Stream: Independent Submission
RFC: 9188
Category: Experimental
Published: February 2022
ISSN: 2070-1721
Authors: J. Zhu *Intel* S. Kanugovi *Nokia*

RFC 9188
Generic Multi-Access (GMA) Encapsulation Protocol

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 the quality of experience for applications that do not have built-in multi-path capabilities. Such optimization requires additional control information, e.g., a sequence number, in each packet. This document presents a new lightweight and flexible encapsulation protocol for this need. The solution has been developed by the authors based on their experiences in multiple standards bodies including the IETF and 3GPP. However, this document 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

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 the 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, fragmentation, concatenation, 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.

```
         +-----------------------------------------------------+
         |   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

GRE (Generic Routing Encapsulation) [LWIPEP] [GRE1] [GRE2] can be used as the encapsulation protocol at the convergence layer to encode additional control information, e.g., key and sequence number. However, there are two main drawbacks with this approach:

- It is difficult to introduce new control fields because the GRE header formats are already defined, and
- IP-over-IP tunneling (required for GRE) leads to higher overhead especially for small packets.

For example, the overhead of IP-over-IP/GRE tunneling with both key and sequence Number is 32 bytes (20-byte IP header + 12-byte GRE header), which is 80% of a 40-byte TCP ACK packet.

This document presents a lightweight Generic Multi-Access (GMA) encapsulation protocol for the convergence layer. It supports three encapsulation methods: trailer-based IP encapsulation, header-based IP encapsulation, and non-IP encapsulation. Particularly, the IP encapsulation methods avoid IP-over-IP tunneling overhead (20 bytes), which is 50% of a 40-byte TCP ACK packet. Moreover, it introduces new control fields to support fragmentation and concatenation, which are not available in GRE-based solutions [LWIPEP] [GRE1] [GRE2].

The GMA protocol only operates between endpoints that have been configured to use GMA. This configuration can be through any control messages and procedures, including, for example, Multi-Access Management Services [MAMS]. Moreover, UDP or IPsec tunneling can be used at the adaptation sublayer to protect GMA operation from intermediate nodes.

The solution described in this document was developed by the authors based on their experiences in multiple standards bodies including the IETF and 3GPP. However, this document 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. The objective of the experiment is to determine whether the protocol meets the requirements, can be safely used, and has support for deployment.

Section 4 describes three possible encapsulation methods that are enabled by this protocol. Part of this experiment is to assess whether all three mechanisms are necessary or whether, for example, all implementations are able to support the main "trailer-based" IP encapsulation method. Similarly, the experiment will investigate the relative merits of the IP and non-IP encapsulation methods.

It is expected that this protocol experiment can be conducted on the Internet since the GMA packets are identified by an IP protocol number and the protocol is intended for single-hop propagation; devices should not be forwarding packets, and if they do, they will not need to examine the payload, while destination systems that do not support this protocol should not receive such packets and will handle them as unknown payloads according to normal IP processing. Thus, experimentation is conducted between consenting end systems that have been mutually configured to participate in the experiment as described in Section 7.

Note that this experiment "reuses" the IP protocol identifier 114 as described in Section 4.4. Part of this experiment is to assess the safety of doing this. The experiment should consider the following safety mechanisms:

- GMA endpoints **SHOULD** detect non-GMA IP packets that also use 114 and log an error to report the situation (although such error logging **MUST** be subject to rate limits).
- GMA endpoints **SHOULD** stop using 114 and switch to non-IP encapsulation, i.e., UDP encapsulation (Figure 7), after detecting any non-GMA usage of 114.

The experiment **SHOULD** use a packet tracing tool, e.g., WireShark or TCPDUMP, to monitor both ingress and egress traffic at GMA endpoints and ensure the above safety mechanisms are implemented.

Path quality measurements (one-way delay, loss, etc.) and congestion detection are performed by the receiver based on the GMA control fields, e.g., Sequence Number, Timestamp, etc. The receiver will then dynamically control how to split or steer traffic over multiple delivery connections accordingly. The GMA control protocol [GMAC] **MAY** be used for signaling between GMA endpoints. Another objective of the experiment is to evaluate the usage of various receiver-based algorithms [GCC] [MPIP] in multi-path traffic management and the impact on the End-to-End (E2E) performance (throughput, delay, etc.) of higher-layer (transport) protocols, e.g., TCP, QUIC, WebRTC, etc.

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, one of which (e.g., LTE) may operate as the anchor connection, and the other (e.g., Wi-Fi) may operate as the delivery connection. The anchor connection provides the IP address and connectivity for end-to-end Internet access, and the delivery connection provides an additional path between the client and Multi-Access Gateway for multi-access optimizations.

```
                 Multi-Access Aggregation

             +---+                          +---+
             |  |A|--- LTE Connection -----|C|  |
             |U|-|                          |-|S|  Internet
             |  |B|--- Wi-Fi Connection ---|D|  |
             +---+                          +---+
            client                Multi-Access Gateway
```

Figure 2: GMA-Based Multi-Access Aggregation

A: The adaptation-layer endpoint of the LTE connection resides in the client.

B: The adaptation-layer endpoint of the Wi-Fi connection resides in the client.

C: The adaptation-layer endpoint of the LTE connection resides in the Multi-Access Gateway, aka N-MADP (Network Multi-Access Data Proxy) in [MAMS].

D: The adaptation-layer endpoint of the Wi-Fi connection resides in the Multi-Access Gateway.

U: The convergence-layer endpoint resides in the client.

S: The convergence-layer endpoint resides in the Multi-Access Gateway.

For example, per-packet aggregation allows a single IP flow to use the combined bandwidth of the two connections. In another example, packets lost due to a temporary link outage may be retransmitted. 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 information, e.g., a sequence number, in each packet, which can be supported by the GMA encapsulation protocol described in this document.

The GMA protocol described in this document is designed for multiple connections, but it may also be used when there is only one connection between two endpoints. For example, it may be used for loss detection and recovery. In another example, it may be used to concatenate multiple small packets and reduce per-packet overhead.

4. GMA Encapsulation Methods

The GMA encapsulation protocol supports the following three methods:

- Trailer-based IP Encapsulation (Section 4.1)
- Header-based IP Encapsulation (Section 4.2)
- Header-based non-IP Encapsulation (Section 4.3)

Non-IP encapsulation **MUST** be used if the original IP packet is IPv6.

Trailer-based IP encapsulation **MUST** be used if it is supported by GMA endpoints and the original IP packet is IPv4.

Header-based encapsulation **MUST** be used if the trailer-based method is not supported by either the client or Multi-Access Gateway. In this case, if the adaptation layer, e.g., UDP tunneling, supports non-IP packet format, non-IP encapsulation **MUST** be used; otherwise, header-based IP encapsulation **MUST** be used.

If non-IP encapsulation is configured, a GMA header **MUST** be present in every packet. In comparison, if IP encapsulation is configured, a GMA header or trailer may be added dynamically on a per-packet basis, and it indicates the presence of a GMA header (or trailer) to set the protocol type of the GMA PDU to "114" (see Section 4.4).

The GMA endpoints **MAY** configure the GMA encapsulation method through control signaling or pre-configuration. For example, the "MX UP Setup Configuration Request" message as specified in Multi-Access Management Service [MAMS] includes "MX Convergence Method Parameters", which provides the list of parameters to configure the convergence layer, and can be extended to indicate the GMA encapsulation method.

GMA endpoint **MUST** discard a received packet and **MAY** log an error to report the situation (although such error logging **MUST** be subject to rate limits) under any of the following conditions:

- The GMA version number in the GMA header (or trailer) is not understood or supported by the GMA endpoint.
- A flag bit in the GMA header (or trailer) not understood or supported by the GMA endpoint is set to "1".

4.1. Trailer-Based IP Encapsulation

```
|<---------------------GMA PDU ---------------------->|
+-----------------------------------------------------+
| IP hdr |        IP payload            | GMA Trailer |
+-----------------------------------------------------+
|<------- GMA SDU (user payload)-------->|
```

Figure 3: GMA PDU Format with Trailer-based IP Encapsulation

This method **SHALL NOT** be used if the original IP packet (GMA service data unit (GMA SDU)) is IPv6.

Figure 3 shows the trailer-based IP encapsulation GMA protocol data unit (GMA PDU) format. A (GMA) PDU may carry one or multiple IP packets, aka (GMA) SDUs, in the payload, or a fragment of the SDU.

The protocol type field in the IP header of the GMA PDU **MUST** be changed to 114 (Any 0-Hop Protocol) (see Section 4.4) to indicate the presence of the GMA trailer.

The following three IP header fields **MUST** be changed:

IP Length: Add the length of "GMA Trailer" to the length of the original IP packet.

Time To Live (TTL): Set to "1".

IP checksum: Recalculate after changing "protocol type", "TTL", and "IP Length".

The GMA (Generic Multi-Access) trailer **MUST** consist of two mandatory fields (the last 3 bytes): Next Header and Flags.

This is defined as follows:

Next Header (1 byte): This is the IP protocol type of the (first) SDU in a PDU; it stores the value before it was overwritten to 114.

Flags (2 bytes): Bit 0 is the most significant bit (MSB), and bit 15 is the least significant bit (LSB).

Checksum Present (bit 0): If the Checksum Present bit is set to 1, then the Checksum field is present.

Concatenation Present (bit 1): If the Concatenation Present bit is set to 1, then the PDU carries multiple SDUs, and the First SDU Length field is present.

Connection ID Present (bit 2): If the Connection ID Present bit is set to 1, then the Connection ID field is present.

Flow ID Present (bit 3): If the Flow ID Present bit is set to 1, then the Flow ID field is present.

Fragmentation Present (bit 4): If the Fragmentation Present bit is set to 1, then the PDU carry a fragment of the SDU and the Fragmentation Control field is present.

Delivery SN Present (bit 5): If the Delivery SN (Sequence Number) Present bit is set to 1, then the Delivery SN field is present and contains the valid information.

Flow SN Present (bit 6): If the Flow SN Present bit is set to 1, then the Sequence Number field is present.

Timestamp Present (bit 7): If the Timestamp Present bit is set to 1, then the Timestamp field is present.

TTL Present (bit 8): If the TTL Present bit is set to 1, then the TTL field is present.

Reserved (bit 9-12): This is set to "0" and ignored on receipt.

Version (bit 13~15): This is the GMA version number; it is set to 0 for the GMA encapsulation protocol specified in this document.

The Flags field is at the end of the PDU, and the Next Header field is the second last. The receiver **SHOULD** first decode the Flags field to determine the length of the GMA trailer and then decode each optional field accordingly. The Generic Multi-Access (GMA) trailer **MAY** consist of the following optional fields:

Checksum (1 byte): This contains the (one's complement) checksum sum of all 8 bits in the trailer. For purposes of computing the checksum, the value of the Checksum field is zero. This field is present only if the Checksum Present bit is set to 1.

First SDU Length (2 bytes): This is the length of the first IP packet in the PDU, only included if a PDU contains multiple IP packets. This field is present only if the Concatenation Present bit is set to 1.

Connection ID (1 byte): This contains an unsigned integer to identify the anchor and delivery connection of the GMA PDU. This field is present only if the Connection ID Present bit is set to 1.

Anchor Connection ID (MSB 4 bits): This contains an unsigned integer to identify the anchor connection.

Delivery Connection ID (LSB 4 bits): This contains an unsigned integer to identify the delivery connection.

Flow ID (1 byte): This contains an unsigned integer to identify the IP flow that a PDU belongs to, for example Data Radio Bearer (DRB) ID [LWIPEP] for a cellular (e.g., LTE) connection. This field is present only if the Flow ID Present bit is set to 1.

Fragmentation Control (FC) (1 byte): This provides necessary information for reassembly, only needed if a PDU carries fragments. This field is present only if the Fragmentation Present bit is set to 1. Please refer to Section 5 for its detailed format and usage.

Delivery SN (1 byte): This contains an auto-incremented integer to indicate the GMA PDU transmission order on a delivery connection. Delivery SN is needed 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 1.

Flow SN (3 bytes): This contains an auto-incremented integer to indicate the GMA SDU (IP packet) order of a flow. Flow SN is needed for retransmission, reordering, and fragmentation. It **SHALL** be generated per flow. This field is present only if the Flow SN Present bit is set to 1.

Timestamp (4 bytes): This contains 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 1.

TTL (1 byte): This contains the TTL value of the original IP header if the GMA SDU is IPv4, or the Hop-Limit value of the IP header if the GMA SDU is IPv6. This field is present only if the TTL Present bit is set to 1.

Figure 4 shows the GMA trailer 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 reverse order to their corresponding flag bits. If a flag bit is clear, the corresponding optional field is absent.

For example, bit 0 (the MSB) of the Flags field is the Checksum Present bit, and the Checksum field is the last in the trailer with the exception of the two mandatory fields. Bit 1 is the Concatenation Present bit, and the FSL field is the second last.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      TTL      |                   Timestamp
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                 |                   Flow SN                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Delivery SN  |      FC       |    Flow ID    | Connection ID |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |       First SDU Length (FSL)  |    Checksum   |  Next Header  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         Flags                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: GMA Trailer Format with All Optional Fields Present

4.2. Header-Based IP Encapsulation

This method **SHALL NOT** be used if the original IP packet (GMA SDU) is IPv6.

Figure 5 shows the header-based IP encapsulation format. Here, the GMA header is inserted right after the IP header of the GMA SDU, and the IP header fields of the GMA PDU **MUST** be changed the same way as in trailer-based IP encapsulation.

```
+-----------------------------------------------+
|IP hdr | GMA Header  |       IP payload        |
+-----------------------------------------------+
```

Figure 5: GMA PDU Format with Header-Based IP Encapsulation

Figure 6 shows the GMA header format. In comparison to the GMA trailer, the only difference is that the Flags field is now in the front so that the receiver can first decode the Flags field to determine the GMA header length.

The "TTL" field **MUST** be included and the "TTL" bit in the GMA header (or Trailer) **MUST** be set to 1 if (trailer- or header-based) IP encapsulation is used.

```
+-------------------------------------------------------+
| Flags | other fields (TTL, Timestamp, Flow SN, etc.) |
+-------------------------------------------------------+
```

Figure 6: GMA Header Format

4.3. Header-Based Non-IP Encapsulation

Figure 7 shows the header-based non-IP encapsulation format. Here, "UDP Tunneling" is configured at the MX adaptation layer. The ports for "UDP Tunneling" at the client are chosen from the Dynamic Port range, and the ports for "UDP Tunneling" at the Multi-Access Gateway are configured and provided to the client through additional control messages, e.g., [MAMS].

"TTL", "FSL", and "Next Header" are no longer needed and **MUST** not be included. Moreover, the IP header fields of the GMA SDU remain unchanged.

```
+--------------------------------------------------------------+
| IP hdr | UDP hdr  | GMA Header | IP hdr |    IP payload     |
+--------------------------------------------------------------+
                                |<-------- GMA SDU------------->|
                    |<------------------- GMA PDU------------->|
```

Figure 7: GMA PDU Format with Non-IP Encapsulation

4.4. IP Protocol Identifier

As described in Section 4.1, IP-encapsulated GMA PDUs are indicated using the IP protocol type 114. This is designated and recorded by IANA [IANA] to indicate "any 0-Hop Protocol". No reference is given in the IANA registry for the definition of this protocol type, and IANA has no record of why the assignment was made or how it is used, although it was probably assigned before 1999 [IANA1999].

There is some risk associated with "reusing" protocol type 114 because there may be implementations of other protocols also using this protocol type. However, because the protocol described in this document is used only between adjacent devices specifically configured for this purpose, the use of protocol type 114 should be safe.

As described in Section 1.1, one of the purposes of the experiment described in this document is to verify the safety of using this protocol type. Deployments should be aware of the risk of a clash with other uses of this protocol type.

5. Fragmentation

If the MTU size of the anchor connection (for GMA SDU) is configured such that the corresponding GMA PDU length adding the GMA header (or trailer) and other overhead (UDP tunneling) **MAY** exceed the MTU of a delivery connection, GMA endpoints **MUST** be configured to support fragmentation through additional control messages [MAMS].

The fragmentation procedure at the convergence sublayer is similar to IP fragmentation [RFC0791] in principle, but with the following two differences for less overhead:

- The fragment offset field is expressed in number of fragments.
- The maximum number of fragments per SDU is 2^7 (=128).

The Fragmentation Control (FC) field in the GMA trailer (or header) contains the following bits:

Bit 7: a More Fragment (MF) flag to indicate if the fragment is the last one (0) or not (1)

Bit 0-6: Fragment Offset (in units of fragments) to specify the offset of a particular fragment relative to the beginning of the SDU

A PDU carries a whole SDU without fragmentation if the FC field is set to all "0"s or the FC field is not present in the trailer. Otherwise, the PDU contains a fragment of the SDU.

The Flow SN field in the trailer is used to distinguish the fragments of one SDU from those of another. The Fragment Offset (FO) field tells the receiver the position of a fragment in the original SDU. The More Fragment (MF) flag indicates the last fragment.

To fragment a long SDU, the transmitter creates n PDUs and copies the content of the IP header fields from the long PDU into the IP header of all the PDUs. The length field in the IP header of the PDU **SHOULD** be changed to the length of the PDU, and the protocol type **SHOULD** be changed to 114.

The data of the long SDU is divided into n portions based on the MTU size of the delivery connection. The first portion of the data is placed in the first PDU. The MF flag is set to "1", and the FO field is set to "0". The i-th portion of the data is placed in the i-th PDU. The MF flag is set to "0" if it is the last fragment and set to "1" otherwise. The FO field is set to i-1.

To assemble the fragments of an SDU, the receiver combines PDUs that all have the same Flow SN. The combination is done by placing the data portion of each fragment in the relative order indicated by the Fragment Offset in that fragment's GMA trailer (or header). The first fragment will have the Fragment Offset set to "0", and the last fragment will have the More Fragment flag set to "0".

GMA fragmentation operates above the IP layer of individual access connection (Wi-Fi, LTE) and between the two endpoints of convergence layer. The convergence layer endpoints (client, Multi-access Gateway) **SHOULD** obtain the MTU of individual connection through either manual configuration or implementing Path MTU Discovery (PMTUD) as suggested in [RFC8900].

6. Concatenation

The convergence sublayer **MAY** support concatenation if a delivery connection has a larger maximum transmission unit (MTU) than the original IP packet (SDU). Only the SDUs with the same client IP address and the same Flow ID **MAY** be concatenated.

If the (trailer- or header-based) IP encapsulation method is used, the First SDU Length (FSL) field **SHOULD** be included in the GMA trailer (or header) to indicate the length of the first SDU. Otherwise, the FSL field **SHOULD** not be included.

```
+-----------------------------------------------------------+
|IP hdr| IP payload    |IP hdr|   IP payload  | GMA Trailer |
+-----------------------------------------------------------+
```

Figure 8: Example of GMA PDU Format with Concatenation and Trailer-Based IP Encapsulation

To concatenate two or more SDUs, the transmitter creates one PDU and copies the content of the IP header field from the first SDU into the IP header of the PDU. The data of the first SDU is placed in the first portion of the data of the PDU. The whole second SDU is then placed in the second portion of the data of the PDU (Figure 8). The procedure continues until the PDU size reaches the MTU of the delivery connection. If the FSL field is present, the IP Length field of the PDU **SHOULD** be updated to include all concatenated SDUs and the trailer (or header), and the IP checksum field **SHOULD** be recalculated if the packet is IPv4.

To disaggregate a PDU, if the (header- or trailer-based) IP encapsulation method is used, the receiver first obtains the length of the first SDU from the FSL field and decodes the first SDU. The receiver then obtains the length of the second SDU based on the length field in the second SDU IP header and decodes the second SDU. The procedure continues until no byte is left in the PDU. If the non-IP encapsulation method (Figure 7) is used, the IP header of the first SDU will not change during the encapsulation process, and the receiver **SHOULD** obtain the length of the first SDU directly from its IP header (Figure 9).

```
                                 |<-------1st GMA SDU------------
+----------------------------------------------------------------+
| IP hdr | UDP hdr  | GMA Header | IP hdr |       IP payload    |
+----------------------------------------------------------------+
         | IP hdr |           IP payload    |
+-----------------------------------------+
-------->|<-------2nd GMA SDU--------------->
```

Figure 9: Example of GMA PDU Format with Concatenation and Header-Based Non-IP (UDP) Encapsulation

If a PDU contains multiple SDUs, the Flow SN field is for the last SDU, and the Flow SN of other SDUs carried by the same PDU can be obtained according to its order in the PDU. For example, if the SN field is 6 and a PDU contains 3 SDUs (IP packets), the SN is 4, 5, and 6 for the first, second, and last SDU, respectively.

GMA concatenation can be used for packing small packets of a single application, e.g., TCP ACKs, or from multiple applications. Notice that a single GMA flow may carry multiple application flows (TCP, UDP, etc.).

GMA endpoints **MUST NOT** perform concatenation and fragmentation in a single PDU. If a GMA PDU carries a fragmented SDU, it **MUST NOT** carry any other (fragmented or whole) SDU.

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.

The GMA protocol at the convergence sublayer is a 0-hop protocol and relies on the security of the underlying network transport paths. When this cannot be assumed, appropriate security protocols (IPsec, DTLS, etc.) **SHOULD** be configured at the adaptation sublayer. On the other hand, packet filtering requires either that a firewall looks inside the GMA packet or that the filtering is done on the GMA endpoints. In those environments in which this is considered to be a security issue, it may be desirable to terminate the GMA operation at the firewall.

Local-only packet leak prevention (HL=0, TTL=1) **SHOULD** be on preventing the leak of the local-only GMA PDUs outside the "local domain" to the Internet or to another domain that could use the same IP protocol type, i.e., 114.

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[GRE1] Dommety, G., "Key and Sequence Number Extensions to GRE", RFC 2890, DOI 10.17487/RFC2890, September 2000, <https://www.rfc-editor.org/info/rfc2890>.

[GRE2] Leymann, N., Heidemann, C., Zhang, M., Sarikaya, B., and M. Cullen, "Huawei's GRE Tunnel Bonding Protocol", RFC 8157, DOI 10.17487/RFC8157, May 2017, <https://www.rfc-editor.org/info/rfc8157>.

[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>.

9.2. Informative References

[ATSSS] 3GPP, "Study on access traffic steering, switch and splitting support in the 5G System (5GS) architecture", Release 16, 3GPP TR 23.793, December 2018, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3254>.

[GCC] Holmer, S., Lundin, H., Carlucci, G., De Cicco, L., and S. Mascolo, "A Google Congestion Control Algorithm for Real-Time Communication", Work in Progress, Internet-Draft, draft-ietf-rmcat-gcc-02, 8 July 2016, <https://datatracker.ietf.org/doc/html/draft-ietf-rmcat-gcc-02>.

[GMAC] Zhu, J. and M. Zhang, "UDP-based Generic Multi-Access (GMA) Control Protocol", Work in Progress, Internet-Draft, draft-zhu-intarea-gma-control-00, 13 October 2021, <https://datatracker.ietf.org/doc/html/draft-zhu-intarea-gma-control-00>.

[IANA] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers>.

[IANA1999] IANA, Wayback Machine archive of "Protocol Numbers", February 1999, <https://web.archive.org/web/19990203044112/http://www.isi.edu:80/in-notes/iana/assignments/protocol-numbers>.

[LWIPEP] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); LTE-WLAN Radio Level Integration Using Ipsec Tunnel (LWIP) encapsulation; Protocol specification", Release 13, 3GPP TS 36.361, July 2020, <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3037>.

[MAMS] Kanugovi, S., Baboescu, F., Zhu, J., and S. Seo, "Multiple Access Management Services Multi-Access Management Services (MAMS)", RFC 8743, DOI 10.17487/RFC8743, March 2020, <https://www.rfc-editor.org/info/rfc8743>.

[MPIP] Sun, L., Tian, G., Zhu, G., Liu, Y., Shi, H., and D. Dai, "Multipath IP Routing on End Devices: Motivation, Design, and Performance", 2017, <https://eeweb.engineering.nyu.edu/faculty/yongliu/docs/MPIP_Tech.pdf>.

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[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>.

Authors' Addresses

Jing Zhu
Intel
Email: jing.z.zhu@intel.com

Satish Kanugovi
Nokia
Email: satish.k@nokia.com