INTAREA
Internet Draft
Intended status: Standards Track
Expires: December 16, 2017

J. Zhu
Intel
S. Seo
Korea Telecom
S. Kanugovi
Nokia
S. Peng
Huawei

June 16, 2017

User-Plane Protocols for Multiple Access Management Service
draft-zhu-intarea-mams-user-protocol-02

Status of this Memo

Copyright Notice

Abstract

Today, a device can be simultaneously connected to multiple communication networks based on different technology implementations and network architectures like WiFi, LTE, DSL. In such multi-connectivity scenario, it is desirable to combine multiple access networks or select the best one to improve quality of experience for a user and improve overall network utilization and efficiency. This document presents the u-plane protocols for a multi access management services (MAMS) framework that can be used to flexibly select the combination of uplink and downlink access and core network paths having the optimal performance, and user plane treatment for improving network utilization and efficiency and enhanced quality of experience for user applications.

Table of Contents

1. Introduction

Multi Access Management Service (MAMS) [MAMS] is a programmable framework to select and configure network paths, as well as adapt to dynamic network conditions, when multiple network connections can serve a client device. It is based on principles of user plane interworking that enables the solution to be deployed as an overlay without impacting the underlying networks.

This document presents the u-plane protocols for enabling the MAMS framework. It co-exists and complements the existing protocols by providing a way to negotiate and configure the protocols based on client and network capabilities. Further it allows exchange of network state information and leveraging network intelligence to optimize the performance of such protocols. An important goal for MAMS is to ensure that there is minimal or no dependency on the actual access technology of the participating links. This allows the scheme to be scalable for addition of newer accesses and for independent evolution of the existing access technologies.

2. Terminologies

Anchor Connection: refers to the network path from the N-MADP to the Application Server that corresponds to a specific IP anchor that has assigned an IP address to the client

Delivery Connection: refers to the network path from the N-MADP to the C-MADP.

"Network Connection Manager" (NCM), "Client Connection Manager" (CCM), "Network Multi Access Data Proxy" (N-MADP), and "Client Multi Access Data Proxy" (C-MADP) in this document are to be interpreted as described in [MAMS].

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The terminologies "Network Connection Manager" (NCM), "Client Connection Manager" (CCM), "Network Multi Access Data Proxy" (N-MADP), and "Client Multi Access Data Proxy" (C-MADP) in this document are to be interpreted as described in [MAMS].

4 MAMS User-Plane Protocols

Figure 1 shows the MAMS u-plane protocol stack as specified in [MAMS_CP].

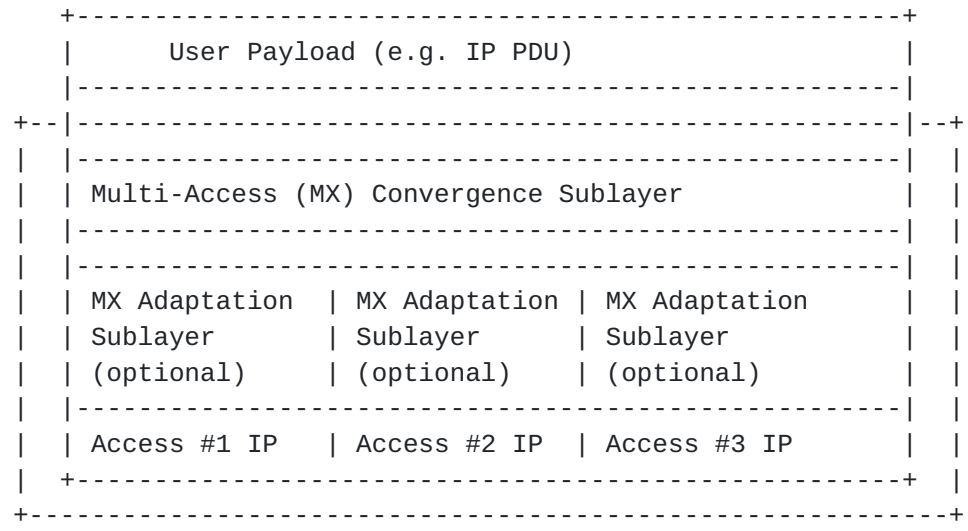


Figure 1: MAMS U-plane Protocol Stack

It consists of the following two Sublayers:

- Multi-Access (MX) Convergence Sublayer: This layer performs multi-access specific tasks, e.g., access (path) selection, multi-link (path) aggregation, splitting/reordering, lossless switching, fragmentation, concatenation, keep-alive, probing etc.
- Multi-Access (MX) Adaptation Sublayer: This layer performs functions to handle tunneling, network layer security, and NAT.

The MX convergence sublayer operates on top of the MX adaptation sublayer in the protocol stack. From the Transmitter perspective, a User Payload (e.g. IP PDU) is processed by the convergence sublayer first, and then by the adaptation sublayer before being transported over a delivery access connection; From the Receiver perspective, an IP packet received over a delivery connection is processed by the MX adaptation sublayer first, and then by the MX convergence sublayer.

4.1 MX Adaptation Layer

The MX adaptation layer supports the following mechanisms and protocols while transmitting user plane packets on the network path:

o UDP Tunneling: The user plane packets of the anchor connection can be encapsulated in a UDP tunnel of a delivery connection between the N-MADP and C-MADP.
o IPsec Tunneling: The user plane packets of the anchor connection are sent through an IPSec tunnel of a delivery connection.
o Client Net Address Translation (NAT): change the Client IP address of user plane packet of the anchor connection, and send it over a delivery connection.
o Pass Through: The user plane packets are passing through without any change over the anchor connection.

The MX adaptation layer also supports the following mechanisms and protocols to ensure security of user plane packets over the network path.

o IPSec Tunneling: An IPsec [RFC7296] tunnel is established between the N-MADP and C-MADP on the network path that is considered untrusted.
o DTLS: If UDP tunneling is used on the network path that is considered "untrusted", DTLS (Datagram Transport Layer Security) [RFC6347] can be used.

The Client NAT method is the most efficient due to no tunneling overhead. It SHOULD be used if a delivery connection is "trusted" and without NAT function on the path.

The UDP or IPSec Tunnelling method SHOULD be used if a delivery connection has a NAT function placed on the path.

4.2 Trailer-based MX Convergence Layer

4.2.1 Trailer-based MX PDU Format

Trailer-based MX convergence integrates multiple connections into a single e2e IP connection. It operates between Layer 2 (L2) and Layer 3 (network/IP).

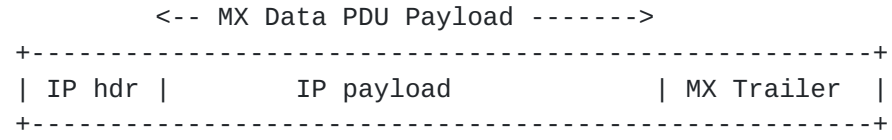


Figure 2: Trailer-based Multi-Access (MX) Data PDU Format

Figure 2 shows the trailer-based Multi-Access (MX) PDU (Protocol Data Unit) format. A MX PDU MAY carry multiple IP PDUs in the payload if concatenation is supported, and MAY carry a fragment of the IP PDU if fragmentation is supported.

The MX trailer may consist of the following fields:

- MX flags (e.g. 1 byte): Bit 0 is the most significant bit, bit 7 is the least significant bit. Bit 6 and 7 are reserved for future.
  - Next Header Present (bit 0): If the Next Header Present bit is set to 1, then the Next Header field is present and contains valid information.
  - Connection ID Present (bit 1): If the Connection ID Present bit is set to 1, then the Connection ID field is present and contains valid information.
  - Traffic Class Present (bit 2): If the Traffic Class Present bit is set to 1, then the Traffic Class field is present and contains valid information.
  - Sequence Number Present (bit 3): If the Sequence Number Present bit is set to 1, then the Sequence Number field is present and contains valid information.
  - Packet Length Present (bit 4): If the Packet Length Present bit is set to 1, then the First SDU Length field is present and contains valid information.
  - Fragmentation Control Present (bit 5): If the Fragmentation Control Present bit is set to 1, then the Fragmentation Control field is present and contains valid information.
  - Bit 6~7: reserved
- Next Header (e.g. 1 byte): the IP protocol type of the (first) IP packet in a MX PDU
- Connection ID (e.g.1 byte): an unsigned integer to identify the anchor connection of the IP packets in a MX PDU
- Traffic Class (TC) ID (e.g. 1 byte): an unsigned integer to identify the traffic class of the IP packets in a MX PDU, for example Data Radio Bearer (DRB) ID [LWIPEP] for a cellular (e.g. LTE) connection
- Sequence Number (e.g. 2 bytes): an auto-incremented integer to indicate order of transmission of the IP packet, needed for lossless switching or multi-link (path) aggregation or fragmentation.
- First SDU Length (e.g. 2 bytes): the length of the first IP packet, only included if a MX PDU contains multiple IP packets, i.e. concatenation.
- Fragmentation Control (FC) (e.g. 1 Byte): to provide necessary information for re-assembly, only needed if a MX PDU carries fragments, i.e. fragmentation.

Figure 3 shows the MX trailer format with all the fields present. The MX flags are always encoded in the last octet of the MX Trailer at the end of a MX PDU. Hence, the Receiver SHOULD first decode the MX flags field to determine the length of the MX trailer, and then decode each MX field accordingly.

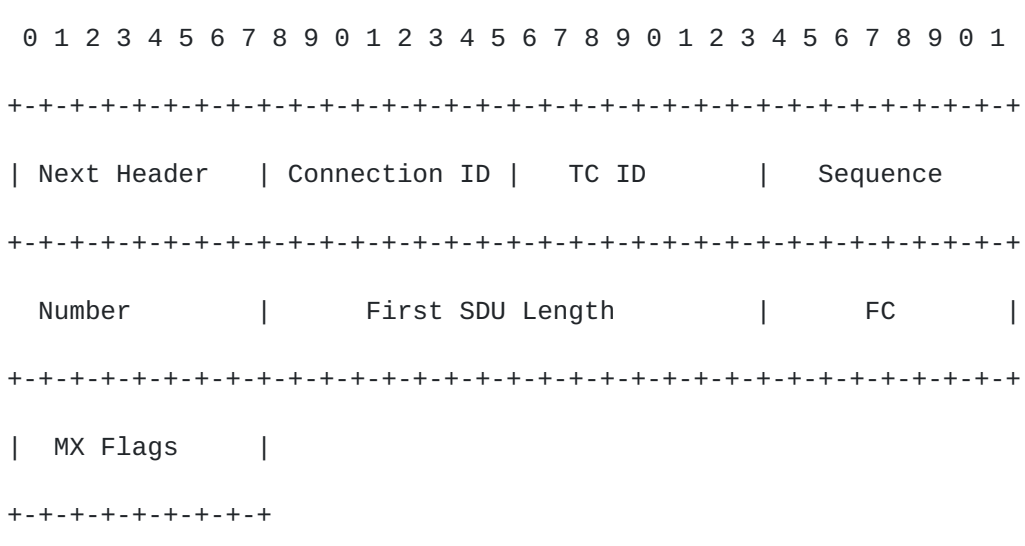


Figure 3: MX Trailer Format

Moreover, the following fields of the IP header of the MX PDU are changed as follows:

o Protocol Type: "114" to indicate that the presence of MX trailer (i.e. the trailer based MAMS u-plane protocol is a "0-hop" protocol, not subject to IP routing)
o IP length: add the length of "MX Trailer" to the length of the original IP packet
o IP checksum: recalculate after changing "Protocol Type" and "IP Length"

The MX u-plane protocol can support multiple Anchor connections simultaneously, each of which is uniquely identified by Connection ID. It can also support multiple traffic classes per connection, each of which is identified by Traffic Class ID.

Moreover, the MX trailer format MAY be negotiated dynamically between NCM and CCM. For example, NCM can send a control message to indicate which of the above fields SHOULD be included for individual delivery connection, on downlink and uplink, respectively.

4.2.2 MX Fragmentation

The Trailer-based MX Convergence Layer SHOULD support MX fragmentation if a delivery connection has a smaller maximum transmission unit (MTU) than the original IP packet (MX SDU), and IP fragmentation is not supported or enabled on the connection. The MX fragmentation procedure is similar to IP fragmentation [RFC791] in principle, but with the following two differences for less overhead:

o The fragment offset unit is fragment not eight-bye blocks.
o The maximum number of fragments per MX SDU is 2^7 (=128)

The Fragmentation Control (FC) field in the MX Trailer contains the following bits:

o Bit #7: a More Fragment (MF) flag to indicate if the fragment is the last one (0) or not (1)
o Bit #0~#6: Fragment Offset (in units of fragments) to specify the offset of a particular fragment relative to the beginning of the MX SDU

A MX PDU carries a whole MX 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 MX PDU contains a fragment of the MX SDU.

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

To fragment a long MX SDU, the MADP transmitter creates two MX PDUs and copies the contents of the IP header fields from the long MX PDU into the IP header of both MX PDUs. The length field in the IP header of MX PDU SHOULD be changed to the length of the MX PDU, and the protocol type SHOULD be changed to "114", indicating the presence of the MX trailer.

The data of the long MX SDU is divided into two portions based on the MTU size of the delivery connection. The first portion of the data is placed in the first MX PDU. The MF flag is set to 1, and the FO field is set to 0. The second portion of the data is placed in the second MX PDU. The MF flag is set to 0, and the FO field is set to 1. This procedure can be generalized for an n-way split, rather than the two-way split described.

To assemble the fragments of a MX SDU, the MADP receiver combines MX PDUs that all have the same MX Sequence Number (in the trailer). 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 MX trailer. The first fragment will have the Fragment Offset zero, and the last fragment will have the More-Fragments flag reset to zero.

4.2.3 MX Concatenation

The Trailer-based MX Convergence Layer MAY support MX concatenation if a delivery connection has a larger maximum transmission unit (MTU) than the original IP packet (MX SDU).

The First SDU Length (FSL) field SHOULD be present in the MX Trailer if a MX PDU contains multiple MX SDUs, i.e. concatenation, and it indicates the length of the first MX SDU in the PDU.

Only the MX SDUs with the same client address, the same anchor connection and the same Traffic Class MAY be concatenated.

To concatenate two or more MX SDUs, the MADP transmitter creates one MX PDU and copies the contents of the IP header field from the first MX SDU into the IP header of the MX PDU. The data of the first MX SDU is placed in the first portion of the data of the MX PDU. The whole second MX SDU is then placed in the second portion of the data of the MX PDU (Figure 4). The procedure continues till the MX PDU size reaches the MTU of the delivery connection.

To disaggregate a MX PDU, the MADP receiver first obtains the length of the first MX SDU from the First SDU Length field in the trailer, and decodes the first MX SDU. The MADP receiver then obtains the length of the second MX SDU based on the length field in the second MX SDU IP header, and decodes the second MX SDU. The procedure continues till no byte is left in the MX PDU.

If a MX PDU contains multiple SDUs, the SN field in the MX trailer is for the last MX SDU, and the SN of other SDU 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 MX PDU contains 3 SDUs (IP packets), then the SN is 4, 5, and 6 for the first, second, and the last IP packet in the PDU, respectively.

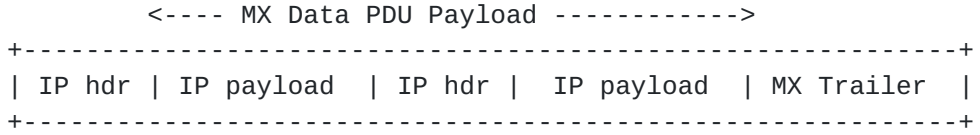
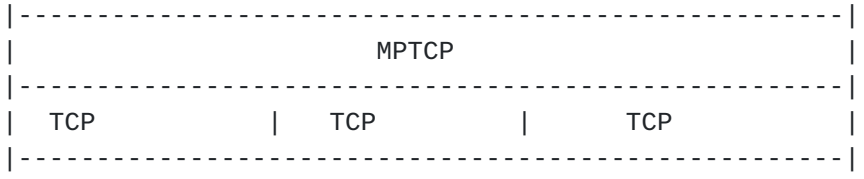


Figure 4: MX PDU Format with Concatenation

4.3 MPTCP-based MX Convergence Layer

Figure 5 shows the MAMS u-plane protocol stack based on MPTCP. Here, MPTCP is reused as the "MX Convergence sub-layer" protocol. Multiple access networks are combined into a single MPTCP connection. Hence, no new u-plane protocol or PDU format is needed in this case.



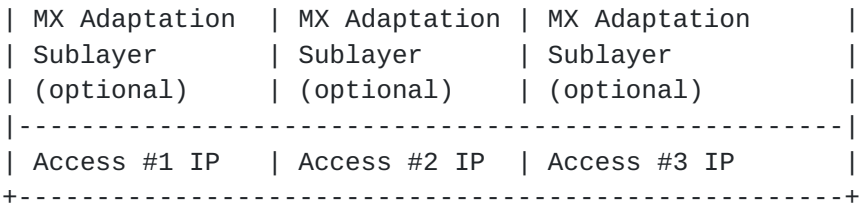


Figure 5: MAMS U-plane Protocol Stack with MPTCP as MX Convergence Layer

If NCM determines that N-MADP is to be instantiated with MPTCP as the MX Convergence Protocol, it exchanges the support of MPTCP capability in the discovery and capability exchange procedures [MAMS_CP]. MPTCP proxy protocols [MPProxy] [MPPlain] SHOULD be used to manage traffic steering and aggregation over multiple delivery connections.

4.4 GRE as MX Convergence Layer

Figure 6 shows the MAMS u-plane protocol stack based on GRE (Generic Routing Encapsulation) [GRE2784]. Here, GRE is reused as the "MX Convergence sub-layer" protocol. Multiple access networks are combined into a single GRE connection. Hence, no new u-plane protocol or PDU format is needed in this case.

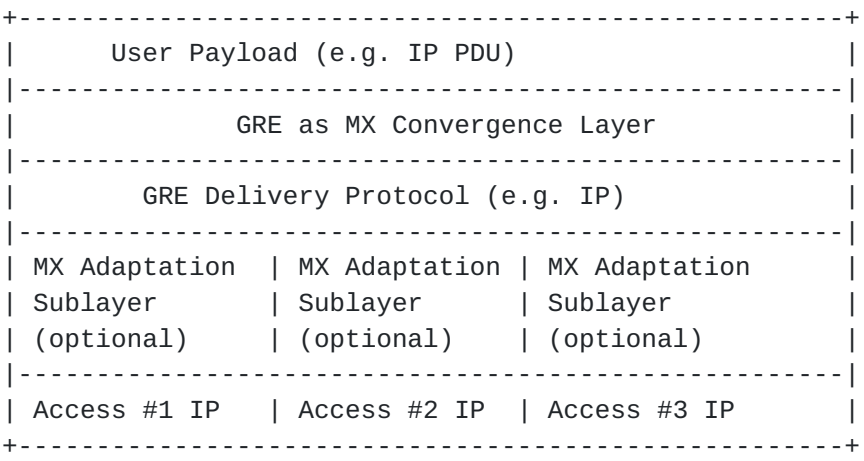


Figure 6: MAMS U-plane Protocol Stack with GRE as MX Convergence Layer

If NCM determines that N-MADP is to be instantiated with GRE as the MX Convergence Protocol, it exchanges the support of GRE capability in the discovery and capability exchange procedures [MAMS_CP].

5.4.1 Transmitter Procedures

Transmitter is the N-MADP or C-MADP instance, instantiated with GRE as the convergence protocol that transmits the GRE packets. The Transmitter receives the User Payload (e.g. IP PDU), encapsulates it with a GRE header and Delivery Protocol (e.g. IP) header to generate the GRE Convergence PDU.

When IP is used as the GRE delivery protocol, the IP header information (e.g. IP address) can be created using the IP header of the user payload or a virtual IP address. The "Protocol Type" field of the delivery header is set to 47 (or 0X2F)[IANA].

The GRE header fields are set as specified below

- If the transmitter is a C-MADP instance, then set the LSB 16 bits to the value of Connection ID for the Anchor Connection associated with the user payload or set to 0xFFFF, if no Anchor Connection ID needs to be specified.
- All other fields in the GRE header including the remaining bits in the key field are set per [GRE_2784][GRE_2890].

5.4.2 Receiver Procedures

Receiver is the N-MADP or C-MADP instance, instantiated with GRE as the convergence protocol, that receives the GRE packets. The receiver processes the received packets per the GRE procedures [GRE_2784, GRE_2890] and retrieves the GRE header.

- If the Receiver is an N-MADP instance,
    - o Unless the LSB 16 Bits of the Key field are 0xFFFF, they are interpreted as the Connection ID of Anchor Connection for the user payload. This is used to identify the network path over which the User Payload (GRE Payload) is to be transmitted.
- All other fields in the GRE header, including the remaining bits in the Key field, are processed per [GRE_2784][GRE_2890].

The GRE Convergence PDU is passed onto the MX Adaptation Layer (if present) before delivery over one of the network paths.

5.5 Co-existence of MX Adaptation and MX Convergence Sublayers

MAMS u-plane protocols support multiple combinations and instances of user plane protocols to be used in the MX Adaptation and the Convergence sublayers.

For example, one instance of the MX Convergence Layer can be MPTCP Proxy [MPProxy] [MPPlain] and another instance can be Trailer-based.

The MX Adaptation for each can be either UDP tunnel or IPsec. IPSec may be set up for network paths considered as untrusted by the operator, to protect the TCP subflow between client and MPTCP proxy traversing that network path.

Each of the instances of MAMS user plane, i.e. combination of MX Convergence and MX Adaptation layer protocols, can coexist simultaneously and independently handle different traffic types.

6 MX Convergence Control

A UDP connection may be configured between C-MADP and N-MADP to exchange control messages for keep-alive or path quality estimation. The N-MADP end-point IP address and UDP port number of the UDP connection is used to identify MX control PDU. Figure 7 shows the MX control PDU format with the following fields:

o Type (1 Byte): the type of the MX control message
  + 0: Keep-Alive
  + 1: Probe REQ/ACK
  + Others: reserved
o CID (1 Byte): the connection ID of the delivery connection for sending out the MX control message
o MX Control Message (variable): the payload of the MX control message

Figure 8 shows the MX convergence control protocol stack, and MX control PDU goes through the MX adaptation sublayer the same way as MX data PDU.

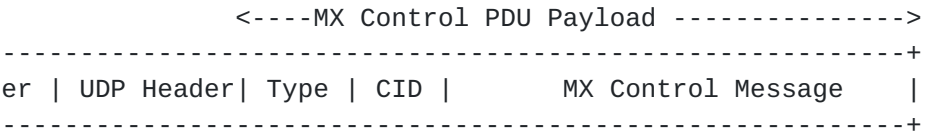


Figure 7: MX Control PDU Format

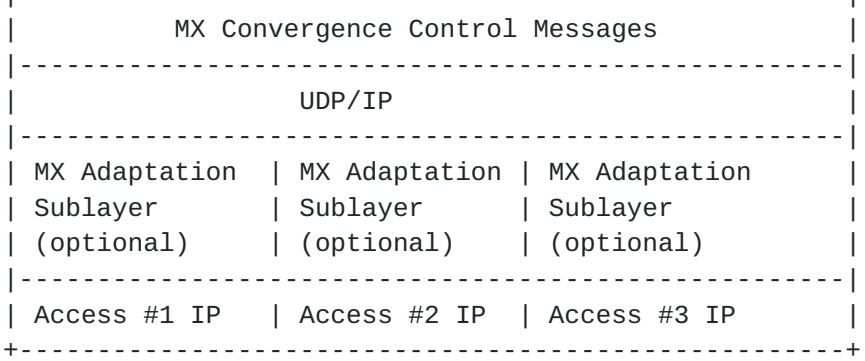


Figure 8: MX Convergence Control Protocol Stack

6.1 Keep-Alive Message

The "Type" field is set to 0 for Keep-Alive messages. C-MADP may send out Keep-Alive message periodically over one or multiple delivery connections, especially if UDP tunneling is used as the adaptation method for the delivery connection with a NAT function on the path.

A Keep-Alive message is 2 bytes long, and consists of the following fields:

- Keep-Alive Sequence Number (2 Bytes): the sequence number of the keep-alive message

6.2 Probe REQ/ACK Message

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

N-MADP may send out the Probe REQ message for path quality estimation. In response, C-MADP may send back the Probe ACK message.

A Probe REQ message consists of the following fields:

- Probing Sequence Number (2 Bytes): the sequence number of the Probe REQ message
- Probing Flag (1 Byte):
  - Bit #0: a Probe ACK flag to indicate if the Probe ACK message is expected (1) or not (0);
  - Bit #1: a Probe Type flag to indicate if the Probe REQ/ACK message is sent during the initialization phase (0) when the network path is not included for transmission of user data or the active phase (1) when the network path is included for transmission of user data;
  - Bit #2~7: reserved
- Padding (variable)

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

C-MADP SHOULD send out the Probe ACK message in response to a Probe REQ message with the Probe ACK flag set to "1".

A Probe ACK message is 3 bytes long, and consists of the following fields:

- Probing Acknowledgement Number (2 Bytes): the sequence number of the corresponding Probe REQ message

7 Security Considerations

User data in MAMS framework rely on the security of the underlying network transport paths. When this cannot be assumed, NCM configures use of appropriate protocols for security, e.g. IPsec [RFC4301] [RFC3948], DTLS [RFC6347].

8 IANA Considerations

TBD

9 Contributing Authors

The editors gratefully acknowledge the following additional contributors in alphabetical order: Salil Agarwal/Nokia, Hema Pentakota/Nokia.

10 References

10.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI10.17487/RFC4301, December 2005, <http://www.rfc-editor.org/info/rfc4301>.

10.2 Informative References

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012,<http://www.rfc-editor.org/info/rfc6347>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <http://www.rfc-editor.org/info/rfc7296>.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, DOI 10.17487/RFC3948, January 2005,<http://www.rfc-editor.org/info/rfc3948>.

[MPProxy] X. Wei, C. Xiong, and E. Lopez, "MPTCP proxy mechanisms", https://tools.ietf.org/html/draft-wei-mptcp-proxy-mechanism-02

[MPPlain] M. Boucadair et al, "An MPTCP Option for Network-Assisted MPTCP", https://www.ietf.org/id/draft-boucadair-mptcp-plain-mode-09.txt

[MAMS] S. Kanugovi, S. Vasudevan, F. Baboescu, and J. Zhu, "Multiple Access Management Protocol", https://tools.ietf.org/html/draft-kanugovi-intarea-mams-protocol-03

[MAMS_CP] S. Kanugovi, et al., "Control Plane Protocols and Procedures for Multiple Access Management Services"

[GRE2784] D. Farinacci, et al., "Generic Routing Encapsulation (GRE)", RFC 2784 March 2000, <http://www.rfc-editor.org/info/rfc2784>.

[GRE2890] G. Dommety, "Key and Sequence Number Extensions to GRE", RFC 2890 September 2000, <http://www.rfc-editor.org/info/rfc2890>.

[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

Authors' Addresses

Jing Zhu

Intel

Email: jing.z.zhu@intel.com

SungHoon Seo

Korea Telecom

Email: sh.seo@kt.com

Satish Kanugovi

Nokia

Email: satish.k@nokia.com

Shuping Peng

Huawei

Email: pengshuping@huawei.com