Network Working Group
Internet Draft
Intended status: Informational
Expires: September 3, 2025

J. Zhu (Ed.)
NSS LLC
M. Zhang
ByteDance
S. Roy
University of Washington
March 3, 2025

GMA Traffic Splitting Control

draft-zhu-gma-tsc-04

Abstract

This document specifies the GMA (Generic Multi-Access) traffic splitting control algorithm. The receiving endpoint measures one-way-delay, round-trip time, and delivery rate for multiple connections and determines how a data flow is split across them. When update is needed, it will send out a control message, aka Traffic Splitting Update (TSU), to notify the transmitting endpoint of the new traffic splitting configuration. Compared to other sender-based multi-path transport protocols, e.g. MPTCP, MPQUIC, the GMA traffic splitting algorithm is receiver-based and does not require per-packet feedback, e.g. Ack. It is designed specifically to support the Generic Multi-Access (GMA) convergence protocol as introduced in [MAMS] [GMA]. 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.

Status of this Memo

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.

The 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 UDP-based GMA control protocol [GMA] has been proposed for the MX convergence layer in the MAMS framework. 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 single-path transport protocol, e.g. TCP, UDP, QUIC, etc. From the perspective of an 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.

This document presents a receiver-based multi-path traffic splitting algorithm for the MX convergence layer. Unlike other sender-based multi-path solutions, e.g. MPTCP, it does not require per-packet feedback, e.g. ACK, and leverages One-Way-Delay (OWD) measurements that are available at the receiver.

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.

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 GMA Traffic Splitting Control (GMA-TSC) Algorithm

There are four modules involved in the GMA-TSC algorithm:

- o minimum OWD measurement
- o congestion measurement
- o connection failure detection
- o multi-path traffic splitting control

3.1 Minimum OWD Measurement

The GMA receiver performs minimum OWD measurement periodically based on received data and control packets. The time unit is milliseconds (ms).

Define the following notations:

- o d(k, i): the OWD of the k-th received packet over the i-th connection
- o Y(i): the minimum OWD of the i-th connection
- o d’(i): the OWD of the last received packet over the i-th connection

The GMA receiver SHOULD update Y(i) at the end of each minimum OWD measurement interval, and obtain the minimum OWD of the i-th connection as following:

$$Y(i) = min(d(k, i), for all k)$$

In addition, the receiver SHOULD set Y(i) to d'(i) immediately when receiving a packet with d'(i) < Y(i) - B1, where B1 is a configurable parameter, say 5ms.

3.2 Congestion Measurement

The GMA receiver performs congestion measurement periodically based on received data & control packets. The congestion measurement interval SHOULD be set much shorter, e.g. < 1s, than the minimum OWD measurement interval, e.g. 12 s. It will start right after a successful TSU/TSA exchange [GMA] or the previous interval if the TSU/TSA exchange is not triggered.

Define the following control parameters:

- T1: the minimum congestion measurement duration
- T2: the minimum number of data packets for congestion measurement
- Dmin: the lower bound of congestion measurement interval
- Dmax: the upper bound of congestion measurement interval

T1 is configured as follows:

$$T1 = a1 * V$$

Herein, a1 is a configurable coefficient, e.g., 1.5. V indicates the maximum of average RTT of all connections and SHOULD be updated at the end of each interval.

Define the following notations:

- t(i): the RTT of the last control message exchange over the i-th connection
- d0(i): the OWD of the received control message for the last RTT measurement t(i)
- v(i): the average RTT of the i-th connection

The average round-trip time of the i-th connection can be measured as

$$v(i)=t(i) - d0(i) + average(d(k, i), for all k)$$

Then, we can get V as

$$V = max(v(i), for all i)$$

T2 is configured as follows:

T2 = a2 * L

Herein, a2 is another configurable coefficient, e.g. 2, and L is the splitting burst size, defined as the total number of packets per traffic splitting cycle.

A congestion measurement interval will end only if both T1 and T2 are reached. It is further bounded by a configurable range of [Dmin, Dmax], e.g. [10ms, 1s].

The GMA receiver will measure the following metrics during each congestion measurement interval:

- o k(i): the number of received data packets experiencing congestions for the i-th connection
- o n(i): the total number of received data packets for the i-th connection
- o b(i): the estimated bandwidth for the i-th connection in unit of packets/second

Let’s use u to indicate one of the last U (e.g. 10) congestion measurement intervals, including the current one. Figure 2 shows an example of four (U=4) consecutive congestion measurement intervals.

```
       |----u=3---|---u=2----|--u=1---|--u=0---|
   ---------------------------------------------------> Time
```

Figure 2: Congestion Measurement Intervals (U=4)

Moreover, define the following notations:

- o D(u): the interval length of the u-th interval
- o n(i, u): the number of received data packets on the i-th connection during the u-th interval
- o k(i, u): the number of received data packets experiencing congestions on the i-th connection during the u-th interval

We can calculate b(i) as follows:

if (max(k(i, u), for all u) > 0

b(i) = max(n(i, u) / D(u), for all u)

else

$$b(i) = 0$$

When u = 0, it refers to the current interval, i.e., n(i, u=0) = n(i), k(i, u=0)= k(i), and D(u=0) = D. In short, b(i) measures the maximum throughput of the i-th connection in the last U intervals when at least one received data packet experienced congestion Here, we define that a packet is experiencing congestion if its OWD exceeds the minimum OWD with a specified threshold, i.e.,

$$d(k, i) > Y(i) + B2$$

Here, B2 is a configurable threshold, e.g. 10ms.

Notice that no congestion measurement SHOULD be performed during the TSU/TSA exchange.

3.3   Connection Failure Detection

At the end of a congestion measurement interval, if the GMA receiver detects any potential connection failure, it will send a control message, e.g. probe, to check the connection status explicitly. Connection failure can then be confirmed through successive retransmission failures of the control message. Otherwise, it is a false alarm.

Define s(i) as the traffic splitting ratio in the range of [0, 1] to indicate how much data traffic of a flow is delivered through the i-th connection. The connection failure detection method works as follows:

- o When a flow is being split over two or more connections, a connection will be flagged with "failure" if the connection has s(i)>0 but n(i)=0, and the total number of received packets reaches or exceeds T3, i.e. sum(n(i))> T3.
- o When a flow is being steered to a single connection, the connection will be flagged with "failure" if no packets are received in the current interval (u=0), e.g., sum(n(i)) = 0 and Q = 1. Here, Q is a bit flag to indicate if any data packet of the flow is received ("1") or not ("0") in the previous interval (u=1).

T3 is configured as follows:

$$T3 = max(P, T2)$$

and P is a configurable lower-bound, e.g. 128 to ensure that when L is too small, e.g. 8, there are still enough measurement samples for connection failure detection to minimize false alarms.

3.4 Multi-path Traffic Splitting Control

The connections are ordered according to their preference, and its index is used to indicate its preference. A connection is preferred over another if its index is smaller. For example, the 1-st connection with i=1 is the most preferred one, and the 2nd connection with i=2 is the next preferred one.

At the end of a congestion measurement interval, the GMA receiver SHOULD recalculate traffic splitting ratio s(i) using the following principals:

- o For any connection flagged as a failure (see 3.3), its traffic splitting ratio s(i) is set to 0.
- o For other connections, consider the following four cases:
  - + Case #1 (No Traffic): If no packet is received at all, i.e., sum(n(i)) = 0, stop splitting and steer the flow to the most preferred available connection.
  - + Case #2 (No Congestion): If no packet experienced congestion, i.e. sum(k(i)) = 0, stop sending over the least preferred connection with none-zero traffic load, i.e., max(i | n(i)>0), and split the flow over the more preferred available connections.
  - + Case #3 (Medium Congestion): When only a subset of connections experienced congestion, i.e., min(k(i))=0 and max(k(i)) > 0, reallocate a portion of the flow from a congested connection to non-congested ones.
  - + Case #4 (Heavy Congestion): If all connections are congested, i.e., min(k(i))>0, split the flow in proportion to n(i).

The GMA traffic splitting algorithm can be described as follows:

If (sum(n(i)) = 0) //no traffic

- o x: the most preferred available connection
- o j: other available connections
- o s(x) = 1.0 and s(j) = 0

else if (sum(k(i)) = 0) //no congestion

- o x: the least preferred connection with none-zero load, i.e., x = max(i | n(i)>0)
- o j: another available connection more preferred than "x"
- o S: the total number of available connections more preferred than "x"
- o R = min(n(x), p*sum(n(i))), where p = S/L
- o s(x) = (n(x) - R)/sum(n(i))

```
      o if (min(b(j)) = 0)
           s(j) = (n(j) + R/S)/sum(n(i))
      o else
           s(j) = (sum(n(j)) + R)*b(j)/sum(b(j))/(sum(n(i))

   else if (min(k(i))= 0 && max(k(i)) > 0) //medium congestion

      o x: the congested connection, i.e., k(x)>0
      o j: the uncongested connection, i.e., k(j)=0
      o s(x) = (n(x) - e*k(x))/sum(n(i)), where e = 0.3
      o if (min(b(j)) = 0)
           s(j)=(n(j) + e*sum(k(x))/M)/sum(n(i)), where M is the
         number of uncongested connections
      o else
           s(j)=(sum(n(j)) + e *sum(k(x)))*b(j)/sum(b(j)/sum(n(i))

   else if (min(k(i)) > 0) //heavy congestion

      o s(i) = n(i)/sum(n(i))
```

4 Enhancements

4.1 Adaptive Control in Medium Congestion

In the case of medium congestion (case #3 in 3.4), some data packets are reallocated from a congested connection to uncongested ones. However, the total amount of reallocated traffic, given by e*sum(k(x)) should not exceed the total available bandwidth of uncongested connections, given by sum(b(j))*D - sum(n(j)), where D is the current interval length. Hence, we can adaptively configure e with the following two steps:

- o step 1: e = (sum(b(j))*D - sum(n(j)))/sum(k(x))
- o step 2: e = max(Emin, min(e, Emax)), where Emin = 0.1 and Emax = 0.5.

Notice that step 2 limits the range of e to [Emin, Emax].

4.2 Minimizing TSU/TSA Overhead

If the splitting ratio update is too small, the GMA receiver SHOULD NOT initiate a TSU/TSA exchange to minimize signaling overhead. Using s’(i) and s(i) to indicate the current and new splitting ratio respectively, the TSU/TSA exchange will be triggered only if

max(|s’(i) - s(i)|) > B3

Here, B3 is a configurable threshold, e.g. 3%.

4.3   Adaptive Splitting Burst Size

The traffic splitting control granularity is given by 1/L. Larger the splitting burst size L, finer the control granularity. L also controls the congestion measurement interval. Smaller the splitting burst size, shorter the measurement interval so that the algorithm will converge faster. Therefore, L should be set as large as possible but not increasing the congestion measurement interval, i.e. D < or = T1. Let’s denote L as

$$L = 2^{(max(p, a3))}$$

Wherein, a3 is a configurable constant to determine the minimum splitting burst size. For example, with a3=3, we have L > or = 8. The GMA receiver SHOULD dynamically adjust p at the end of each measurement interval as

$$p = floor(log2(sum(n(i))* T1/(D * a2)))$$

If D = T1, p can be simplified as

$$p = floor(log2(sum(n(i))/a2))$$

For example, if sum(n(i)) = 50 and a2 = 2, we will have L = 16.

4.4   Traffic Splitting Ratio Quantization

Let’s use S(i) to indicate the number of packets sent to the i-th connection for every traffic splitting burst, i.e., sum(S(i))=L. S(i) is an integer in the range of [0, L], and the following method MAY be used to obtain S(i) from the new traffic splitting ratio s(i), which is a decimal in the range of [0, 1].

- step 1: determine S(j) for the connections with reduced traffic splitting ratio, i.e. $s(j)<s'(j)$, where s’(j) is the current splitting ratio.

$$S(j)=round(L*s(j)-a4)$$

Herein, a4 is a configurable constant, e.g. 0.3.

- Step 2: determine S(i) for other connections:

$$S(i)=round((L-sum(S(j)|s(j)<s'(j)))*s(i)/sum(s(i)|s(i)>or=s'(i)))$$

o Step 3: adjust S(m) for the connection with highest traffic splitting ratio, i.e., S(m)=max(S(i)), to ensure sum(S(i))=L.

S(m)=S(m) + L - sum(S(i))

5 Security Considerations

This proposal makes no changes to the underlying security of GMA protocol [GMA].

6 IANA Considerations

This document makes no requests of IANA.

7 References

7.1 Informative References

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

[GMA] J. Zhu, M. Zhang, A UDP-based GMA (Generic Multi-Access) Protocol <https://www.ietf.org/archive/id/draft-zhu-intarea-gma-control-06.txt>

Authors’ Addresses

Jing Zhu

Network Simulation Solutions (NSS) LLC

Email: jing.zhu.ietf@gmail.com

Menglei Zhang

ByteDance

Email: mengleizhang.mz@gmail.com

Sumit Roy

University of Washington

Email: sroy@uw.edu