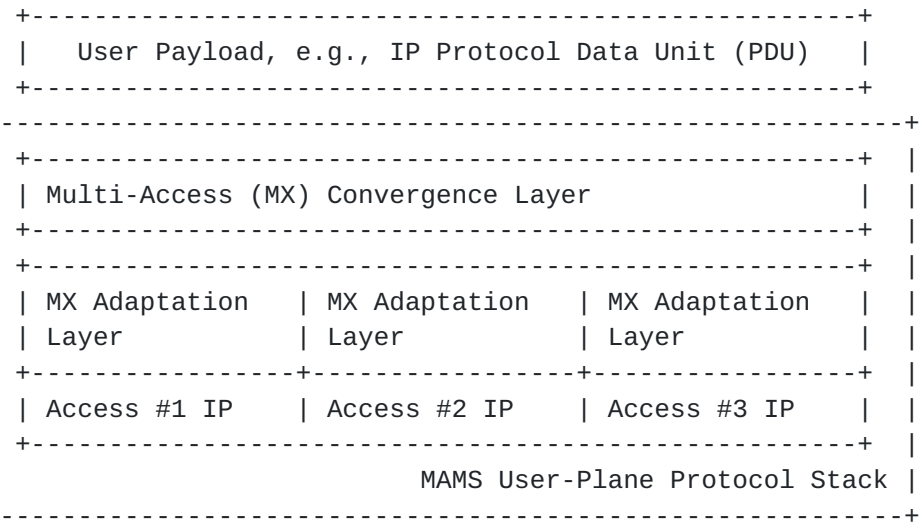


Figure 1: MAMS User-Plane Protocol Stack [MAMS]

A new encapsulation protocol [GMAE] has been specified for the convergence layer to encode additional control information, e.g., Timestamp, Sequence Number, required for multi-access traffic management. This document presents a UDP-based GMA control protocol for the convergence layer. The GMA control protocol only operates between endpoints that have been configured to use GMA. This configuration can be through any management messages and procedures, including, for example, Multi-Access Management Services [MAMS].

The solution described in this document was been developed by the authors based on their experiences in multiple standards 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 ATSSS Phase 2 [ATSSS2] requirements, can be safely used, and has support for deployment. Particularly, the proposed GMA protocol addresses the following issues of using QUIC for ATSSS Phase 2:

- 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 overhead is 16 bytes.
- Multiple Encryptions: the GMA encapsulation protocol does not require encryption and avoids redundant encryption overhead.
- Congestion Control in Congestion Control: the GMA control protocol does not require congestion control. All incoming packets (from higher layer) are sent over one of the delivery connections immediately without any delay due to congestion control.

In addition, the GMA protocol does not require Acknowledgement (ACK) and reliable delivery for data-plane traffic to avoid any delay due to retransmission as well as any ACK traffic overhead on the reverse path.

Path quality measurements (e.g. one-way-delay, loss, etc.) and congestion detection are performed by receiver based on the GMA header fields, e.g. sequence number, timestamp, etc. Another objective of the experiment is to evaluate the usage of various receiver-based congestion detection algorithms [GCC] [MPIP] in multi-access traffic management.

It is expected that this protocol experiment can be conducted on the Internet since the GMA packets are encapsulated with UDP. Thus, experimentation is conducted between consenting end systems that have been mutually configured to participate in the experiment.

The authors will continually assess the progress of this experiment and encourage other implementers to contact them to

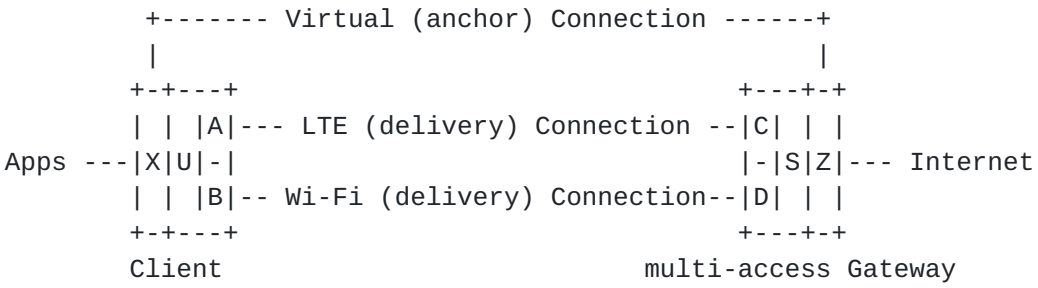
report the status of their implementations and their experiences with the protocol.

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 the 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 connection provides multiple paths between client and multi-access gateway for multi-access traffic management, aka Access Traffic Steering, Switching, and Splitting (ATSSS) in 3GPP [ATSSS].



- 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 the two 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 optimization requires additional control message exchange between client and multi-access gateway.

"UDP" is used for the adaptation layer in this document. Figure 3a and 3b show the UDP-based GMA user-plane and control-plane protocol, respectively.

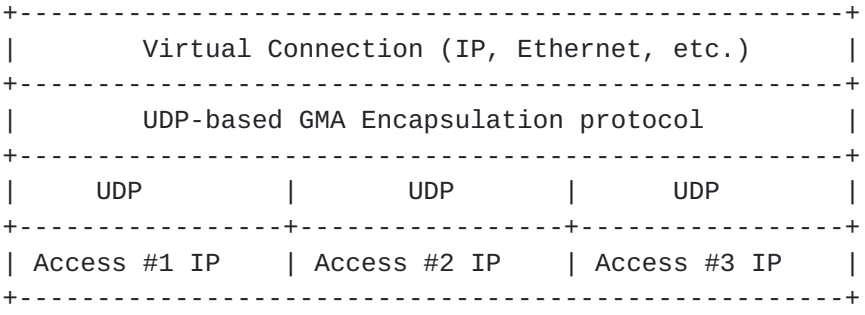


Figure 3a: UDP-based GMA User-Plane Protocol Stack

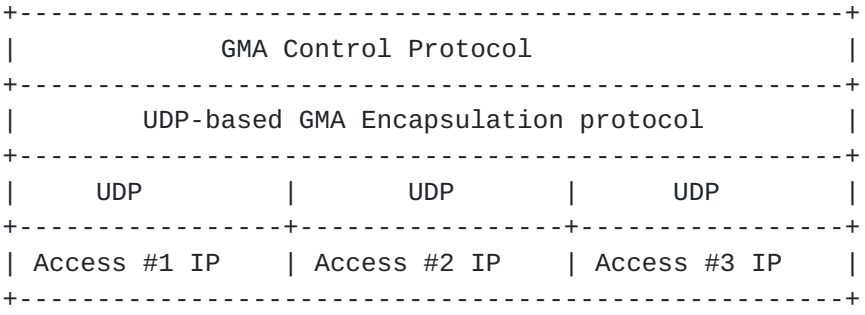


Figure 3b: UDP-based GMA Control-Plane Protocol Stack

4. UDP-based GMA Encapsulation Protocol

Figure 4 shows the UDP-based GMA encapsulation format as specified in [GMAE]. The ports for "UDP Tunnelling" at Client are chosen from the Dynamic Port range, and the ports for "UDP Tunnelling" at

multi-access gateway are configured and provided to client through the MAMS message (MX UP Setup Config) [MAMS].

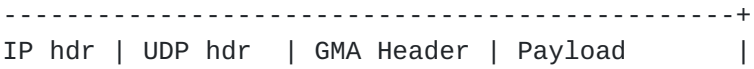


Figure 4: UDP-based GMA PDU Format

The GMA (Generic Multi-Access) header MUST consist of the mandatory "Flags" field (the first two bytes), defined as follows:

- o Client ID Present (bit 0): If the Client ID Present bit is set to 1, then the Client ID field is present
- o Flow ID Present (bit 1): If the Flow ID Present bit is set to 1, then the Flow ID field is present
- o Per-Packet Priority (PPP) Present (bit 2): If the PPP Present bit is set to 1, then the PPP field is present
- o SN Present (bit 3): If the SN (Sequence Number) Present bit is set to 1, then the Delivery and Flow SN fields are present and contains the valid information
- o Timestamp Present (bit 4): If the Timestamp Present bit is set to 1, then the Timestamp field is present
- o Reserved (bit 5-15): 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 each optional field accordingly. The GMA (Generic Multi-Access) header MAY consist of the following optional fields:

- o Client ID (2 Byte): an unsigned integer to identify the client
- o Flow ID (1 Byte): an unsigned integer to identify the IP flow of the GMA SDU.
- 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).
- o Delivery SN (1 Byte): an auto-incremented integer to indicate the GMA PDU transmission order on a delivery connection. Delivery SN is used to measure packet loss of each delivery connection and therefore generated per delivery connection per flow. This field is present only if the Delivery SN Present bit is set to one.
- o Flow SN (3 Bytes): an auto-incremented integer to indicate the GMA SDU (IP packet) order of a flow. Flow SN is used for

reordering, and therefore generated per flow. This field is present only if the Flow SN Present bit is set to one.

- o Timestamp (4 Bytes): to contain the current value of the timestamp clock of the transmitter in the unit of 1 millisecond. This field is present only if the Timestamp Present bit is set to one.

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 order to their corresponding flag bits. If a flag bit is clear, the corresponding optional field is absent.

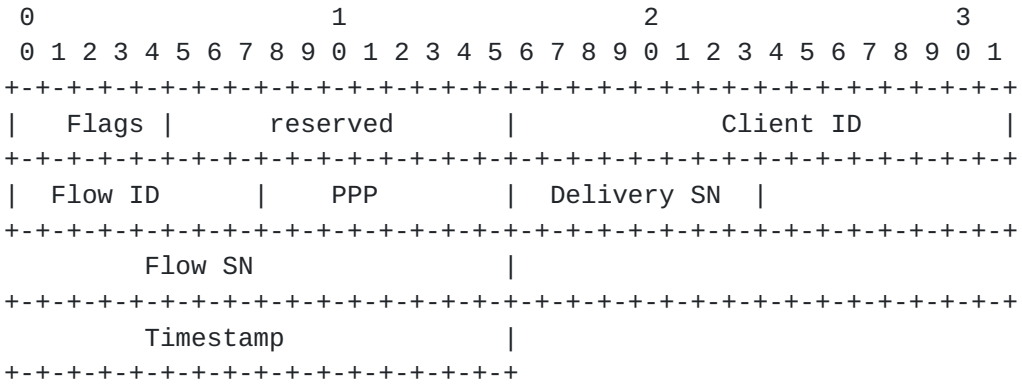


Figure 5: GMA Header Format with all Optional Fields Present

Some GMA header fields, e.g. Client ID, Flow ID, and PPP are designed to support hierarchical QoS (hQoS) and fine granular packet classification. Notice that GMA header fields (unlike IP header field) won't change regardless how a GMA PDU is delivered on the way, since they are encapsulated as part of UDP payload. Therefore, an intermediate node, e.g. router, Access Point, Base Station, etc., can perform hQoS scheduling and active queue management (AQM) directly based on these GMA header fields without additional packet classification processing.

Other GMA header fields, e.g. Delivery SN, Flow SN, and Timestamp, are designed to support multi-access 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, and Timestamp allows one-way-delay measurement, which can then be used to detect congestion and buffer overflow at intermediate nodes.

5. GMA Control Messages

A GMA control message is encapsulated as the payload of a GMA PDU (see Figure 4) and the GMA header MUST include the Client ID field, but not any other optional fields. As a result, the Flag in the GMA header is always set to 0x8000 for a GMA control message.

GMA control message MAY be encrypted with a symmetric key cipher, e.g. AES256-GCM. If a GMA control message is encrypted, the receiver will use the Client ID field to obtain the corresponding key for decryption. Notice that only the GMA control message is encrypted. The GMA header is authenticated but not encrypted.

Figure 6 shows the format of an encrypted GMA control message, where IV (initialization vector) is 12 bytes long and GCM Tag is 16 bytes long. The GMA header (Flag (2B) + Client ID (2B)) is used as additional authenticated data (AAD).

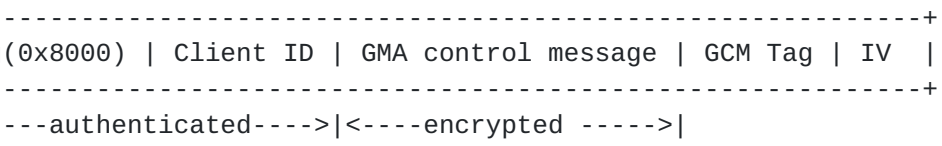


Figure 6: Encrypted GMA Control Message

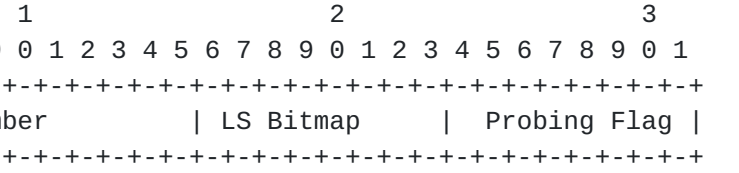
A GMA control message consists of the following fields:

- Header (2 Bytes)
  - Type (1 Byte): the GMA control message type
  - Connection ID (1 Byte): an unsigned integer to identify the anchor connection for the GMA control message
- Payload (variable): the payload of the GMA control message

5.1 Probe Message

The "Type" field is set to "1" for Probe messages.

Client (or multi-access gateway) may send out Probe message for path quality estimation or keepalive. In response, multi-access gateway (or client) may send back the ACK message.



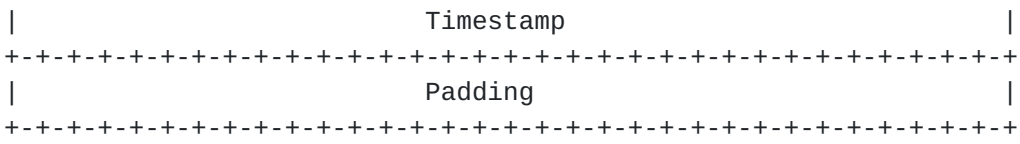


Figure 7: Probe Message Format

A Probe message consists of the following fields:

- o Sequence Number (2 Bytes): the sequence number of the message
- o Link Status (LS) Bitmap (1 Bytes): to indicate the status (0: not connected; 1: connected) of the i-th delivery connection, where connections are ordered according to their Connection ID, 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 the ACK message is expected (0) or not (1)
  - + Bit #1: a bit flag to indicate if multi-access Gateway SHOULD update the UDP tunnel end-point (0) or not (1) based on the received Probe message.
  - + Bit #2~7: reserved
- o Timestamp (4 Bytes): the current value of the timestamp clock of the sender
- o Padding (variable)

The "Padding" field is used to control the length of a Probe message.

Multi-Access Gateway SHOULD update the UDP tunnel end-point of the client based on the received Probe message if the Bit #1 Probing flag is set to 0 (default).

5.2 Acknowledgement (ACK) Message

The "Type" field is set to "2" for ACK messages. The ACK message consists of the following fields:

- o Acknowledgment Number (2 Bytes): the sequence number of the corresponding request message
- o Reserved (1 Byte)
- o Request Type (1 Byte): the corresponding request message type, e.g. Probe, etc.
- o Timestamp (4 Bytes): the current value of the timestamp clock of the sender

Figure 8: Ack Message Format

5.3 Traffic Splitting Update (TSU) Message

The "Type" field is set to "3" for TSU messages.

Client (or multi-access gateway) may send out a TSU message to change the traffic splitting/steering/duplicating configuration for downlink flows. Let's use N to denote the number of delivery connections.

A TSU message consists of the following fields:

- o Sequence Number (2 Bytes): the sequence number of the TSU message
- o Link Status 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 Connection ID
- o Number of Flows (1 Byte): the number of flows that are configured by the TSU message
- o Timestamp (4 Bytes): the current value of the timestamp clock of the sender

For each flow, the following Traffic Splitting control parameters are included:

- o Flow ID (1 Byte): an unsigned integer to identify the flow
- o L (1 Bytes): 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 K1[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 K2[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 N = 2, i.e. two delivery connections, the configuration of K1[1] = K1[2] = 0, K2[1] = K2[2] = 1, and L = 2 indicates sending every packet of the flow over both connections, i.e. duplication. In another example, the configuration of K1[1] = K2[1] = 0, K1[2] = K2[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.

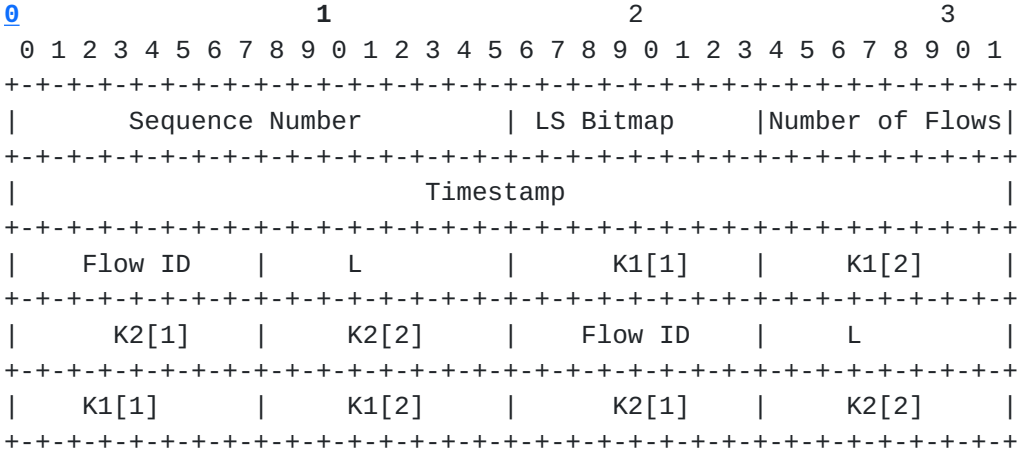


Figure 9: TSU Message Format (N = 2, Number of Flows = 2)

Multi-access gateway SHALL always update the UDP tunnel end-point of the client based on the received TSU message.

5.4 Traffic Splitting Acknowledgement (TSA) Message

The "Type" field is set to "4" for TSA messages. Multi-access gateway (or client) SHALL send out a TSA message in response to a received TSU message. A TSA message consists of the following fields:

- Acknowledgment Number (2 Bytes): the sequence number of the corresponding TSU message
- Reserved (1 Byte)
- Number of Flows (1 Byte): the number of flows that are configured by the TSU message
- Timestamp (4 Bytes): the current value of the timestamp clock of the sender in the unit of 1 millisecond

For each flow, the message further consists of the following fields:

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

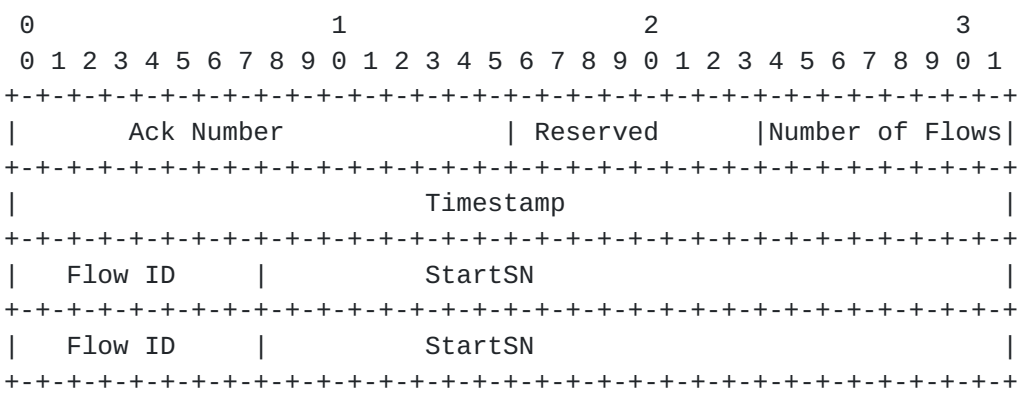


Figure 10: TSA Message Format (Number of Flows = 2)

Figure 11 shows the traffic splitting update procedure for downlink traffic, where client performs path quality measurement based on received packets and determines traffic splitting parameters. Once update is needed, client will send the TSU message carrying the new traffic splitting parameters to multi-access gateway. Multi-access gateway will send back the TSA message in response, and perform traffic splitting accordingly. The TSA message carries the "StartSN" parameter to indicate the first packet using the new configuration so that client can perform measurements accordingly.

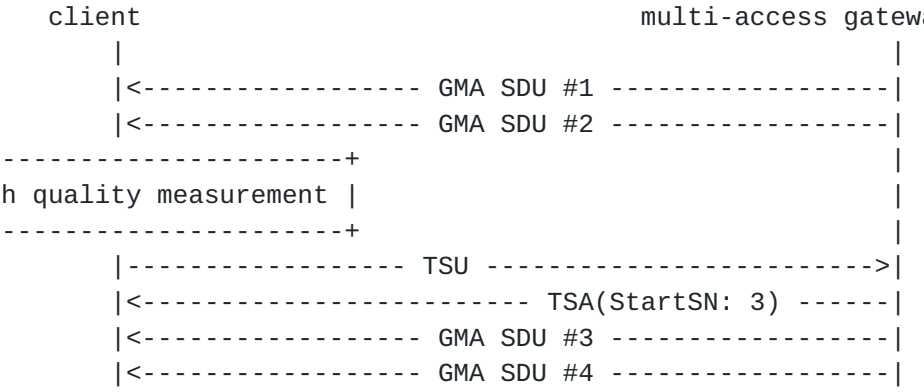


Figure 11: Downlink Traffic Splitting Update Procedure

5.5 Timestamp Reset Request (TSR) Message

The "Type" field is set to "5" for TSR messages.

A TSR message consists of only one field:

- Sequence Number (2 Bytes): the sequence number of the TSR message.

Client SHOULD send out a TSR message to reset timestamp and prevent it from overflowing for one-way delay measurement due to the limited size (4 Bytes) when its local timestamp timer exceeds a pre-defined value, e.g. 0x7FFF0000.

Once receiving a TSR message, multi-access gateway SHOULD reset the timestamp timer to "0" for the client and respond with a ACK message. Client SHOULD reset its timestamp timer to "0" after the TSR message is successfully acknowledged. As a result, the timestamp field in a GMA PDU indicates the duration between the last successful TSR message exchange and the transmission of the GMA PDU.

6. GMA Control Flows

GMA control sequence consists of the following three phases:

- Phase 1 (Initialization): client and gateway exchange MAMS messages [MAMS] to configure the GMA-based multi-access traffic management.
- 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.
- Phase 3 (Termination): client and gateway exchange MAMS messages to terminate the GMA operation.

6.1. Initialization

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 the MAMS messages.

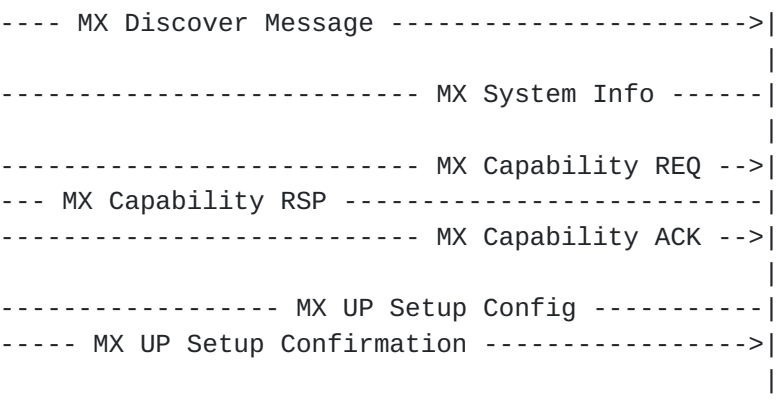


Figure 12: MAMS-based Initialization Procedure

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

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

- 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 for GMA control message.

6.2. GMA Operation

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

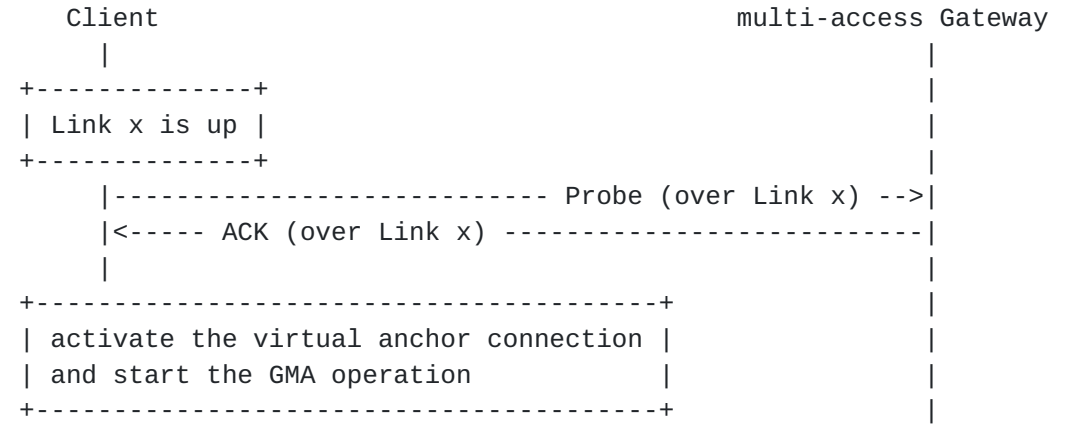
After successful probing, client will activate the virtual anchor connection based on the information in the MX UP Setup Config message and start (GMA-based) multi-access traffic management.

First of all, client will send out the TSR message to reset the timestamp clock. Afterwards, client SHOULD only send out the TSR message to reset timestamp when its local timestamp clock exceeds a pre-defined value, e.g. 0x7FFF0000.

During the GMA operation, client SHOULD continuously perform path quality measurements (e.g. one-way delay, loss, etc.) based on probing as well as received user-plane packets, and manage user-plane 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, and not considered in this document. Moreover, it is up to client implementation which delivery connection is used to send control messages, e.g. TSU, TSR, etc. However, the ACK message SHALL use the same delivery connection as its corresponding request message.

If client decides to update the traffic splitting configuration for downlink flows, it SHOULD send out the TSU message to gateway, notifying the updated configuration, and gateway SHOULD send out the TSA message to confirm the update and also indicate the Flow SN Of the first packet with the updated configuration.

For uplink traffic, if splitting is not enabled, client SHOULD control how to steer traffic without any GMA control message exchange with multi-access gateway. Otherwise, if splitting is enabled, multi-access gateway SHOULD perform measurements for the splitting-enabled uplink flow based on received data packets and send the TSU message to client for updating the traffic splitting configuration.



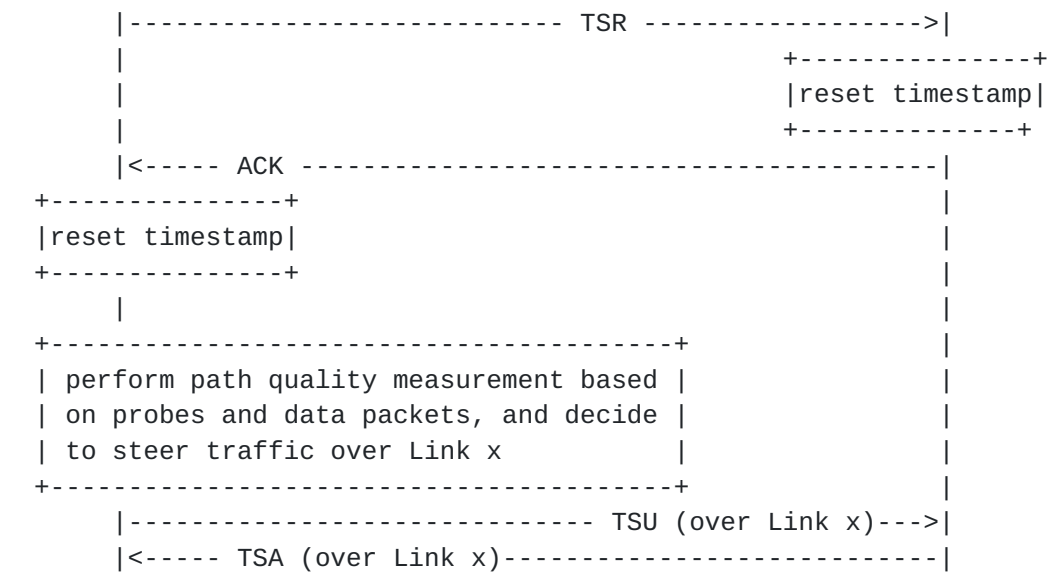


Figure 13: GMA-based Multi-Access Traffic Management Procedure

6.3. Termination

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.

Figure 14: MAMS-based Termination Procedure

7. Security Considerations

Security in a network using GMA should be relatively similar to security in a normal IP network. GMA is unaware of IP or higher layer end-to-end security as it carries the IP packets as opaque payload. Deployers are encouraged to not consider that GMA adds any form of security and to continue to use IP or higher layer security as well as link-layer security.

8. IANA Considerations

This document makes no requests of IANA.

9. References

9.1. 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> .

9.2. Informative References

[MAMS] S. Kanugovi, F. Baboescu, J. Zhu, and S. Seo "Multi-Access Management Services (MAMS)https://tools.ietf.org/rfc/rfc8743.txt

[IANA] https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml

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

[RFC791] Internet Protocol, September 1981

[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

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

[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. Generic Multi-Access (GMA) Encapsulation, Protocolhttps://www.ietf.org/archive/id/draft-zhu-intarea-gma-10.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, https://eeweb.engineering.nyu.edu/faculty/yongliu/docs/MPIP_Tech.pdf

Authors' Addresses

Jing Zhu

Intel

Email: jing.z.zhu@intel.com

Menglei Zhang

Intel

Email: menglei.zhang@intel.com