Telecommunications and Information Exchange Between Systems

ISO/IEC JTC 1/SC 6

Document Number:	N13890
Date:	2009-03-04
Replaces:	
Document Type:	Text for FCD ballot
Document Title:	Text for 2nd FCD ballot, ISO/IEC 16512-3 RMCP-3 (X.603.2)
Document Source:	SC 6/WG 7 Convenor
Project Number:	
Document Status:	SC 6 NBs are requested to ballot on e-balloting system of SC 6 website (www.iso.org/jtc1/sc6) no later than 2009-07-04.
Action ID:	LB
Due Date:	2009-07-04
No. of Pages:	92

ISO/IEC JTC1/SC6 Secretariat Ms. Jooran Lee, KSA (on behalf of KATS)

Korea Technology Center #701-7 Yeoksam-dong, Gangnam-gu, Seoul, 135-513, Republic of Korea ;

Telephone: +82 2 6009 4808 ; Facsimile: +82 2 6009 4819 ; Email : jooran@kisi.or.kr

INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2009-2012

STUDY GROUP 11

TD 95 Rev.1 (GEN/11)

English only

Original: English

Question(s): 15/11 Geneva, 19-23 January 2009

TEMPORARY DOCUMENT

Source: Editors

Title: Revised text of Draft Recommendation ITU-T X.603.2 | ISO/IEC 16512-3 RMCP-3

[Editor's Note] This is a revised text of Draft Recommendation ITU-T X.603.2 | ISO/IEC 16512-3 (RMCP-3) from the joint meeting of ITU-T Q.15/11 and JTC1/SC6/WG7, Geneva, 19-23 January 2009.

One contribution (C64) was submitted to this meeting and was reviewed.

During the review and comments through e-mail from UK, some sub-control data was found missing and is added during the drafting session.

Major modifications through contribution C64 are as follows:

- ***Aligned definition with the X.603.1(2007)/Amd.1 document and revised definition to its current meaning.***
- ***Removed "service administrator" as much as possible, since the protocol should be operated automatically without "service administer" intervention, but some part has been left out for cases as in initial configuration of RMCP-3 network.***
- ***Changed the style of Section 8. Message and Section 9. Parameter to the style to align with the X.603.1(2007)/Amd.1.***
- ***Modified AUTH control data to resolve the same problem defined in defect report 9 of X.603.1(2007)***
- ***Modified Data profile control data to resolve the same problem defined in defect report 13 of X.603.1(2007)***

Additional sub-control data that are added during discussions are SI_DELAY, SI_SND_BW, SI_SND_PACKET, SI_SND_BYTES, SI_RCV_BW, SI_RCV_PACKET, and SI_RCV_BYTES.

This document will be sent to the ISO/IEC JTC 1/SC 6 secretariat for the processing of the 2nd FCD ballot according to relevant resolution of ISO/IEC JTC 1 SC 6 meeting (Montreux, November 2008).

Contact: Shin-Gak Kang
ETRI
Korea
Tel: +82-42-860-6117
Fax: +82-42-861-5404
Email: sgkang@etri.re.kr

Contact: Sung Hei Kim
ETRI
Korea
Tel: +82-42-860-4915
Fax: +82-42-861-5404
Email: shkim@etri.re.kr

Summary

This Recommendation | International Standard describes an application-layer relayed multicast protocol that operates over the IP-based network in which the IP multicast is not fully deployed. This protocol constructs a multicast tree for delivering data from multiple senders to multiple receivers. The relayed multicast tree consists of multicast agent and session manager; the multicast agent relays many-to-many data, whereas the session manager manages the RMCP-3 service session. This document specifies the functions and the procedures of the multicast agent and session manager. RMCP can support applications requiring many-to-many data delivery capability; examples of such applications are multimedia conference, panel discussion, and network gaming.

CONTENTS

Introduction

This Recommendation | International Standard specifies the Relayed Multicast Protocol (RMCP) part 3 for the construction of the relayed multicast tree to support many-to-many group services. Through the realization of the RMCP-3 protocol, it is possible to support many-to-many data delivery capability in the IP-based network environment without full deployment of the IP multicast function. RMCP-3 protocol provides many-to-many multicast service in unicast and multicast network environment.

INTERNATIONAL STANDARD 16512-2
ITU-T RECOMMENDATION X.603.2

Information technology – Relayed multicast protocol: Specification for N-plex group applications

1 Scope

This Recommendation | International Standard describes the RMCP-3 protocol as an application-level protocol that realizes and supports relayed multicast data transport capability for N-plex group applications. The RMCP topology and service scenario described in this specification follows the definition of the RMCP framework without any modification.

2 Normative references

The following ITU-T Recommendations and International Standards contain provisions which, through references in the text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision; users of this Recommendation | International Standard are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations | International Standards and other references listed below. IEC and ISO members maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU-T maintains a list of currently valid ITU-T documents.

- ITU-T Recommendation X.603 (2004) | ISO/IEC 16512-1: 2004, Information technology – Relayed Multicast Protocol: Framework
- ITU-T Recommendation X.603.1 (2007) | ISO/IEC 16512-2, Information technology – Relayed Multicast Protocol Part 2: Specification for simplex group applications
- IETF RFC 2104 (1997), HMAC: Keyed-Hashing for Message Authentication
- IETF RFC 1321 (1992), The MD5 Message-Digest Algorithm

3 Definitions

3.1 Terms defined elsewhere

This Recommendation| International Standard uses the following terms defined elsewhere:

3.1.1 Child multicast agent (CMA): Next downstream MA in the RMCP data delivery path.

3.1.2 IP multicast [ITU-T X.603]: Realizes a multicast scheme in the IP network with the help of multiple several multicast-enabled IP routers.

3.1.3 Multicast [ITU-T X.603]: A data delivery scheme where the same data unit is transmitted from a single source to multiple destinations in a single invocation of service.

3.1.4 Multicast agent (MA) [ITU-T X.603]: Intermediate node which relays group application data.

3.1.5 N-plex [ITU-T X.603]: Wherein anyone can send something, and, if someone does so, all others may receive it.

3.1.6 Parent multicast agent (PMA) [ITU-T X.603]: Next upstream MA in the RMCP data delivery path.

3.1.7 Relayed multicast [ITU-T X.603]: A multicast data delivery scheme that can be used in unicast environments, which is based on the intermediate Multicast Agents to relay multicast data from the media server to media players over a tree hierarchy.

3.1.8 Relayed Multicast Protocol (RMCP) [ITU-T X.603]: The control protocol used for realizing and managing the relayed multicast data transport.

3.1.9 Receiver multicast agent (RMA) [ITU-T X.603]: The MA attached to the receiving application in the same system or local network.

3.1.10 RMCP session [ITU-T X.603]: A set of MAs which configures the data delivery path using RMCP.

3.1.11 Sender multicast agent (SMA) [ITU-T X.603]: MA attached to the sender in the same system or local network.

3.1.12 Session manager (SM) [ITU-T X.603]: An RMCP entity that is responsible for the overall RMCP operations; it may be located in the same system as the sending application or located separately from the sending application.

3.1.13 Simplex [ITU-T X.603]: Wherein only one sender is send only and all others are receive-only.

3.2 Terms defined in this Recommendation

For the purposes of this Recommendation | International Standard, the following definitions apply.

3.2.1 Core domain: Top-level domain consisting of group of CoreMAs and SM.

3.2.2 Core multicast agent (CoreMA): Group of MAs that configures RMCP-3 core domain.

3.2.3 CoreRing: A ring topology consisting of CoreMAs and SM in the core domain.

3.2.4 Edge domain: Bottom-level domain consisting of group of EdgeMAs.

3.2.5 Edge multicast agent (EdgeMA): MAs that configures RMCP-3 edge domain.

3.2.6 EdgeTree: Tree topology consisting of EdgeMAs and single CoreMA as a root node in the edge domain.

3.2.7 Head multicast agent (HMA): Head of the MAs inside the local multicast area which relays multicast data to the local multicast area.

3.2.8 RMCP-3 Hybrid Tree: Mixed topology with CoreRing and EdgeTree.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

4.1 Abbreviations of RMCP-3 messages

HANNOUNCE	HMA announce
HHB	Horizontal heartbeat
HLEAVE	HMA leave
HSOLICIT	HMA solicit
LEAVANS	Leave answer
LEAVREQ	Leave request
PPROBANS	Parent probe answer
PPROBREQ	Parent probe request
RELANS	Relay answer
RELREQ	Relay request
STANS	Status report answer
STREQ	Status report request
SUBSANS	Subscription answer
SUBSREQ	Subscription request

TERMANS	Termination answer
TERMREQ	Termination request
VHB	Vertical heartbeat

4.2 Abbreviations of non-messages

CoreMA	Core Multicast Agent
CoreRing	RMCP-3 Core Ring
CMA	Child Multicast Agent
EdgeMA	Edge Multicast Agent
EdgeTree	RMCP-3 Edge Tree
HMA	Head Multicast Agent
KO	Kick-Out
MA	Multicast Agent
MAID	Multicast Agent Identification
PMA	Parent Multicast Agent
RMCP	Relayed Multicast Protocol
SID	Session Identification
SM	Session Manager
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

5 Conventions

<None>

6 Overview

RMCP-3 is an application-level multicast transport protocol for providing efficient N-plex group communication services over IP-network environment without full IP multicast deployment. This clause gives an overview of the RMCP-3, particularly, RMCP-3 services, entities, protocol block, N-to-N data delivery model, and messages.

6.1 RMCP-3 services

RMCP-3 is an application-level multicast protocol that supports various N-to-N (N-plex) group communication services in a unicast based IP network. To support the N-to-N group communication services, RMCP-3 uses the relayed multicast mechanism. The entities of RMCP-3 protocol configure an efficient data delivery path for N-to-N group communications. RMCP-3 entities forward group data to each participant along the constructed delivery path.

RMCP-3 can support various application services that require N-to-N group communications such as video conference and panel discussion. There can be two types of N-to-N group communications, one such type would be the video conference service in which every participant can send and receive data simultaneously. The other type would be the panel discussion service in which only few participants are able to send data and the rest of the participant can only receive data. MA can be implemented in various ways; such as a dedicated server, as a set-top box, or as a part of client application.

Figure 1 shows a typical service model of RMCP-3 for supporting N-to-N group communications service. The RMCP-3 protocol can be used in a unicast-based Internet environment where multicast capability is partially deployed. In RMCP-3, the area where multicast capability is partially deployed is called a local multicast region. One such example of local multicast region is a campus network with multicast capability deployed. The network administrator installs the multicast capability in all of the network entities constructing the campus network.

As shown in Figure 1, the RMCP-3 protocol can provide N-to-N group communications service in both unicast and multicast region. The RMCP-3 protocol utilizes the multicast capability inside the local multicast region and, also, provides multicast capabilities to entities in the unicast region.

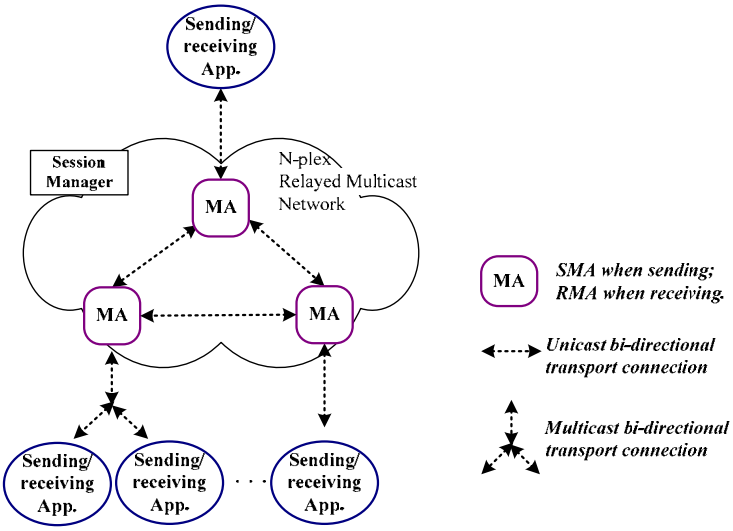

Figure 1 – RMCP-3 service model

The main entities of the RMCP-3 protocol are session manager (SM) and multicast agent (MA). The SM manages the multicast delivery path and multicast session. The MA is an intermediate node that provides data delivery capability.

The following features support the N-plex group communications.

a) RMCP-3 constructs a logical control tree by using one or more MAs; the control tree supports the transmission of data in a reliable or real-time manner.

b) RMCP-3 has the capability of selecting optimal peers; because the RMCP-3 control tree consists of logical links between MAs, and each MA configures a logical links based on the selected peers. The selection of optimal peers can be based on various metrics; example of such metrics includes hop count, delay, and/or bandwidth.

c) RMCP-3 supports pure IP multicast, NAT/Firewall, and different versions of IP.

d) RMCP-3 allows participants to join or leave at any time during a session.

e) RMCP-3 manages the participants of a session; the capability of managing the session includes membership monitoring and expulsion of members.

f) RMCP-3 provides an auto-configuration mechanism for the N-plex group communications path.

g) RMCP-3 provides network fault detection and service recovery.

h) RMCP-3 avoids the one-point failure problem; RMCP-3 uses decentralized administration.

i) RMCP-3 has various ways of managing the session; e.g., tightly or loosely.

6.2 RMCP-3 entities

This clause describes the roles of each RMCP-3 entity. The RMCP-3 consists of session manager (SM) and multicast agent (MA). The RMCP-3 entities follow the same definition as defined in [ITU-T X.603].

As shown in Figure 1, RMCP-3 configures the relayed multicast data delivery path for many-to-many multicast using the following configuration:

a) One SM per session;

b) One or more MAs (MA includes both CoreMA and EdgeMA);

c) One or more sending and receiving applications.

SM supports the following functions:

a) Session initiation;

b) Session termination;

c) Membership management;

d) Monitoring session status.

MA is required to support capabilities in both sending and receiving group data. MA configures the data delivery path for N-to-N group data. The followings are the functionalities supported by MA:

a) Session subscription;

b) Session join;

c) Session leave;

d) Session management;

e) Reporting session status;

f) Data delivery.

6.3 RMCP-3 protocol block

RMCP-3 uses two different types of protocol blocks. The first block is used for controlling RMCP-3 session, and the second block is used for delivering group data. Since the SM is used to control the RMCP-3 session, SM only has a control module. On the other hand, since the MA is used for both control and data delivery, it consists of two modules which are control module and data module.

Figure 2 shows the three types of path and interfaces that are used in RMCP-3.

– RMCP-3 control message path between SM and MA and between MAs;

– Data path between MA data modules;

– Local MA interfaces inside the MA; that is, between the control module and the data module.

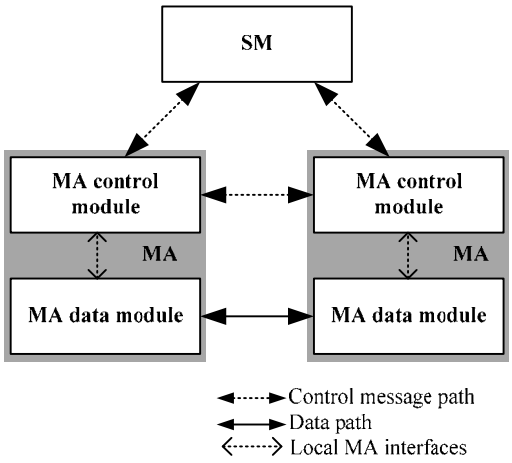

Figure 2 – Three types of interfaces in RMCP-3

RMCP-3 needs to use reliable transport protocol in exchanging RMCP-3 protocol message to construct a robust and reliable multicast session. Thus, RMCP-3 uses TCP in transmitting control messages.

Figure 3 shows the protocol stack for SM. RMCP-3 SM would need to use the TCP for reliable delivery of the control message.

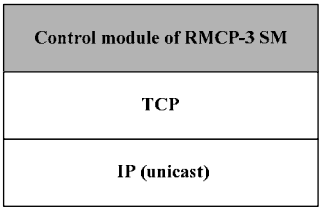

Figure 3 – Protocol stack for RMCP-3 SM

For the data delivery model, RMCP-3 configures a RMCP-3 Hybrid Tree to deliver many-to-many group data; along the configured RMCP-3 Hybrid Tree, each MA can send and receive group data. Figure 4 shows the protocol stack for MA. As mentioned earlier, the MA's protocol stack consists of control module and data module. The control module is used for constructing the RMCP-3 Hybrid Tree and the data module for delivering N-to-N group data.

The MA control module configures the RMCP-3 Hybrid Tree according to the MA controller's request. The RMCP-3 Hybrid Tree is configured through exchanges of control messages between SM and other MAs. To exchange reliable control messages, MA's control module uses a reliable transport protocol (i.e., TCP) in the unicast area and UDP in the multicast-enabled area.

The MA data module delivers N-to-N group data along the constructed RMCP-3 Hybrid Tree. The characteristics of the data delivery channel may vary by application data type. For example, real-time data delivery channel is required for real-time application services, and reliable data delivery channel is required for reliable application services. On the other hand, RMCP-3 allows the service implementer to choose to various transport protocols for delivering application data to support various application services.

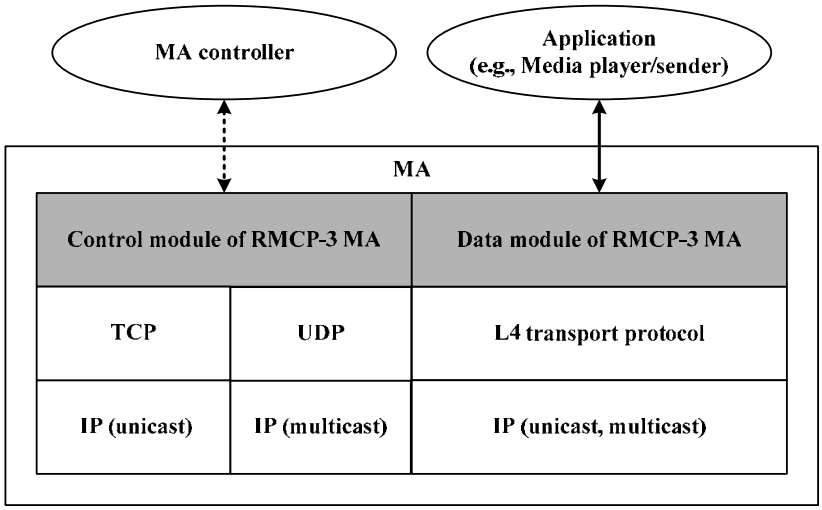

Figure 4 – Protocol stack for RMCP-3 MA

6.4 RMCP-3 control model

The RMCP-3 protocol configures a RMCP-3 Hybrid Tree for control connection that is based on the RMCP-2 one-to-many tree. The RMCP-2 tree is suitable for real-time and reliable data deliveries and is robust to control network fault problem with its auto-configuration and self-improvement capability. The RMCP-2 tree can be extended and used in RMCP-3 Hybrid Tree. The RMCP-3 Hybrid Tree uses multiple RMCP-2 one-to-many trees for many-to-many multicast data delivery.

The RMCP-3 Hybrid Tree has two-level hierarchy. The bottom level is a one-to-many tree which is called the EdgeTree. It is a similar tree that is used in RMCP-2 protocol. The EdgeTree consists of more than zero EdgeMAs with a root node being the CoreMA. The top level is a CoreRing with more than one CoreMAs connected to SM to control RMCP-3 network.

The two-level hierarchy can correspond to the two RMCP-3 domains which is the core domain and the edge domain. The core domain is a top level network with CoreRing. The edge domain is the bottom level with EdgeTrees. The RMCP-3 protocol uses the concept of domain in order to distinguish the control and data flow.

The RMCP-3 Hybrid Tree is controlled by the RMCP-3 session manager (SM). The SM can configure, directly control, and monitor the RMCP-3 Hybrid Tree. The SM has the complete list and the connection status of the CoreMA's of the RMCP-3 session. To control the CoreMAs, SM constructs a CoreRing which is a ring topology made up of CoreMAs and SM. Figure 5 shows an example of a CoreRing control connection.

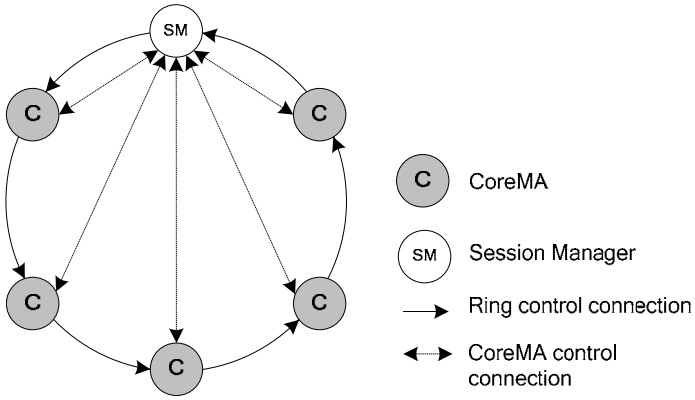

Figure 5 – Control connection in the core domain

The following lists the control connections between the RMCP-3 entities in the core domain:

- One ring connection between SM and one or more CoreMAs;
- Direct connections between SM and CoreMAs.

EdgeTree consists of multiple EdgeMAs with CoreMA as the top node. The SM can directly monitor and control EdgeTree by controlling each MA in the EdgeTree. The CoreMA has the complete list of EdgeMAs in its EdgeTree, which makes it possible to control each EdgeMA in the EdgeTree.

Figure 6 shows the EdgeTree and its connectivity in exchanging control messages.

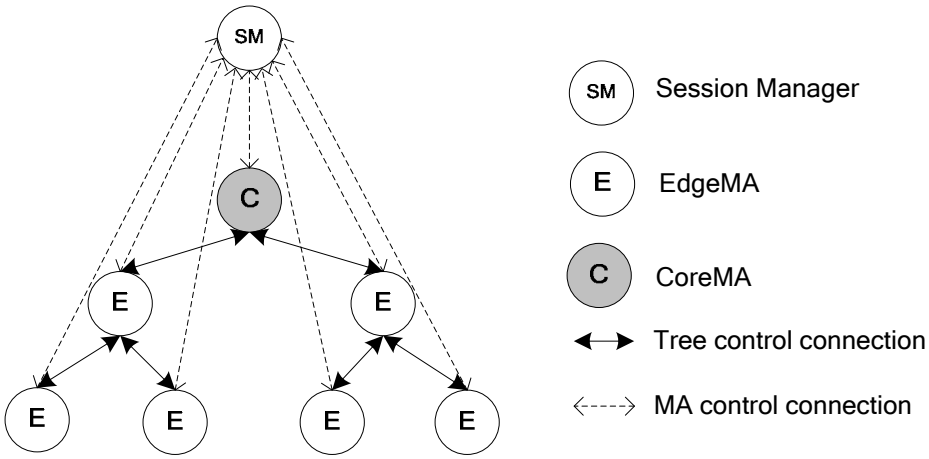

Figure 6 – Control connection in the edge domain

EdgeTree consists of CoreMA and zero or more EdgeMAs. The following are the control connections that exist in the EdgeTree:

- Connections between MAs forming EdgeTree;
- Direct connection between SM and MAs.

The SM and CoreMAs that constitutes the core domain are fixed and dedicated devices that are initially installed by the RMCP-3 service administrator. Those nodes do not participate in the RMCP-3 service, but provide control and data delivery path for applications serviced by EdgeMA. The EdgeMA is a dynamic node that can join and leave multicast session. Thus, the SM and CoreMAs must be constructed before providing RMCP-3 multicast service.

Note, however, that this specification does not specify how the CoreMAs are installed because it is closely related to implementation and design issues defined by the RMCP-3 service administrator at the beginning of the RMCP-3 services. The RMCP-3 service administrator needs to assign adequate number of CoreMAs according to the anticipated size of multicast session and also position the CoreMAs in a suitable part of the network. How the RMCP-3 service administrator construct the RMCP-3 core domain is strictly dependent on the service and is out of scope of this document.

6.5 N-plex data delivery model of RMCP-3

The top-level topology of connection for N-plex data delivery is different from the topology of connection in the RMCP-3 control connection. The top-level topology of the control connection is in form of a ring, i.e., CoreRing. However, the top-level topology of N-plex data delivery connection can be of any structure, such as mesh as shown in Figure 7 or ring as shown in Figure 8. The reason for the difference in top-level topology is that the main object of the RMCP-3 is to make an efficient data delivery topology. Therefore, it does not have to use the same topology as for the control connection. However, for the EdgeTree, the control path and data path is equivalent, since it is impossible to define different path for the constant changing EdgeTree.

The RMCP-3 defines control mechanism to construct a multicast tree that can be used for n-plex multicast service. The n-plex data delivery model is constructed by the SM which control and manage n-plex multicast session. The SM must make sure that the data delivery model do not form a loop and must prevent the MAs from receiving duplicate packet. The edge domain is in a form of a tree, thus, it would be not be difficult to create a loop-free data delivery path.

The n-plex data delivery model can vary with the type of application that is used and is implementation dependent. Thus, this recommendation does not define a data delivery model. But, the annex gives various example of such n-plex data flow mechanism that can be used with the RMCP-3 protocol.

The basic RMCP-3 data delivery model is shown in Figure 7 and Figure 8. The EdgeMA is the sender and a receiver of n-plex multicast service. If an EdgeMA wants to send data to the RMCP-3 session, it sends the data to its CoreMA and its child MAs (CMAs). The CoreMA forwards the data to other EdgeMAs in its EdgeTree and also to other CoreMAs along the data forwarding path which may have mesh or ring topology in the core domain. The receiving CoreMA would forward the data to its EdgeMA in the EdgeTree.

Figure 7 shows the data delivery model where the SM arranges CoreMAs to have full mesh topology. In this case, the CoreMA only forward data originating from its own edge domain. The CoreMA can distinguish the source of the data through the received port or by examining the packet source.

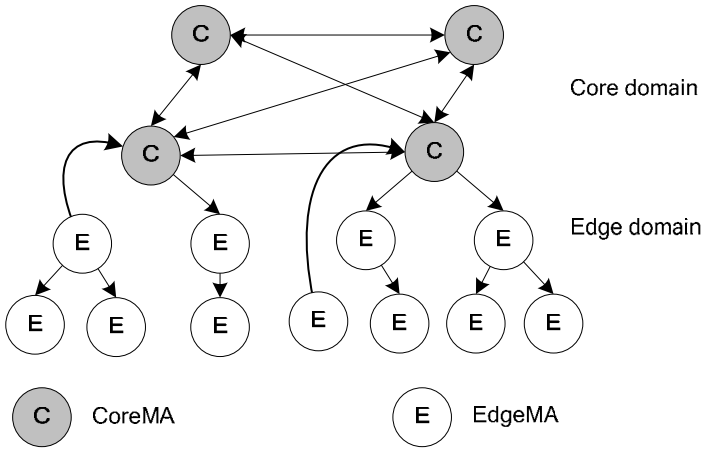

Figure 7 – RMCP-3 data delivery model with mesh-linked core-domain

Figure 8 shows the data delivering model using the CoreRing of the control connection. The CoreMA forwards the data to both direction of the CoreRing, except for the last CoreMA which is connected to the SM. The SM does not receive or relay data packet.

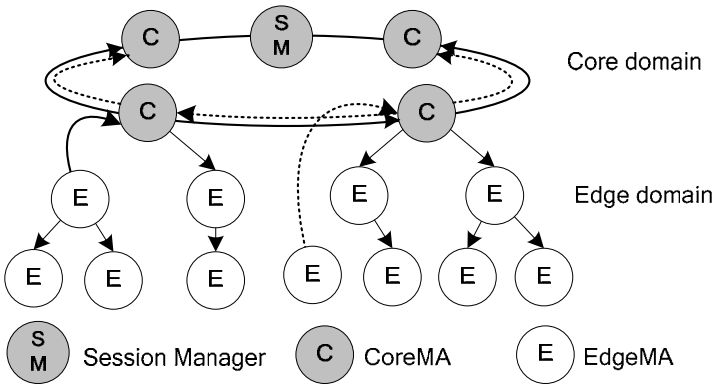

Figure 8 – RMCP-3 data delivery model with ring-linked core-domain

6.6 Types of RMCP-3 messages

Table 1 lists the RMCP-3 messages with its meaning and the operation that is used.

Table 1 – RMCP-3 messages

Messages	Meaning	Operation
SUBSREQ	Subscription request	Session subscription
SUBSANS	Subscription answer	
PPROBREQ	Parent probe request	Neighbor discovery
PPROBANS	Parent probe answer	
RELREQ	Relay request	Data channel control
RELANS	Relay answer	
LEAVREQ	Leave request	Session leave
LEAVANS	Leave answer	
TERMREQ	Termination request	Session termination
TERMANS	Termination answer	
HSOLICIT	HMA solicit	Management for multicast enabled network
HANNOUNCE	HMA announce	
HLEAVE	HMA leave	
VHB	Vertical heartbeat	EdgeTree maintenance
HHB	Horizontal heartbeat	CoreRing maintenance
STREQ	Status report request	Session monitoring
STANS	Status report answer	

7 Protocol operation

This clause gives detailed description of the protocol operation of each RMCP-3 entities which are session manager (SM), Core Multicast Agent (CoreMA), and Edge Multicast Agent (Edge MA). The three entities form a RMCP-3 Hybrid Tree to provide multicast function in non-multicast IP-based network. Figure 9 shows an example model of the RMCP-3 Hybrid Tree that is used thought clause 7 to assist in understanding the operation of the RMCP-3 protocol.

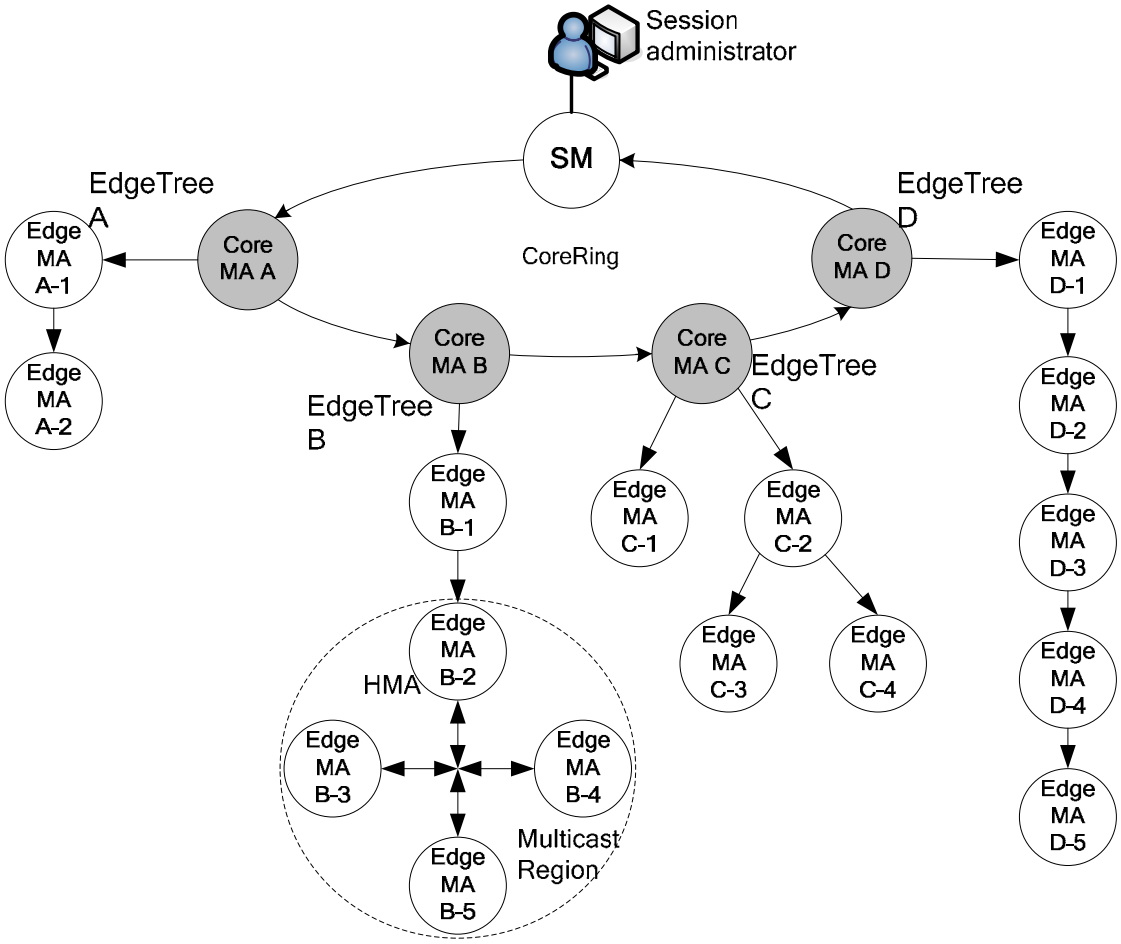

Figure 9 – Example model of RMCP-3 Hybrid Tree

In the example model, RMCP-3 Hybrid Tree consists of four CoreMAs with each forming its own EdgeTree. The CoreMA A is the root of EdgeTree A with two EdgeMAs arranged in a row. The CoreMA B is the root of EdgeTree B with five EdgeMAs. Four of the five EdgeMAs forms a multicast region with EdgeMA B-2 being the Head MA (HMA). The CoreMA C is the root of the EdgeTree C with four EdgeMAs forming a binary tree. CoreMA D is the root of the EdgeTree D with five EdgeMAs arranged in a row. The example model is used through this clause to explain the operation of the RMCP-3 protocol.

7.1 SM's operation

SM (Session Manager) plays a key role in RMCP-3 session management. SM provides capabilities related to session initiation, admission control, session monitoring, membership control, and session termination. The SM manages the RMCP-3 service. The SM functions include neighbor list management, data profile negotiation, and group management. The followings are the details of SM's functions:

- Neighbor list management involves aggregating a set of MAIDs activated in the RMCP-3 session. The neighbor list is required for the EdgeMA subscribing to the RMCP-3 session. Whenever a new EdgeMA subscribes to a session, the neighbor list can be updated;
- Data profile negotiation enables SM to negotiate data profile with new a subscriber; the data profile is defined when the RMCP-3 session is created;
- Group management enables SM to create and manage the RMCP-3 service group. When a SM creates a new RMCP-3 service group, it can impose group characteristics that include membership management (tightly or loosely), the openness of group (opened or closed), and status of member (minimum member or core-member). SM manages the RMCP-3 service group after it is created according to the imposed group characteristics.

7.1.1 **Initiation**

This capability enables the SM to create a new RMCP-3 session. The SM can create a new session by acquiring a session profile which includes session name, media characteristics, and group address. The SM creates a globally unique SID to distinguish the session, and returns SID to the RMCP-3 service provider.

After the successful RMCP-3 session initiation, the RMCP-3 service provider announces the session information and SID. Examples of media used for announcing the RMCP-3 session include web server and e-mail. SM then waits for EdgeMA's subscription request. Upon receiving EdgeMA's subscription request, SM decides whether the subscription request should be accepted or not; the detailed procedure for subscription is described in the following clause.

7.1.2 **Session subscription**

This capability enables SM to allow EdgeMA to join the RMCP-3 session by accepting EdgeMA's subscription request. Figure 10 shows how each EdgeMA can subscribe to the RMCP-3 session. To subscribe to the RMCP-3 session, each EdgeMA sends SUBSREQ message to the SM; the SM then responds with SUBSANS.

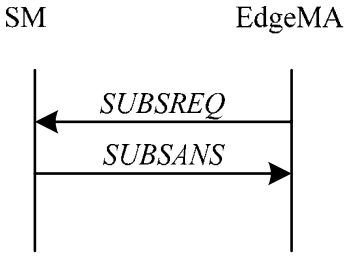

Figure 10 – RMCP-3 session subscription

Subscribing to the RMCP-3 session requires for each EdgeMA to have a correct SID for the session. When the SM receives subscription request message from a new user, SM checks the SID contained in the request message, and subsequently makes a decision as to whether or not to accept the request.

To be identified in the RMCP-3 session, each MA requires a unique MAID. MAID can be proposed by MA and confirmed by SM. In case MAID is not proposed, or if the proposed MAID is duplicated, SM creates a unique MAID for the new MA. MAID can be generated from the IP address and port number of the new MA. The details of creating MAID are described in the clause 9.1.

If the RMCP-3 session requires more specific information on each subscriber, then SUBSREQ is required to include more extended information that includes system and network resources, authentication, affordable number of children, data profile, etc. SM can make decision on the admission based on the information included in SUBSREQ. An example of using extended information is in the case where RMCP-3 allows negotiation for the data channel. In case a data profile is in SUBSREQ, SM checks the data profile; if the data profile resides in the capabilities of the session, then SM can modify the data profile and then sends the result back using SUBSANS. Otherwise, SM can reject the subscription request by sending SUBSANS with an appropriate error code.

If SM allows MA's subscription, SM responds to the requestor with SUBSANS containing the confirmed MAID, bootstrapping information, and other extended information such as negotiated data profile. For the bootstrapping information, the SM may give the requestor a complete list of CoreMAs or partial list of candidate CoreMAs which is implementation and service dependent. If the SM rejects MA's subscription, SM responds to the requestor with SUBSANS containing an appropriate error code.

7.1.3 **Session monitoring**

Session monitoring function is used to monitor each MA; in this clause, MA includes both CoreMA and EdgeMA. SM can provide various level of session management according to the application. For tight session management in the RMCP-3 protocol, RMCP-3 should provide the means of monitoring the status of the session members. In RMCP-3, status information can be acquired individually. If the SM needs status information of a specific MA, it issues STREQ to the MA requesting specific information. The MA responds with STANS containing the requested information.

Figure 11 shows an example procedure for monitoring a specific MA by SM. SM sends STREQ to MA B to request one or more specific status information of MA B. In response, node B sends STANS to SM containing the requested information. The "Report time" in the Figure 11 is the maximum waiting time of SM for STANS. If the STANS message does not arrive within the "Report time", the SM would acknowledge that the MA C is not functioning properly.

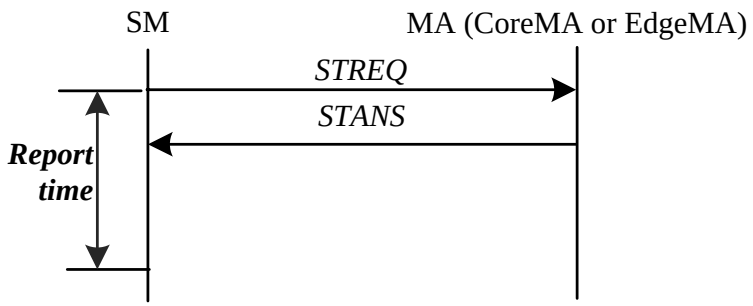

Figure 11 – MA monitoring (status report)

7.1.4 **Membership control**

Through membership control, it is possible to provide managed multicast service. The membership control can be made through session monitoring and EdgeMA controlling. SM can expel a specific EdgeMA using the SM's expulsion function to control the multicast service. When the SM decides to expel a specific EdgeMA, SM sends LEAVREQ with the reason code of KO. Upon receiving LEAVREQ from SM, EdgeMA sends an answer with LEAVANS message, and reports to its neighboring EdgeMAs before leaving the session. The detailed procedure for EdgeMA leaving the session will be described in Clause 7.3.4.1. Figure 12 shows the message flow of expulsion of EdgeMA A-2. The EdgeMA A-2 will need to report to its parent and child through LEAVREQ and LEAVANS message before leaving the RMCP-3 session.

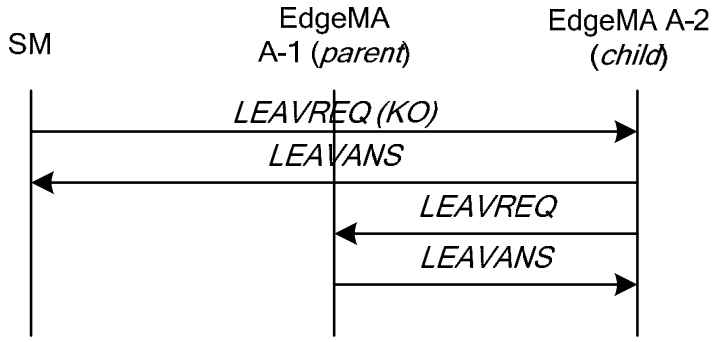

Figure 12 – SM's expulsion of EdgeMA

7.1.5 **Session termination**

The SM can terminate the on-going RMCP-3 session using the session termination function. Figure 13 shows the detailed procedure of the RMCP-3 session termination. To terminate the RMCP-3 session, SM sends TERMREQ to each of the CoreMAs. The CoreMA sends TERMANS to SM and then forwards TERMREQ to its EdgeMAs until it reaches the last EdgeMA of the EdgeTree. Once the MA receives a TERMREQ message, it must leave the session after notifying its PMA and CMA of its leave.

Figure 13 shows the flow for session termination made by SM to all EdgeTree. The CoreMAs in the EdgeTree will receive a TERMREQ message from the SM. In the EdgeTree A, the CoreMA A terminates its EdgeMAs which are EdgeMA A-1 and EdgeMA A-2. The other CoreMAs in each EdgeTree will perform same function as CoreMA A upon receiving TERMREQ from the SM.

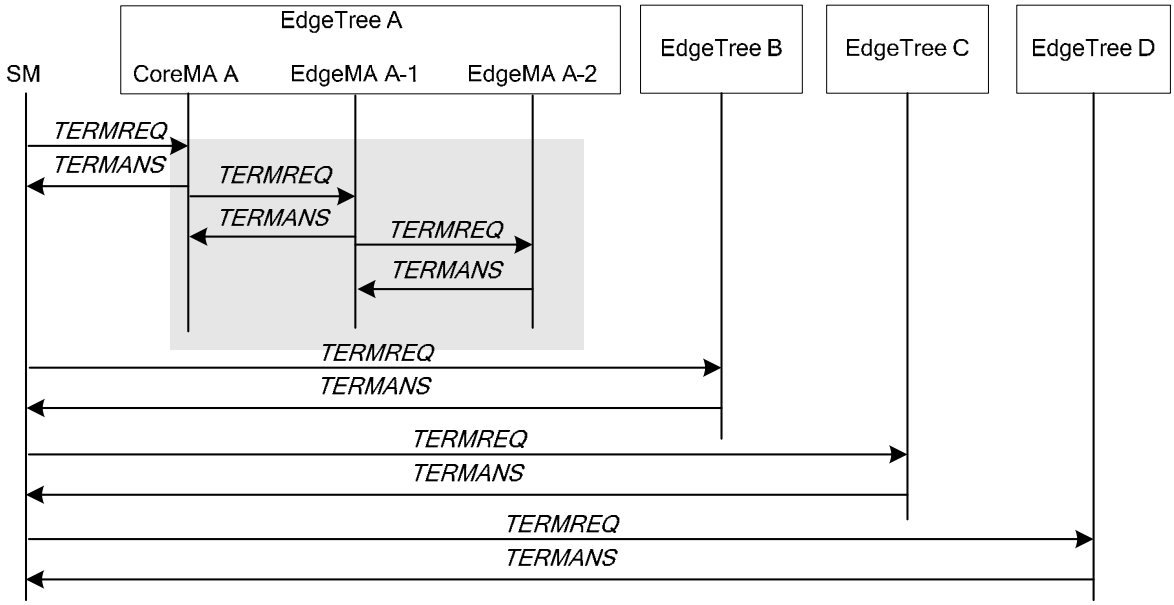

Figure 13 – Session termination issued by SM

7.2 CoreMA's operation

CoreMA is a special-purpose MA that makes RMCP-3 session scalable and robust. RMCP-3 assumes that MA is an end-system with RMCP-3 functionalities. Some examples of end-system include desktop PC, PC server, and set-top box. RMCP-3 allows MA to join and leave RMCP-3 session at any time, thereby making the RMCP-3 session more flexible. Since MA can be configured with the end-system, such capability can cause network faults such as loop and network partitioning. To overcome the resulting network havoc, RMCP-3 defines a highly reliable and manageable CoreMA.

The roles of CoreMA include supporting the followings:

- Scalability of the RMCP-3 session;
- Distributed management of RMCP-3 participants by acting as the root node of EdgeTree (i.e., top node of EdgeTree);
- Bidirectional data delivery.

The detailed operations of CoreMA are described in the following clause.

7.2.1 Initiation

The SM and CoreMAs in the CoreRing are static and are not changed automatically during the RMCP-3 session. It is possible for the CoreMA to terminate during the RMCP-3 session for administrative purpose. The CoreMAs receive and forward data of all sessions which means that the CoreMAs are already joined in all RMCP-3 session. Moreover the RMCP-3 protocol presumes that the SM and CoreMA are fixed during the RMCP-3 session.

The SM can directly monitor and control CoreRing by controlling each CoreMA. Such control can be used to check for the status of CoreMA and/or its workload. For tight control of the core domain, SM must have a complete list of CoreMAs participating in the session.

7.2.2 Support for RMCP-3 join of EdgeMA

CoreMA must support EdgeMA to join the RMCP-3 session. After the EdgeMA subscribes to the RMCP-3 session through SM, it would need to select an appropriate CoreMA to perform joining in pertaining EdgeTree. This process is a part of the neighbor discovery procedure of EdgeMA described in clause 7.3.2. The EdgeMA sends a PPROBREQ message to each CoreMAs in a list which is given by SM in the bootstrap information. The CoreMA returns a PPROBANS message which includes a list of EdgeMAs in its EdgeTree along with other information related to joining in pertaining EdgeTree. Using SYSINFO control data with SI_TREE_MEM sub-control data, CoreMA can give a list of EdgeMAs in its EdgeTree. The received information from the CoreMA, EdgeMA can continue with the neighbor discovery procedure.

The remaining functions of the CoreMA in supporting EdgeMA is equivalent to function of PMA (parent multicast agent) function described in clause 7.3

7.2.3 **Maintenance**

The RMCP-3 protocol has scalability due to the use of CoreMAs. The CoreMA combines multiple EdgeTrees to form a single RMCP-3 Hybrid Tree. To make the RMCP-3 service stable, CoreMA is required to guarantee integrity.

RMCP-3 provides the following functionalities to guarantee the stability of the RMCP-3 session:

- Function to check the aliveness of each logical link that constitutes the CoreRing;
- Function of monitoring the status of each CoreMA;
- Function of notifying SM the status of CoreRing.

7.2.3.1 Horizontal heartbeat

Horizontal heartbeat (HHB) message is used to check the link status of the CoreRing. The HHB message can be used to check the followings status:

- Activity of CoreMA;
- List of all CoreMAs making up CoreRing.

The HHB procedure is started with SM transmitting a HHB message to the adjacent CoreMA of the CoreRing in one direction. The received CoreMA propagates the HHB message to its adjacent CoreMA along the CoreRing. Eventually, the SM will receive the HHB message sent by the last CoreMA, and it will know the status of each CoreMA along the CoreRing. If the SM failed to receive the HHB message, then it would know that the CoreRing is not functioning properly.

Figure 14 shows how the SM can check the status of the CoreRing by relaying a periodic HHB message along the CoreRing. The SM sends periodically HHB message to the adjacent CoreMA, which is CoreMA A as in the example model of Figure 9. The CoreMA A would add its information into the received HHB message and then forwards the changed HHB message to the next CoreMA, which is CoreMA B. The HHB message will eventually be relayed and returned to the SM within a specific time which is the HHB time.

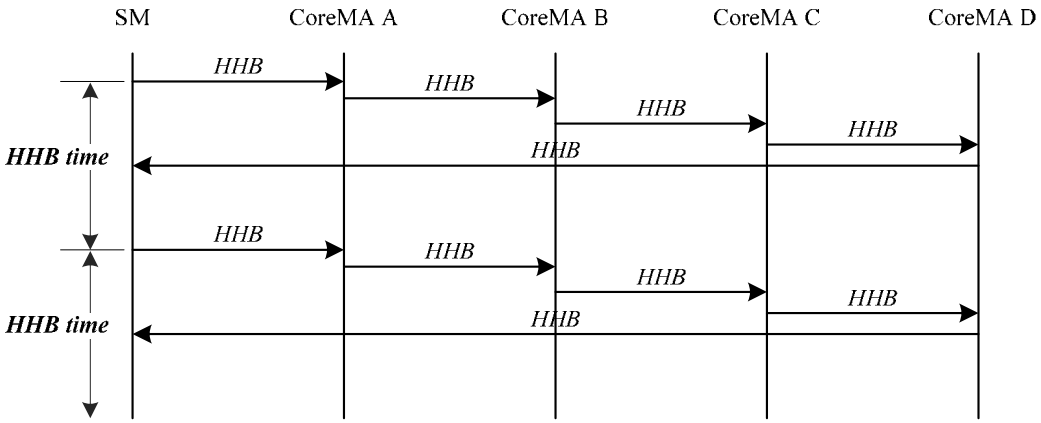

Figure 14 – Successful HHB

SM sends a periodic HHB message at every heartbeat time (HHB time) and expects its return within a specified time (i.e., until HHB time expires). If all the CoreMAs in the CoreRing is functioning well, the SM will have the HHB message returned. The normal HHB message includes the following information:

- List of all CoreMAs in the CoreRing (mandatory);
- System information of each CoreMA; the information may include the number of bytes of the relayed data, and the number of EdgeMA handled by each CoreMA (optional).

If the CoreRing has faults, the periodic HHB message will not be returned to the SM within a specified time (i.e., HHB time). If a CoreMA does not receive any HHB message within the HHB time, it notifies the SM of missing HHB message. Figure 15 shows the flow in which the HHB message does not arrive within a HHB time due to a fault in CoreRing.

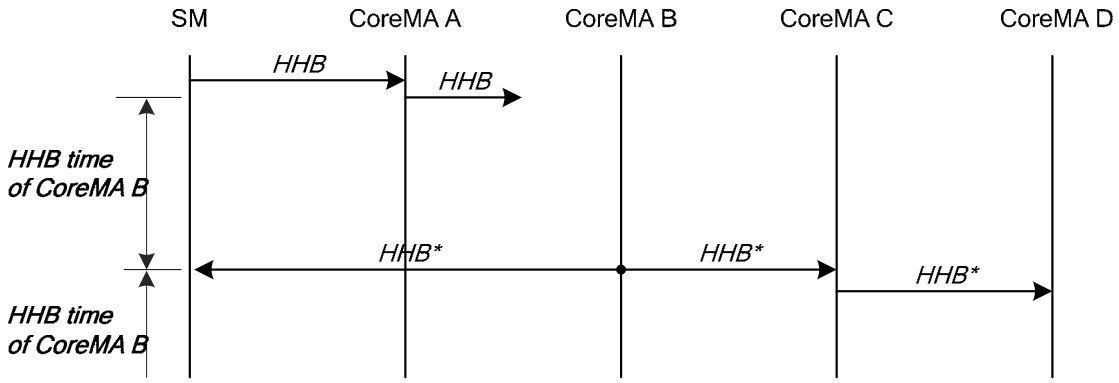

Figure 15 – Unsuccessful HHB delivery

Figure 15 assumes that the network fault has occurred between CoreMA A and CoreMA B in the example model. As a routine procedure, SM sends a HHB message to the first CoreMA on the CoreRing in a certain direction which is the CoreMA A. The CoreMA A propagates the HHB message to CoreMA B. Since, the connection between the CoreMA A and CoreMA B is lost; the CoreMA B will not receive the HHB message. After a certain period of time (expiration of CoreMA B's HHB timer), CoreMA B can detect a network fault and starts a fault recovery procedure. The fault recover procedure involves two steps: (1) reporting the fault to SM, and; (2) announcing the fault to the downward CoreMA. To report a fault to SM, CoreMA B directly sends a pseudo HHB (HHB* in the Figure 15) message to SM. CoreMA B also sends a pseudo HHB message to the downward CoreMA to notify fault. The pseudo HHB is an equivalent to HHB message with an indication in the message specifying that it is a pseudo message.

Note, however, that fault reporting to the downward CoreMA may cause recovery implosion. Assume that there is a problem in one of the links, the CoreMAs right after the failed link can detect failure and subsequently start the fault recovery procedure, simultaneously. This simultaneous fault recovery procedure can cause SM to implode with the number of reports by subsequent CoreMAs in the RMCP-3 session for the same failure. Therefore it is required for the CoreMA to avoid fault recovery implosion by sending a pseudo HHB message to the downward CoreMA as well. Avoiding fault recovery implosion requires each CoreMA to set a different HHB time. Figure 16 shows how each CoreMA sets a different HHB time based on the distance from SM. Smaller HHB time should be set to CoreMA that is closer to the SM.

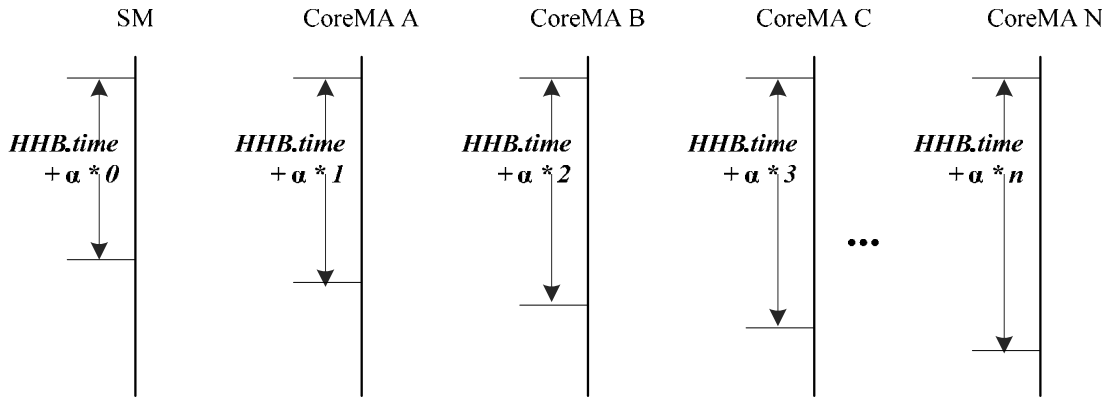

Figure 16 – Varying HHB timeout to avoid an HHB implosion

7.2.3.2 Status monitoring

This capability enables the SM to query the status information of CoreMA. Figure 17 shows how SM query the status of the CoreMA C. SM sends the STREQ message to CoreMA C. The CoreMA C answers the query by sending STANS message within a certain time (i.e., Report time) to report its status.

Since the required status information depends on the RMCP-3 service type, the detailed status information can be extended. The status information includes current available room for additional CMA, bandwidth, and list of EdgeMAs in the EdgeTree.

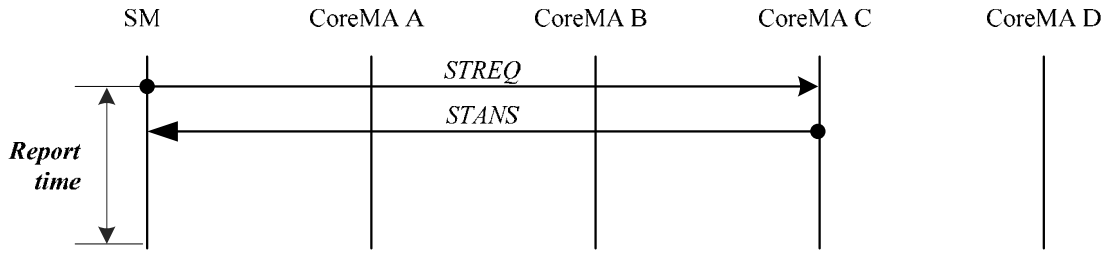

Figure 17 – Status monitoring of CoreMA

STREQ message includes the following information:

– Status information needed by the SM (mandatory).

STANS message includes the following information:

– Status information requested in the STREQ message (mandatory).

7.2.4 Session termination

This function is equivalent to clause 7.1.5. This clause describes the session termination function of the CoreMA. Figure 18 shows the procedure for session termination. SM requests each CoreMA to leave the session by sending a TERMREQ message. Once the CoreMA receives TERMREQ message from SM, it responds to SM by sending a TERMANS and subsequently forwards the TERMREQ message to its EdgeMA in the EdgeTree. The processing of TERMREQ message inside the EdgeTree is described in the clause 7.3.6.

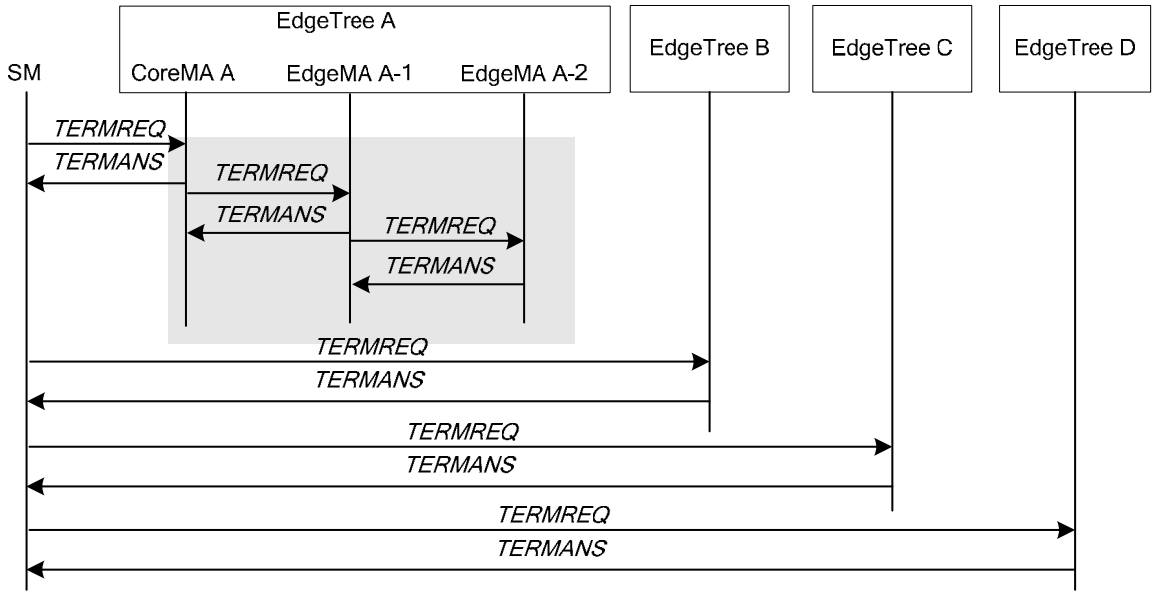

Figure 18 – Session termination

TERMREQ message includes the following information:

– Reason for terminating RMCP-3 session (mandatory).

TERMANS message includes the following information:

– Result of leaving the RMCP-3 session (mandatory).

7.2.5 Fault detection in the core domain

This clause explains the details of the CoreMA in detecting fault in the core domain. The CoreMA can detect fault with the horizontal heartbeat procedure mentioned in clause 7.2.3.1. Upon the reception of pseudo HHB message, the SM will need to make CoreRing recovery by finding more efficient structure for CoreRing. Although the recovery procedure is very important in maintaining the resilience and efficiency of the CoreRing, this specification does not describe the details of the CoreRing recovery procedure, because it is an implementation issue.

7.3 **EdgeMA's operation**

The user get access to IP multicast application through EdgeMA. Unlike CoreMA, EdgeMA can dynamically join or leave a RMCP-3 session anytime, since the EdgeMA is the end-system. The following summarizes the required capabilities of EdgeMA in the RMCP-3 protocol.

- Session subscription;
- Dynamic join and leave of RMCP-3 session;
- Auto-configuration to form EdgeTree;
- Bidirectional data relaying;
- Fault detection and recovery in EdgeTree.

7.3.1 **Subscription**

The EdgeMA must go through the initiation process to subscribe to the RMCP-3 session. EdgeMA subscribes to the RMCP-3 session through SM. Before the EdgeMA initiation process, the EdgeMA needs the RMCP-3 session information. The RMCP-3 session information is announced through various methods, such as web-page, e-mail, etc. The user of the EdgeMA selects the multicast session and starts the EdgeMA initiation process.

The EdgeMA sends a SUBSREQ message to the SM to subscribe to the multicast session as shown in Figure 19. The SUBSREQ message contains information for the SM to decide whether to accept or deny the session subscription request by the EdgeMA. In order to implement a manageable multicast service, the SM needs an authentication method to verify the EdgeMA and authorization method to approve of the EdgeMA's request. The decision rule for authentication and authorization is defined at the initial stage of RMCP-3 services. Therefore, this Recommendation does not specify the precise SM's decision rule in accepting the new EdgeMA's subscription request.

Once the SM decides to accept the EdgeMA's session subscription, bootstrap information is given to the EdgeMA in the SUBSANS message which contains the result of the subscription and the list of CoreMAs participating in the RMCP-3 session. If the SM decides not to accept the EdgeMA's session subscription, then it would also return the SUBSANS message containing the reason for denial.

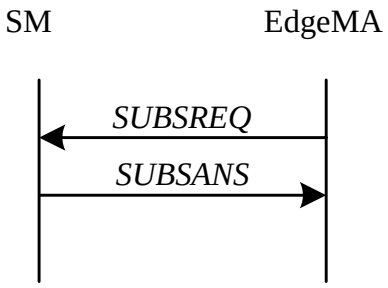

Figure 19 – EdgeMA's subscription

SUBSREQ message issued by EdgeMA to subscribe to the RMCP-3 session includes following information:

- System information of the EdgeMA, e.g., IP address (optional);
- Data profile to be used in the RMCP-3 session (optional);
- Authentication information (optional).

SUBSANS message used to notify the result of the subscription includes the following information:

- Results of subscription (mandatory);
- Bootstrap information including list of CoreMAs (mandatory);
- Data profile to be used in the RMCP-3 session (optional);
- Authentication information (optional).

After a successful subscription, EdgeMA can acquire bootstrap information which consists of a list of CoreMAs in RMCP-3 session. With the bootstrap information, EdgeMA can choose the most appropriate CoreMA to join. Next, the EdgeMA must go through neighbor discovery procedure to find an appropriate PMA in the EdgeTree. Some EdgeMA may be attached to the CoreMA, but most EdgeMAs need to interrelate with its neighboring EdgeMA to configure an optimal EdgeTree. The following clause describes the neighbor discovery procedure in finding the neighboring EdgeMA in the EdgeTree.

7.3.2 **Neighbor discovery**

The neighbor discovery procedure enables the EdgeMA to discover the logical network distances from other neighboring EdgeMAs in RMCP3 session. This capability enables the EdgeMA to measure the logical network distance, since it is impossible to know the exact physical distance of other EdgeMAs.

Once the EdgeMA successfully subscribes to the RMCP-3 session, it needs to acquire contact points of other EdgeMAs. Note, however, that it is not sufficient to determine the exact network distances. The EdgeMA must have a distance measuring capability. The network distance can be measured by various metrics, such as per-hop delay, the number of hop, bandwidth, etc. SM defines adequate metrics which is dependent on the service applications. This Recommendation defines a network distance measurement mechanism by using transmission delay. The EdgeMA include the current time before sending a request message and the responding MA append the current time before sending the reply message. The EdgeMA will get the feel of the distance with the timestamp in the reply message. The time maintained by the systems may vary, thus the SM should have a method for the MAs to synchronize the RMCP-3 service clock.

The EdgeMA can exist in unicast area or in multicast area. The neighbor discovery procedure differs according to the area type. In the multicast area, the only one EdgeMA needs to perform RMCP-3 protocol function and other EdgeMAs do not perform neighbor discovery. The node that performs RMPC-3 protocol function in the multicast area is called HMA. The HMA is the representing node that performs RMCP-3 protocol for all the EdgeMA in the multicast area. The HMA also performs decapsulation to the tunneled RMCP-3 data packet and multicasts the received packet to the multicast domain. The EdgeMAs in the multicast domain can receive the decapsulated multicast packet to be used by the application. Examples of a local multicast area includes the LAN multicast area, which is a small-sized network where the IP multicast is partially deployed. The neighbor discovery procedure enables the EdgeMA to find other EdgeMAs participating in the same RMCP-3 session and to measure the distance.

In the unicast area, the EdgeMA must be able to represent itself in the RMCP-3 protocol. Examples of the such area includes LAN and WAN where IP multicast has yet to be deployed. The neighbor discovery procedure enables the EdgeMA to find other EdgeMAs participating in the same RMCP-3 session and to measure the distance.

The neighbor discovery procedure is used by the EdgeMA after a successful session subscription until it leaves the RMCP-3 session.

7.3.2.1 **Neighbor discovery in the local multicast area**

This capability enables the EdgeMA to find the neighboring EdgeMAs inside local multicast area. It is much efficient to use the multicast capability of the network for the group service, then using the unicast capability. Therefore, the neighboring EdgeMA in the same local multicast area should be designed to be much closer than the neighboring EdgeMA outside the same local multicast area. Thus, finding the neighboring EdgeMA inside same local multicast area should have higher priority than the EdgeMA outside the local multicast area.

a) HMA solicitation and announcement

The EdgeMA in the local multicast area must find the HMA of the multicast area. The HMA solicitation and announcement function enables the EdgeMA to find the HMA in the local multicast area. This capability allows the RMCP-3 protocol to utilize the IP multicast. One EdgeMA in the multicast area must be elected as the HMA taking charge of relaying the multicast data to the local multicast area. The HMA selection criteria are defined by SM at the initial stage of RMCP-3 service. Note, however, that factors such as session subscription time, distance from the CoreMA and precedence of MAID can be used as parameters for choosing HMA. The HMA will receive a control and data packet from its PMA. The PMA is a node that forwards packet to the HMA. The PMA can be either EdgeMA or CoreMA.

Figure 20 is based on the example model. Assume that the EdgeMA B-5 is a new subscriber of the session. Figure 20 shows the procedure in which the EdgeMA B-5 finds other EdgeMAs in local multicast area. The EdgeMA B-5 needs to find HMA in local multicast area. The EdgeMA B-5 multicasts the HSOLICIT message using a specific group address to query the HMA. If the HMA exists, the HMA multicasts the HANNOUNCE message to the local multicast area. The EdgeMA B-5 will receive the HANNOUNCE message and stop neighbor discovery because there is a HMA in the local network. When the HMA receives an HSOLICIT message in local multicast area, HMA will issue HANNOUNCE message to notify its existence as a response.

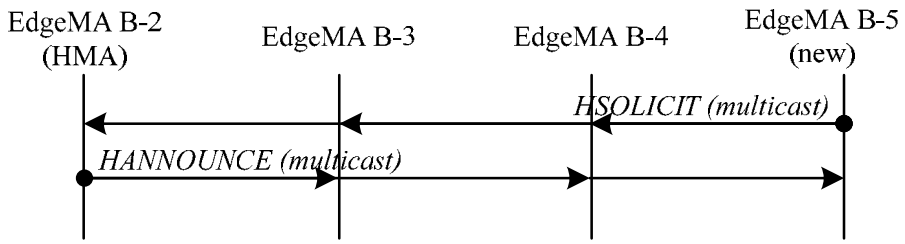

Figure 20 – HMA solicitation and announcement

Since eavesdropping can occur within local network, authentication mechanism is needed. Thus, both HSOLICIT and HANNOUNCE control messages have to include authentication information. However, the method of authentication is not specified in this recommendation. When notifying non-HMAs in the same multicast area of its existence, HMA includes its information, such as Local IP of HMA and HMA lifetime.

HSOLICIT message includes the following information:

– Authentication information (mandatory).

HANNOUNCE message includes the following information:

– Authentication information (mandatory);

– System information such as IP address, service uptime, and EdgeTree connection status with PMA and CMAs (mandatory);

b) New HMA election

The new HMA election function elects HMA in a local multicast area. The new HMA election can occur when there is no HMA in the local network. Initially, the local multicast area will not have any HMA. If new EdgeMA have joined the multicast session, it would have to go through the new HMA election procedure; it multicasts HSOLICIT message to its local network. If there is no HANNOUNCE message for a certain time which is T_HSOLICIT * N_HSOLICIT, the EdgeMA becomes an HMA and multicasts HANNOUNCE message.

If there are other EdgeMAs in the local network, all EdgeMAs should have different HSOLICIT timer value (T_HSOLICIT) to prevent flooding of HSOLICIT message. The EdgeMA with the shortest T_HSOLICIT value has high probability of being an HMA. If the HMA is the only node in the multicast area, then it would not receive HSOLICIT message for HSOLICIT waiting time (W_HSOLICIT). A multicast area with a single HMA may cease to perform the HMA function and does not multicast received session data to the multicast area. In this case, the HMA may act as an EdgeMA in the unicast area.

Another case in which the EdgeMAs must go through new HMA election is when an HMA decides to leave the RMCP-3 session. The rest of the EdgeMA must compete in the new HMA election. Figure 21 shows how the EdgeMA B-2 becomes a new HMA in a local multicast area. As mentioned above, each EdgeMA has its own HSOLICIT timer value (T_HSOLICIT) and the timer value would be common T_HSOLICIT plus a criteria factor. The criteria factor can be derived from various factors such as the distance from CoreMA, IP address, etc.

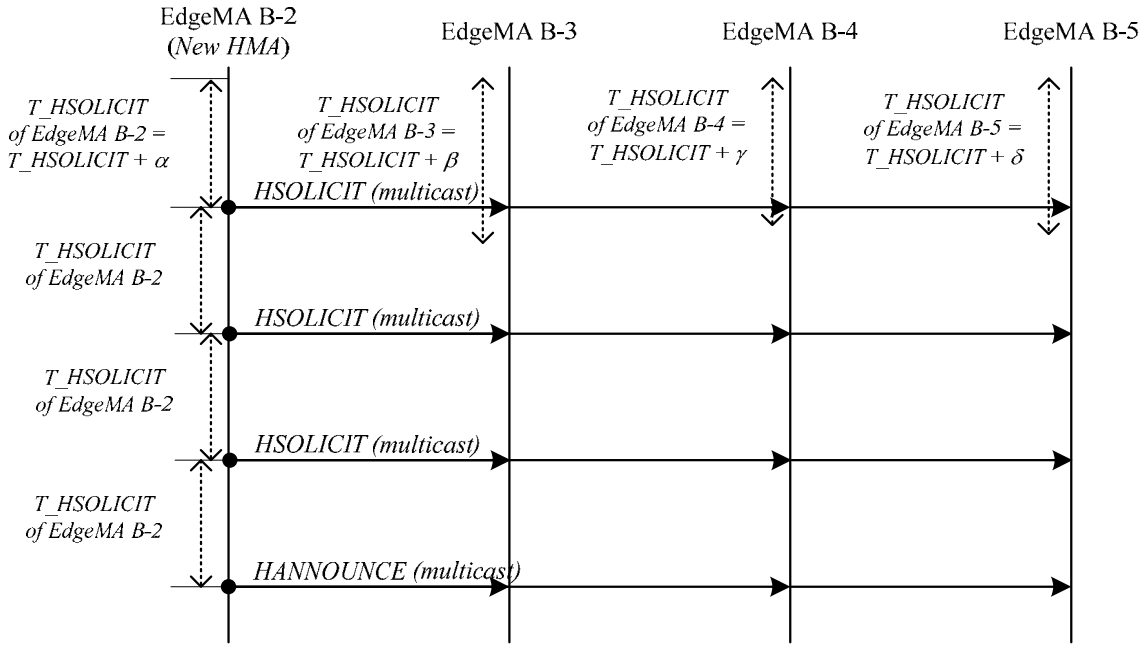

Figure 21 – Procedure of HMA election

As shown in Figure 21, four EdgeMAs are competing to be the HMA in the local multicast area. Four EdgeMAs start its own HSOLICIT timer. The EdgeMA with the shortest HSOLICIT timer will expire, first. As in Figure 21, EdgeMA B-2 has the shortest timer and multicasts the HSOLICIT message to the local multicast area. Other EdgeMAs (i.e., EdgeMA B-3, EdgeMA B-4, and EdgeMA B-5) receive the HSOLICIT message and suppress sending the HSOLICIT message and restart the HSOLICIT timer, again. The EdgeMA B-2 will wait for the HANNOUNCE message for HSOLICIT time. Since there is no answer, EdgeMA B-2 sends the HSOLICIT message for the fixed amount of time. The number of HSOLICIT message sent is defined as a N_HSOLICIT with a default value being 3. The EdgeMA B-2 will know that there is no HMA in the multicast area and decides to be the HMA by multicasting the HANNOUNCE message to the local multicast area.

If the EdgeMA which has already sent an HSOLICIT message receives another HSOLICIT message which is sent from another EdgeMA, the EdgeMA does not suppress nor restart its HSOLICIT timer. This can occur when two or more EdgeMAs multicast HANNOUNCE message to the local network. In such case, one EdgeMA has to be selected as the HMA. Figure 22 shows example of HMA contention. In the example, it is assumed that both EdgeMA B-2 and EdgeMA B-4 have same HSOLICIT timer and same session subscription time.

Since, the EdgeMA B-2 and EdgeMA B-4 has different MAID, this problem can be solved by defining HMA selection rule. In the example, assume that the EdgeMA B-2 has smaller MAID than the EdgeMA B-4's. Since HSOLICIT timer value of EdgeMA B-2 equals to HSOLICIT timer value of EdgeMA B-4, both EdgeMAs issue HANNOUNCE after sending HSOLICIT message N_HSOLICIT times. Upon receiving HANNOUNCE message from other EdgeMA, each EdgeMA decides which EdgeMA should be selected as an HMA according to HMA selection rule. As a result, EdgeMA B-2 is selected as an HMA. Although the rule for selecting HMA can be varied, this recommendation uses following HMA selection rule. If they have different session subscription time, earlier session subscriber will be selected as HMA. Otherwise, the EdgeMA which has lower MAID will be elected as HMA.

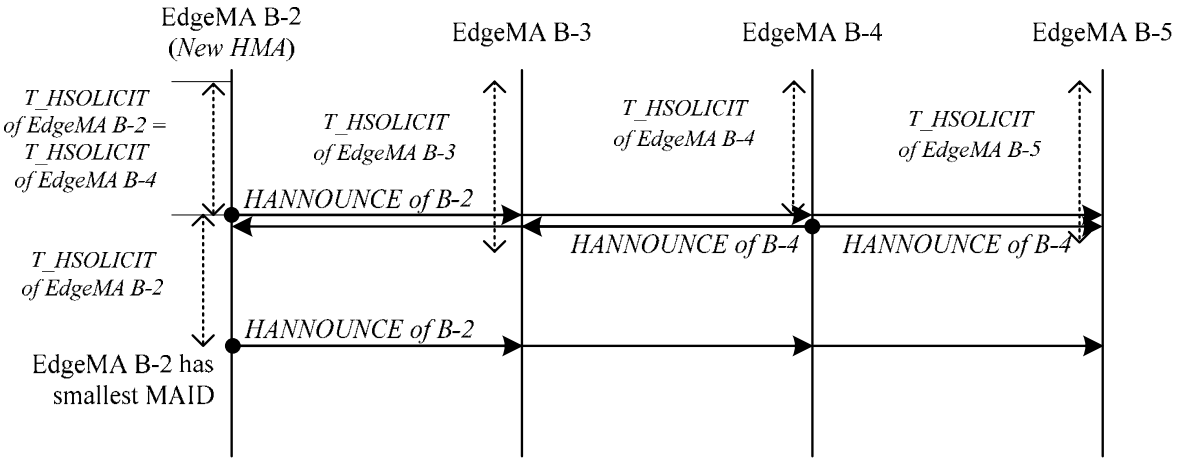

Figure 22 – Example of HMA contention

c) HMA continuity

Once the EdgeMA is elected as an HMA, it must continue its role as an HMA in local multicast area. Figure 23 shows how the HMA can continue its role as an HMA. As mentioned in the new HMA election method, each EdgeMA has its own HSOLICIT timer (T_HSOLICIT). Each EdgeMA starts its HSOLICIT timer and the EdgeMA with the shortest HSOLICIT timer starts sends a HSOLICIT message at the expiration of its timer. Other EdgeMAs with longer HSOLICIT timer will suppress sending HSOLICIT message at the reception of HSOLICIT message from other EdgeMA. The HMA responds by multicasting HANNOUNCE message. This procedure will continue periodically.

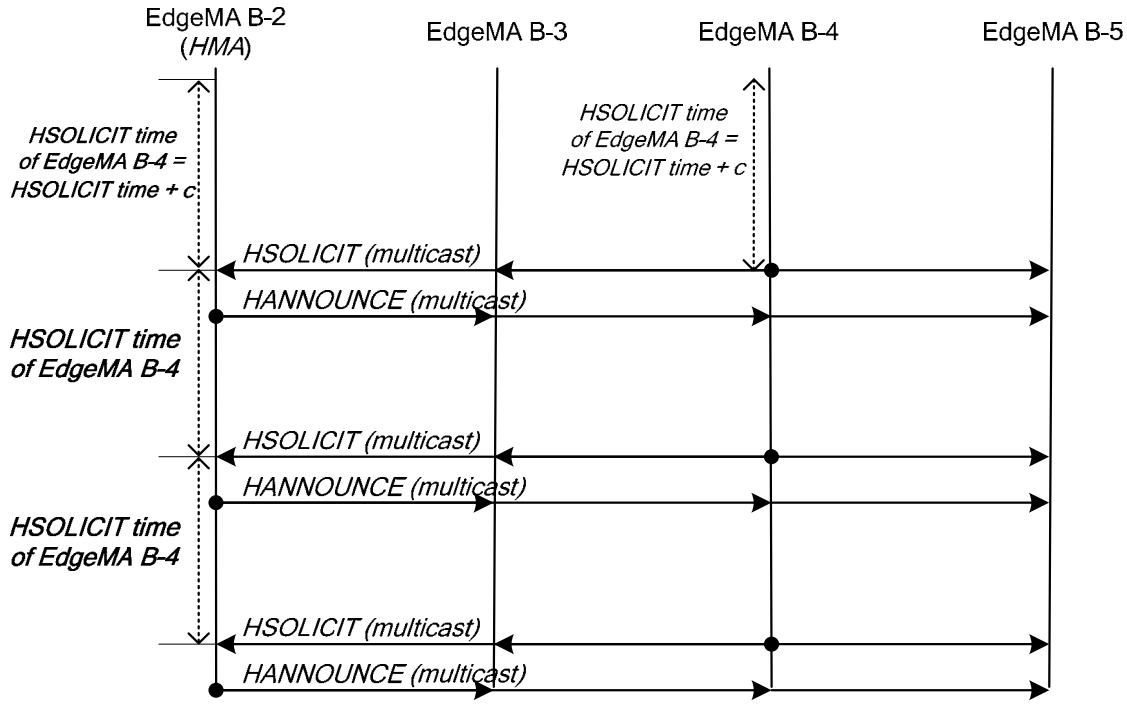

Figure 23 – Periodic HMA announcement caused by HSOLICIT

If HMA does not receive HSOLICIT message within certain period of time (HSOLICIT waiting time), then it will know that there is no other EdgeMA in the local multicast area. The HSOLICIT wait time is defined as W_ HSOLICIT. Recommended HSOLICIT waiting time is T_HSOLICIT * N_SOLICIT. The HMA may discontinue being the HMA and act as a normal EdgeMA in the unicast area.

There are cases when a new EdgeMA may have smaller HSOLICT timer than the current HMA's in the local multicast area. In this case, the role of HMA does not change. The new EdgeMA may become the next HMA if the current HMA

cannot serve session data to the local multicast area. But, the new EdgeMA with the smallest HSOLICIT time will send a HSOLICIT message with the expiration of its timer.

7.3.2.2 Neighbor discovery in the unicast area

Neighbor discovery in the unicast area enables the EdgeMA to find other EdgeMAs in the non-multicast network. Note that the HMA must perform same function to the area as the EdgeMA in the unicast area along with the HMA function for the multicast area.

In RMCP-3, EdgeMA is required to have information about other EdgeMAs participating in the RMCP-3 session, since the EdgeMAs needs to make logical connection with the participating node. This also pertains to the HMA in the multicast area which also needs to have logical unicast connection with the participating node outside of the local multicast area.

The neighbor discovery procedure involves two steps. In the first step, EdgeMA tries to find the nearest CoreMA based on the bootstrap information given by SM. The CoreMA will give the new EdgeMA the list of the EdgeMAs in the EdgeTree that it is managing. With the received information, the new EdgeMA attempts to find the most appropriate EdgeMA to join.

Figure 24 shows the procedure of EdgeMA C-4 obtaining the neighbor list of EdgeMAs. EdgeMA C-4 sends PPROBREQ message to each CoreMAs (which is CoreMA B and CoreMA C) based on the bootstrap information given by SM in the subscription phase. EdgeMA C-4 will get PPROBANS message from each CoreMAs. EdgeMA C-4 will need to make selection based on the PPROBANS messages. As in Figure 24, EdgeMA C-4 decides to select CoreMA C. The PPROBANS message sent by the CoreMAs contains the list of EdgeMAs in its EdgeTree. The EdgeMA C-4 sends PPROBREQ message to each EdgeMAs in the EdgeTree which is managed by selected CoreMA. The PPROBANS message sent by the EdgeMAs contained the needed information for the EdgeMA C-4 to make decision for join procedure.

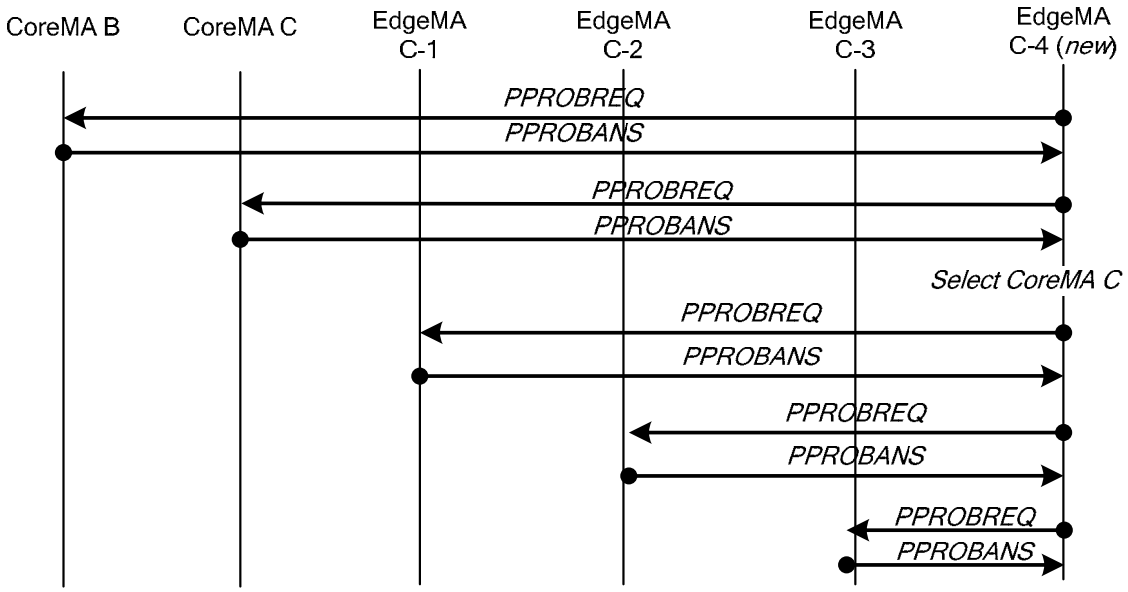

Figure 24 – Sequence of neighbor discovery in the unicast area

PPROBREQ and PPROBANS message may include data profile to check whether probed EdgeMA can support requirements of probing EdgeMA or not. Since EdgeMA may not know all EdgeMAs on the EdgeTree, probed EdgeMA should include its neighbor list in the PPROBANS message. By exchanging neighbor list, EdgeMAs can know all EdgeMAs on the same EdgeTree. To prevent loop, probed EdgeMA has to include rootpath information in the PPROBANS message. Probing EdgeMA should not establish connection with the closest probed EdgeMA if the connection incurs loop on the EdgeTree. Moreover system information should be included within PPROBANS message to prevent performance degradation. By using system information within PPROBANS message, placing low capability node in high position within the tree hierarchy that may cause entire performance degradation can be prevented.

PPROBREQ message includes the following information:

- Timestamp for measuring distance between requesting EdgeMA (mandatory);
- Data profile for data channel negotiation (optional).

PPROBANS message includes the following information:

- Timestamp for measure distance between the requesting EdgeMA (mandatory);
- Neighbor list known by the EdgeMA (mandatory);
- ROOTPATH containing path starting from CoreMA (mandatory);
- System information such as bandwidth, remaining CMA capacity, uptime, and PMA and CMAs connected (mandatory);
- Data profile for data channel negotiation (optional).

7.3.3 **Join**

After the EdgeMA finds the neighboring EdgeMAs, it must attempt to make connection to the EdgeTree to get the RMCP-3 service. The EdgeTree consists of multiple EdgeMAs with a CoreMA as the root node. Figure 25 shows the procedure of the new EdgeMA C-4 making a successful tree join by exchanging RELREQ and RELANS messages. The EdgeMA C-4 sends RELREQ message to the EdgeMAs in the EdgeTree C, which consists of EdgeMA C-1, EdgeMA C-2, and EdgeMA C-3. The EdgeMA C-1 and EdgeMA C-3 sends a RELANS with a refusal. Eventually, the new EdgeMA C-4 selects EdgeMA C-3 which has sent a positive RELANS message.

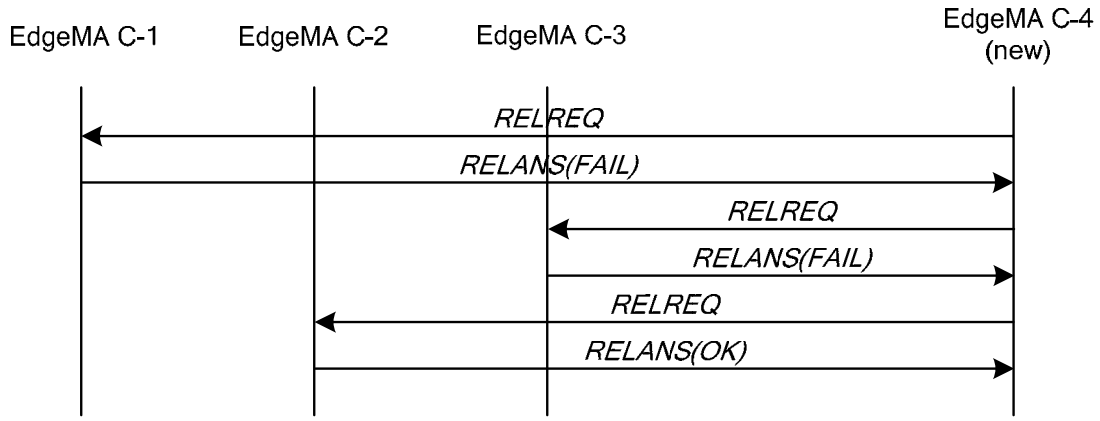

Figure 25 – Sequence of successful tree join

If the CMA wants to re-negotiate data channel with its PMA, it includes new data profile within RELREQ message; success of re-negotiation is not guaranteed. Although data channel is already established, loop can be occurred after establishment of data channel as reconfiguration of EdgeTree. Thus PMA can include its rootpath within RELANS message. Upon the RELANS message, the CMA checks whether loop is occurred. If loop is occurred, the CMA should change its PMA.

RELREQ message includes the following information:

- Neighbor list known by the EdgeMA (mandatory);
- System information such as bandwidth, CMA capacity, and uptime (optional);
- Data profile for data channel negotiation (optional).

RELANS message includes the following information:

- Result for the relaying request (mandatory);
- ROOTPATH containing path starting from CoreMA (optional);
- SYSINFO such as bandwidth, CMA capacity, PMA and CMS connected, and uptime (optional);
- Data profile for data channel negotiation (optional).

7.3.4 **Leave**

Differing from the CoreMA, EdgeMA can leave the RMCP-3 session at anytime based on its own will or administrative purpose. There are three different cases in which the EdgeMA leaves the RMCP-3 session.

- Leaving at own will;
- Leaving with request by SM;
- Leaving with request by PMA.

7.3.4.1 EdgeMA session leave

EdgeMA may leave the RMCP-3 session any time. Before leaving the RMCP-3 session, it must go through some process to reconstruct the EdgeTree. Two types of procedure should be considered in the EdgeMA session leave.

– Non-HMA leave;

– HMA leave.

a) Non-HMA leave

This procedure is for the EdgeMA which is not acting as an HMA to leave the RMCP-3 session. The goal of this procedure is to make sure that the child of the leaving EdgeMA finds an appropriate PMA in order to construct a robust EdgeTree. Before leaving, the EdgeMA informs its CMA that it is leaving from the session. The CMAs will try to find and connect to new PMA by performing the join procedure defined in 7.3.3. After a successful connection with a new PMA is established, CMA notifies the leaving EdgeMA of its reconnection. If the CMA cannot find a new PMA, it must start the RMCP-3 session from the initiation phase defined in **오류! 참조 원본을 찾을 수 없습니다.**. After receiving a successful response from its CMAs, the leaving EdgeMA informs its PMA that it will be leaving.

Figure 26 shows the procedure for session leave of non-HMA. Assume that EdgeMA C-2 of the example model decides to leave the RMCP-3 session. The EdgeMA C-2 has two CMAs; EdgeMA C-3 and EdgeMA C-4. The EdgeMA C-2 informs its CMAs with a LEAVREQ message. The EdgeMA C-3 tries to connect to EdgeMA C-1 and EdgeMA C-4 to CoreMA C. This procedure omits the PMA finding procedure for EdgeMA C-3 and EdgeMA C-4 which is equivalent to EdgeMA join procedure in 7.3.3. The two CMAs may need to start from the initiation phase, if it cannot find adequate PMA. After a successful join, CMAs reply with successful LEAVANS message to EdgeMA C-2. The EdgeMA C-2 send a LEAVREQ message to its PMA (i.e., EdgeMA C-1) indicating that it is ready to leave the RMCP-3 session. The EdgeMA C-1 responds with LEAVEANS message.

After sending in LEAVREQ message, the leaving EdgeMA should wait for LEAVANS message until LEAVE timer is expired. LEAVE timer is a timer maintained by the leaving EdgeMA in waiting for LEAVANS message from its CMAs. This timer used to prevent leave of EdgeMA before its CMA finds a new PMA. Considering that finding a new PMA does not take long time, the leave timer should be set to appropriate time value and it is defined as a T_LEAVE. When the leave timer expires, the leaving EdgeMA should perform the remaining procedure even if LEAVANS message does not arrive from all of its CMAs. In that case, the leaving EdgeMA considers that the CMA which does not send answer leaves abnormally; it does not care for the CMA.

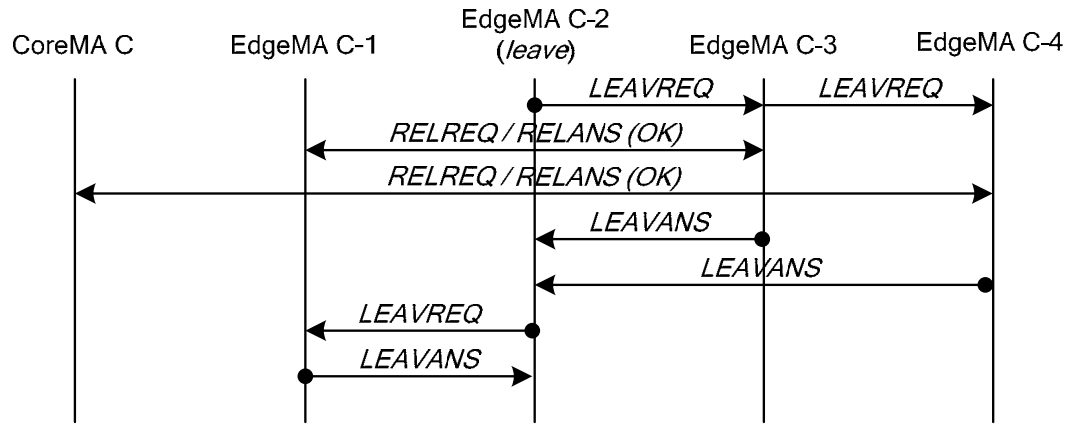

Figure 26 – Non-HMA leave procedure

The EdgeTree C will change its structure as shown in Figure 27 with EdgeMA C-2 leaving the RMCP-3 session.

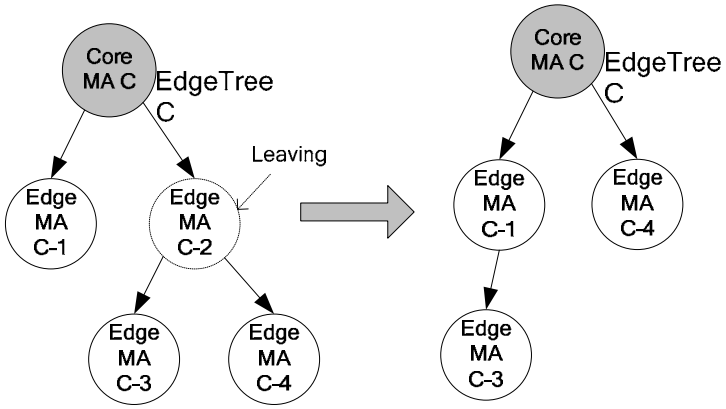

Figure 27 – EdgeTree transition after EdgeMA C-2 leave

The LEAVREQ and LEAVANS messages used in the non-HMA leave are as follows.

LEAVREQ message includes the following information:

– Reason for leaving (mandatory).

LEAVANS message includes the following information:

– Result of leaving (mandatory).

b) HMA leave

If the HMA in the local multicast network leaves the RMCP-3 session, new HMA needs to be elected. Basically, the procedure of HMA leave is same as non-HMA leave procedure, except that the HMA can inform other EdgeMAs in the multicast domain to start HMA election process.

Figure 28 shows the HMA leave procedure. As in the example model, EdgeMA B-1 is the PMA of the HMA. The EdgeMA B-2 is the HMA that tends to leave the RMCP-3 session. The EdgeMA B-2 multicasts the HLEAVE message to the local multicast area notifying EdgeMAs to select new HMA. EdgeMA B-2 notifies its PMA of RMCP-3 session leave with LEAVREQ message. If the EdgeMA B-2 has CMAs, which is not in the multicast area, it must notify them of its session leave by sending LEAVREQ.

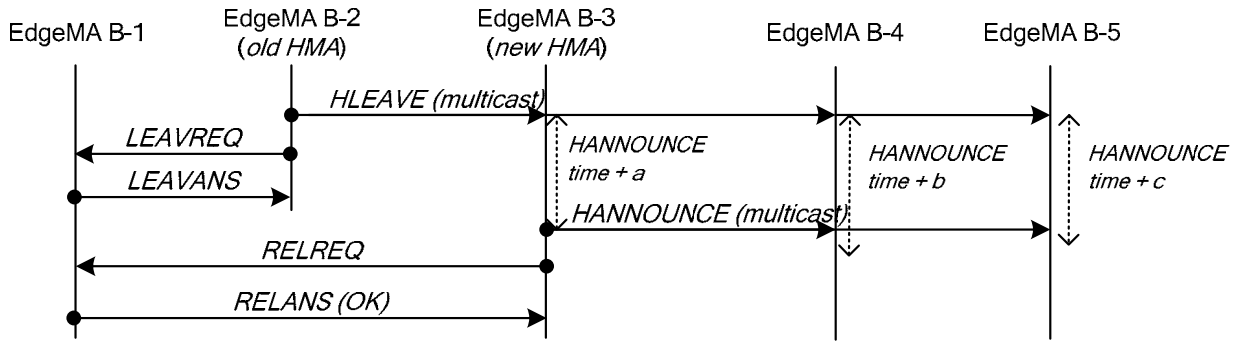

Figure 28 – HMA leave procedure

The remaining EdgeMAs, which are EdgeMA B-3, EdgeMA B-4, and EdgeMA B-5, must participate in the new HMA election. This HMA election procedure is slightly different from the HMA election procedure in the (b) of clause 7.3.2.1. The reason is that if the old HMA can inform the remaining EdgeMAs, then the local multicast area can quickly recover from the lost of old HMA. The remaining EdgeMAs will start a HANNOUNCE timer. The HANNOUNCE timer is defined as a T_HANNOUNCE. The HANNOUNCE time must be a small value for quick HMA election. Each EdgeMA adds a small number to the HANNOUNCE time to prevent HANNOUNCE message flooding. Any EdgeMA receiving the HANNOUNCE message needs to suppress sending HANNOUNCE message upon receiving such message from other EdgeMA. The EdgeMA which succeed in sending the HANNOUNCE message becomes the HMA of the local multicast area. The structure of the EdgeTree B will change as shown in Figure 29.

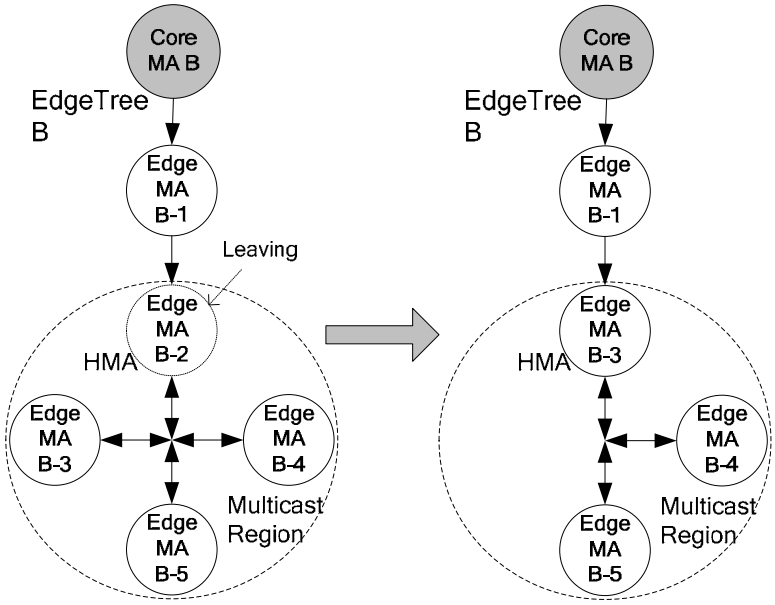

Figure 29 – EdgeTree transition after EdgeMA B-2 leave

As mentioned above, if two or more EdgeMAs simultaneously send the HANNOUNCE message, the EdgeMA must be able to select an HMA with an equivalent selection rule. In such case, one of them has to be selected as the HMA. Although the rule for selecting HMA can vary, this recommendation uses following HMA selection rule. If they have different session subscription time, earlier session subscriber will be selected as HMA. Otherwise, the EdgeMA which has lower MAID will be elected as HMA.

The HMA newly elected should attach to EdgeTree for service continuation. For fast attachment, HLEAVE message includes the rootpath and neighbor list of leaving HMA. Using such information, new HMA can establish connection with PMA of leaving HMA and also know EdgeMAs which exist outside the local network.

HLEAVE message includes the following information:

- Neighbor list known by the EdgeMA (mandatory);
- ROOTPATH containing path starting from CoreMA (mandatory);
- Authentication information (mandatory);
- Reason of leaving (mandatory).

7.3.4.2 EdgeMA expulsion

The EdgeMA expulsion function enables both PMA and SM to expel certain EdgeMA. In RMCP-3, EdgeMA can be expelled by SM or by the PMA with the following reasons.

- SM expelling specific EdgeMA for administrative reason;
- PMA expelling CMAs due to lack of system resources;
- PMA expelling CMAs to eliminate loop in the EdgeTree.

a) Expulsion by SM

Figure 30 shows the procedure of EdgeMA expulsion procedure by the SM. The SM can expel certain EdgeMA for causing trouble to the RMCP-3 service or for unauthorized access of RMCP-3 service. The SM sends a LEAVREQ message to the pertaining EdgeMA with appropriate reason code; SM_KICKOUT. The EdgeMA must notify its PMA and CMA of its leave and leave the RMCP-3 session. The CMA of leaving EdgeMA should find a new PMA.

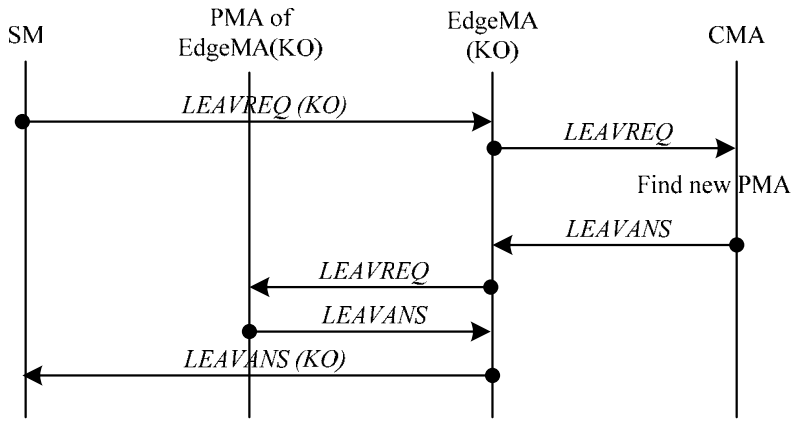

Figure 30 – EdgeMA expelled by SM

b) Expulsion by PMA

During the RMCP-3 service, the PMA may suffer from system resource shortage. The PMA may need to expel some CMAs to relieve from the system resource shortage problem. Figure 31 shows the procedure of CMA demission. The old PMA ask the EdgeMA to leave with LEAVREQ message. The expelled EdgeMA does not need to leave the RMCP-3 session, but it needs to find a new PMA. If the expelled EdgeMA finds a new PMA, it replies to the old PMA with LEAVANS message.

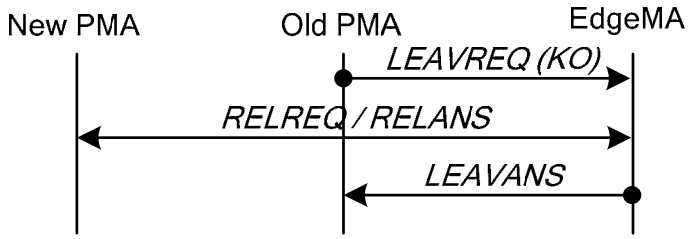

Figure 31 – EdgeMA expelled by PMA

This procedure can be used to recover from loop as described in clause 7.3.7.1.

7.3.5 EdgeTree reconstruction

The RMCP-3 Hybrid Tree changes with join/leave by various EdgeMAs. The Hybrid Tree may be reconstructed to provide more efficient structure which leads to more efficient protocol and data processing. The EdgeMA continually exchange PPROBREQ/PPROBANS message with the neighboring MAs to find more efficient PMA. The mechanism used in EdgeTree reconstruction is the same as neighbor discovery procedure described in 7.3.2. The EdgeMA has sequentially send PPROBREQ message to its neighbor and the neighbor would reply with PPROBANS message containing QoS-related parameter for the EdgeMA to analyze in reconstructing the EdgeTree. If the EdgeMA finds a better PMA, then it can switch to new PMA through join mechanism described in 7.3.3.

The EdgeTree send the PPROBREQ message every PPROBE time. The PPROBE time is defined as T_PPROBE with a default value of 45 seconds. A random number time should be added to the PPROBE time in order to prevent PPROBREQ implosion which could occur if large number of EdgeMA simultaneously tries to change their PMA altogether.

The EdgeTree reconstruction mechanism is an optional function and may be used for application with large participants. Since, the RMCP-3 Hybrid Tree has a distributed structure; small change in the EdgeTree would not affect the protocol or data performance. The changes should be made if there is a large difference in performance. Maintenance

RMCP-3 must maintain a robust EdgeTree. This clause defines functions in the RMCP-3 for maintaining robust EdgeTree.

7.3.5.1 EdgeTree maintenance

EdgeTree maintenance enables the EdgeMAs to maintain the interconnection in the EdgeTree. CMA periodically exchanges RELREQ and RELANS message with its PMA. The CMA has a RELREQ timer; it is defined as a

T_RELREQ. If the timer expires, CMA sends RELREQ message to its PMA. The PMA answers the RELANS message with a positive response as shown in Figure 32.

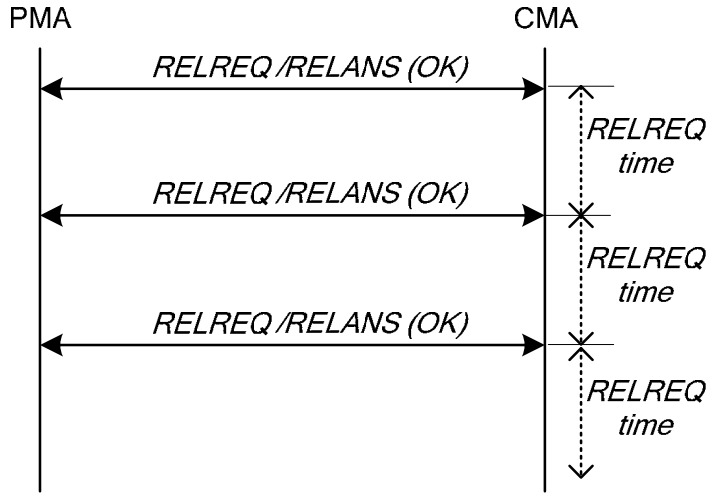

Figure 32 – EdgeTree maintenance procedure

If the PMA cannot relay session data to the CMA, it can respond with RELANS including a reason of rejection. The CMA would then find a new PMA by performing join procedure. If the PMA suddenly breakdown before going through a normal EdgeMA session leave procedure, its CMA can acknowledge such incident if it does not receive a RELANS message from its PMA. The CMA sends RELREQ message for N_RELREQ times before determining the fault of its PMA. In this case, the CMA needs to find new PMA through join procedure.

7.3.5.2 Vertical heartbeat

Vertical heartbeat (VHB) is used to detect fault in the EdgeTree. This capability is tightly coupled with the HHB mechanism used in CoreRing. Figure 33 shows the two-level heartbeat scheme of RMCP-3. The SM periodically generates HHB message with the expiration of HB timer. The HB timer is defined as T_HB which is a timer for issuing heartbeat message. If the CoreMA receives a HHB message, it issues a VHB message for its EdgeTree. The CoreMA appends its MAID to the HHB message and forwards it to the next CoreMA. When EdgeMA receives a VHB, it appends its MAID to the VHB message and forwards it to its CMAs. This procedure continues until the VHB reaches the leaf EdgeMA. As VHB message propagates along the EdgeTree, each EdgeMA can make its own rootpath by using hop information within the VHB message.

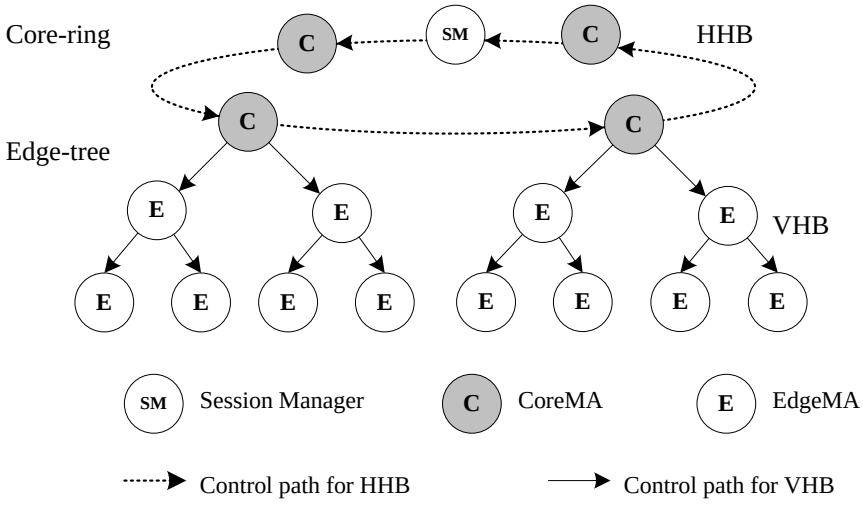

Figure 33 – Two-level heartbeat

Figure 34 shows how VHB message is propagated in the EdgeTree.

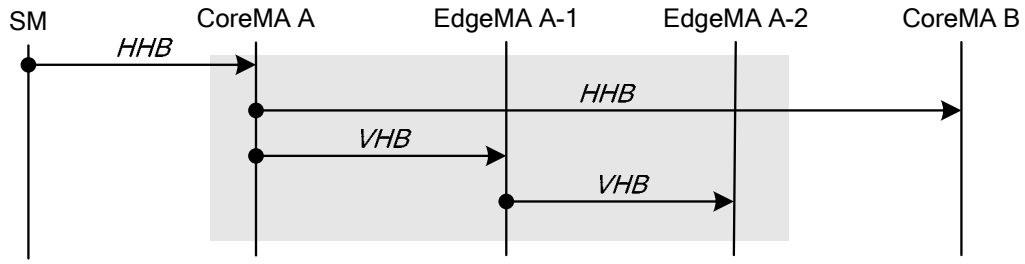

Figure 34 – Successful VHB propagation sequence

The normal heartbeat sequence is described in this clause. The abnormal heartbeat sequence is considered in the clause 7.3.7,

VHB message is equivalent to HHB message which includes the following information:

- ROOTPATH which keep track of MAs starting from the CoreMA (mandatory);
- Authentication information (optional).

7.3.5.3 Monitoring

RMCP-3 provides the mechanism to query the status of the session members. Figure 35 shows the procedure of SM querying the status of EdgeMA A-2. SM requests the EdgeMA A-2 to report its status by sending a STREQ message containing information request on the pertaining node. EdgeMA A-2 responds with a STANS message status of the information requested by the SM. The EdgeMA A-2 must respond the message within the report time.

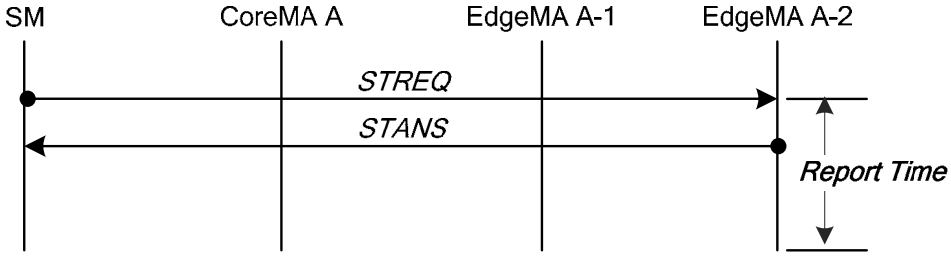

Figure 35 – Tree Monitoring – Status Report

STREQ message includes the following information:

- Requesting information (mandatory).

STANS message includes the following information:

- Requested information (mandatory).

7.3.6 Service termination

The RMCP-3 session can be terminated. The session termination is initiated by the SM. Since, the RMCP-3 session is a many-to-many session, the SM, which is in charge of managing the RMCP-3 session, needs to explicitly terminate the RMCP-3 session.

Figure 36 shows the procedure of session termination. The SM sends TERMREQ message to each CoreMAs. If the CoreMA receives the TERMREQ message, it sends TERMANS message back to SM and then forwards the TERMREQ message to its EdgeMAs. All MAs including CoreMA and EdgeMA must leave the session upon the reception of the TERMREQ message.

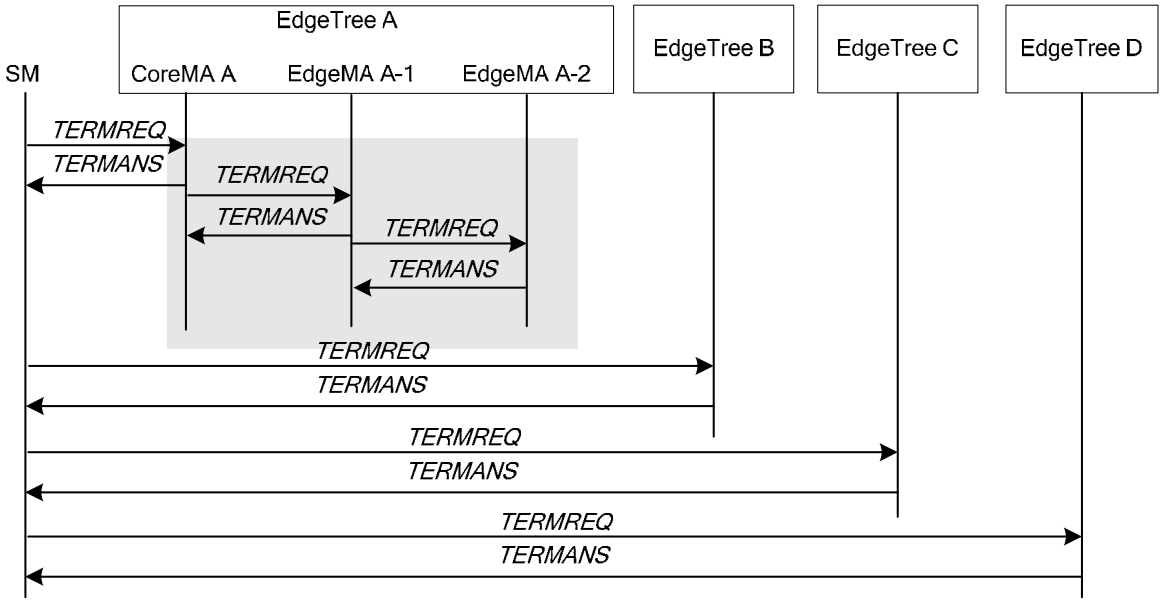

Figure 36 – Session termination

TERMREQ message includes the following information:

– Reason for termination (mandatory).

TERMANS message includes the following information:

– Result of termination (mandatory).

7.3.7 Fault detection and recovery in the edge domain

Two types of fault can occur in the EdgeTree.

– Network partitioning;

– Loop.

This clause describes the fault detection and recovery procedures of the edge domain.

7.3.7.1 Loop detection

The loop detection function enables the RMCP-3 to detect network loop in the EdgeTree. The loop can be detected through periodic heartbeat procedure. Each EdgeMA appends its MAID into the VHB message before forwarding to its CMA. This would leave a trace and topology of MAIDs in the EdgeTree. The EdgeMA check the VHB message for the MAID of its CMA before appending its MAID into the VHB message. If the MAID of its CMA appears in the VHB message, then the EdgeMA would know that a loop been formed in the EdgeTree. The EdgeMA must disconnect the loop with a LEAVREQ/LEAVANS message to the CMA in the VHB message.

Figure 37 shows the procedure of EdgeMA C detecting loop in the EdgeTree and resolving the problem of loop. Assume that loop has been formed in the EdgeTree of EdgeMA A, EdgeMA B, EdgeMA C. VHB message will be delivered from the CoreMA to the EdgeTree with the sequence of EdgeMA A, EdgeMA B, and EdgeMA C. EdgeMA C is required to forward VHB to its CMA, i.e. EdgeMA A. But, the EdgeMA C checks the list in the VHB message and sees that its CMA which is the EdgeMA A is already in the list of VHB message. Thus, EdgeMA C will know that a loop has occurred in the EdgeTree. To resolve the loop problem, EdgeMA C make disconnection with the EdgeMA A by exchanging LEAVREQ and LEAVANS messages and does not send the VHB message to EdgeMA A.

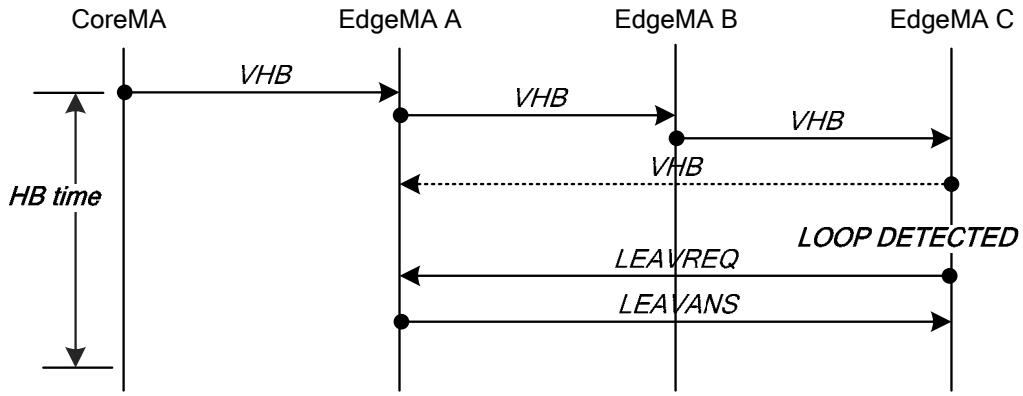

Figure 37 – Loop detection and recovery in edge domain

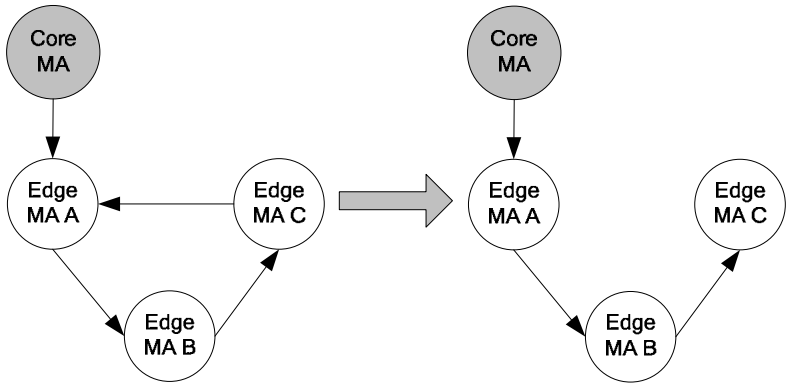

Figure 38 – EdgeTree structure after loop recovery

Figure 38 illustrates the EdgeTree structure after the loop recovery process. The connection from EdgeMA C to EdgeMA A has been disconnected by the EdgeMA C.

7.3.7.2 Network partition detection

The network partition detection enables the EdgeMA to detect network partitioning with the use of periodic VHB message. RMCP-3 service has a heartbeat function to ensure service continuity. The EdgeMA can detect network partitioning if it does not receive the periodic VHB message within the expected heartbeat time. The expected heartbeat time is a maximum waiting time by the EdgeMA for VHB message with the value of HB time * N_HB. Both HB time and N_HB is defined as T_HB and N_HB with a default value of 15 second and 3, respectively.

Figure 39 shows the procedure for network partition detection and resolution. Assume that link between EdgeMA D-3 and EdgeMA D-4 has been disconnected. The VHB message sent by the EdgeMA D-3 will be lost due to the link error. The EdgeMA D-4 will be expecting a VHB message for the expected heartbeat time. The EdgeMA D-4 will not receive the VHB message, and detects network partition. EdgeMA D-4 sends a pseudo VHB message to its CMAs and selects other PMA by neighbor discovery and join procedure.

However, in case of network partition, all the lower EdgeMAs (i.e., EdgeMA D-5) also detects network partitioning along with the EdgeMA D-4. If both EdgeMAs tries to process network partitioning, partitioning recovery flooding may arise at the EdgeTree. To avoid partitioning recovery flooding, the EdgeMA issues a pseudo VHB to suppress the simultaneous recovery flooding. If the EdgeMA receives a pseudo VHB message, it suppresses the partitioning recovery procedure. The pseudo VHB message is an equivalent to VHB message with an indication in the message specifying that it is a pseudo message.

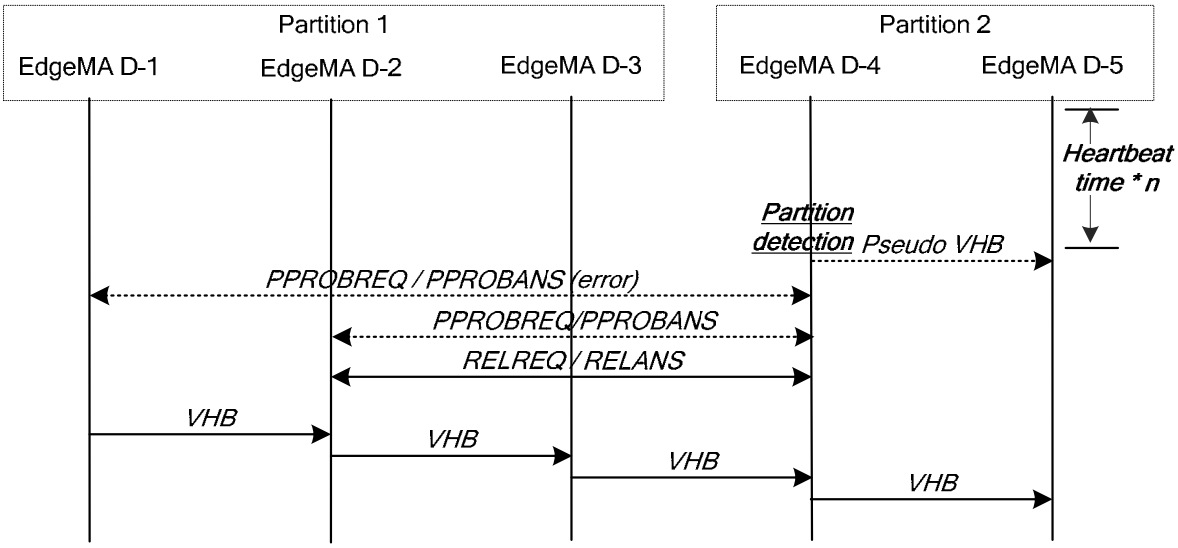

Figure 39 – Partitioning detection and recovery in edge domain

Figure 40 illustrates the EdgeTree structure after partition detection. The EdgeMA D-4 finds new PMA which is EdgeMA D-2 after partition recovery. The EdgeMA D-5 does not change its PMA, since it has received a VHB message from PMA.

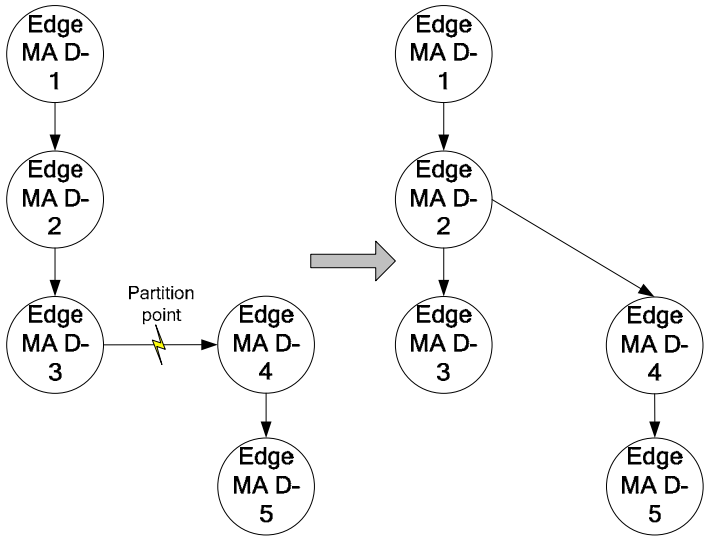

Figure 40 – EdgeTree structure after partition detection

8 RMCP-3 message

8.1 Common RMCP-3 message format

Figure 41 shows the common format for all RMCP-3 messages. The value in the parenthesis represents the length of each field in bits. Each field has the following meaning and value:

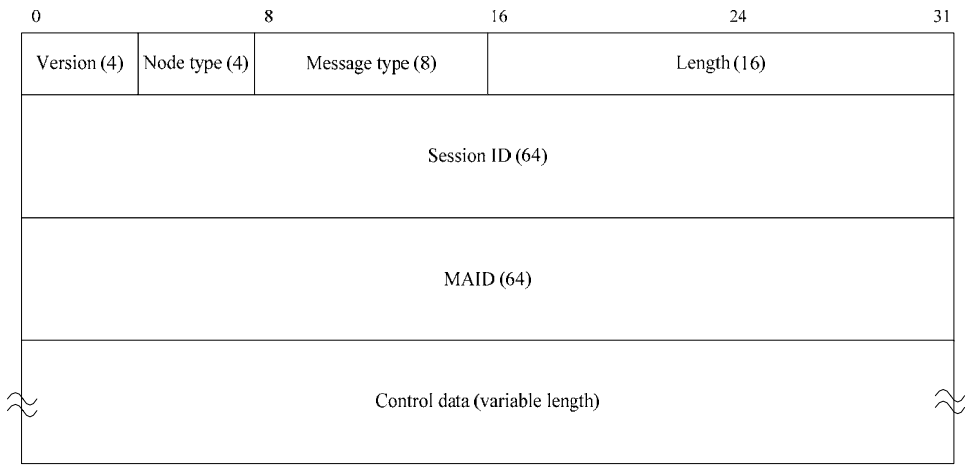

Figure 41 – Common RMCP-3 message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message (see Table 3);

c) *Message type* – denotes the type of the message (see Table 2);

d) *Length* – denotes the total length of the message including control data (in bytes);

e) *Session ID* – denotes a 64-bit integer value that identifies a session;

f) *MAID* – denotes the MAID of the originator or sender of the message. Its value shall contain the local IP address and port number as defined in clause 9.1.2;

g) *Control data* – denotes the control data used by each type of message as necessary.

Session ID and MAID must be a unique value to identify the session and MA, respectively. RMCP-3 provides rule for generating the ID value used for session ID and MAID. The rule is described in clause 9.1.

8.2 Control data format

Figure 42 shows the RMCP-3 control data format. The Control type field describes the type of control data used, and the Length field is the total size of the control data excepting the size of sub-control data. Since the Control type field is 1-byte long, the maximum number of unique control types is limited to 256 cases.

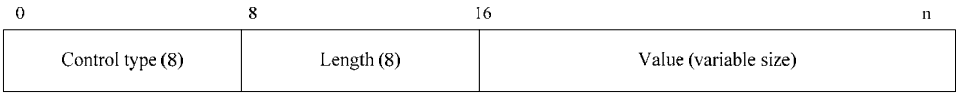

Figure 42 – RMCP-3 control data format

a) *Control type* – denotes the type of control data. Its value shall be set to one of coded value in Table 4);

b) *Length* – denotes the total length of the control data (in bytes, except the length of sub-control data field);

c) *Value* – denotes the value for each control data.

Whenever the message needs to specify detailed control information, RMCP-3 sub-control data is used. The format of the sub-control data is shown in Figure 43.

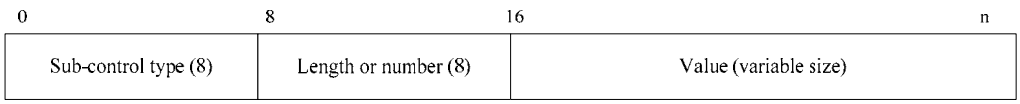

Figure 43 – RMCP-3 sub-control data format

a) *Sub-control type* – denotes the type of sub-control data. Its value shall be set to one of coded value in Table 5 through Table 7;

b) *Length or number* – denotes the length in byte or the number of sub-control data values (depending on the sub-control type);

c) *Value* – denotes the value for each sub-control data.

Control data can be used with only the control type as shown in Figure 44.

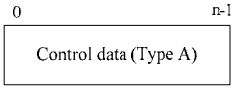

Figure 44 – Use of control type only

Whenever sub-control data is used, an appropriate control data must precede. Figure 45 shows that an example of control data with a sub-control data.

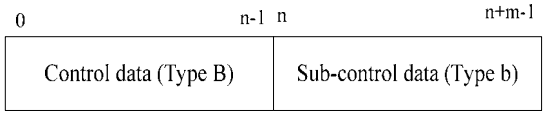

Figure 45 – Use of control data with sub-control data

One or more control data can be used in the RMCP-3 Control data field. An RMCP-3 message which needs to include multiple control data should align multiple control data as shown in Figure 46.

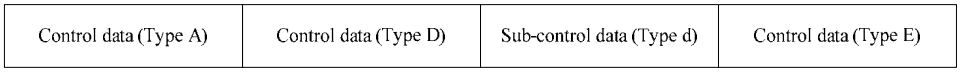

Figure 46 – Use of multiple control data

8.3 RMCP-3 control messages

This clause defines control messages used in RMCP-3. RMCP-3 defines seven pairs in *request and answer* manner (sometimes in *request and confirm* manner) messages and two heartbeat messages.

8.3.1 SUBSREQ

8.3.1.1 The SUBSREQ control message is used to subscribe to an RMCP-3 session. By issuing the SUBSREQ control message, each MA can obtain bootstrap information from SM when accepted. The message format is shown in Figure 47.

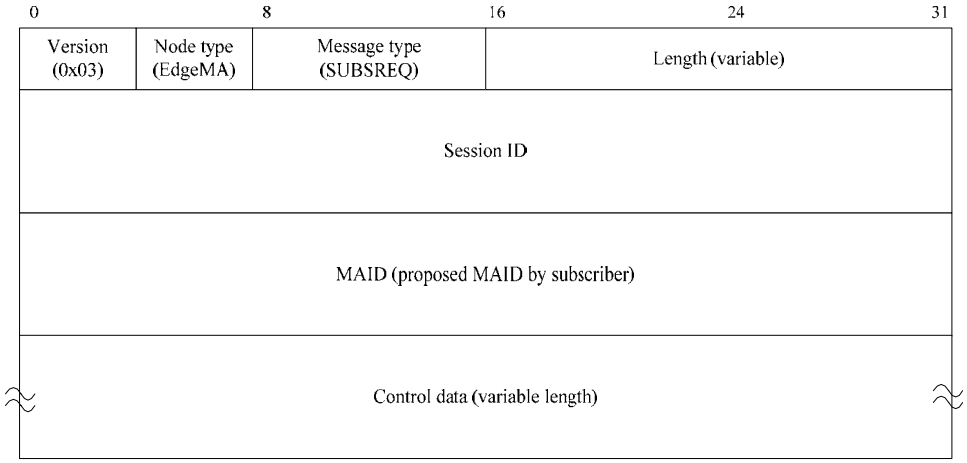

Figure 47 – SUBSREQ control message format

Each field has the following meaning and value:

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x01 (see Table 2);

d) *Length* – denotes the size of the SUBSREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the proposed MAID of the subscriber (proposed by subscriber). Its value shall contain the local IP address and port number as defined in 9.1.2;

g) *Control data* – may include the following information:

8.3.1.2 Following table shows the control data types which can be used within the SUBSREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
SYSINFO	A description of the system information of MA.	O
DATAPROFILE	A description of the requirements for forwarding data.	O
AUTH	Authentication information for verifying the sender. To support several types of authentication mechanism, extensive AUTH sub-control format is defined followed by 2 bytes length AUTH control.	O

8.3.2 **SUBSANS**

8.3.2.1 The SUBSANS control message is used by SM to give the results of the session subscription request and bootstrap information for the session. The message format is shown in Figure 48.

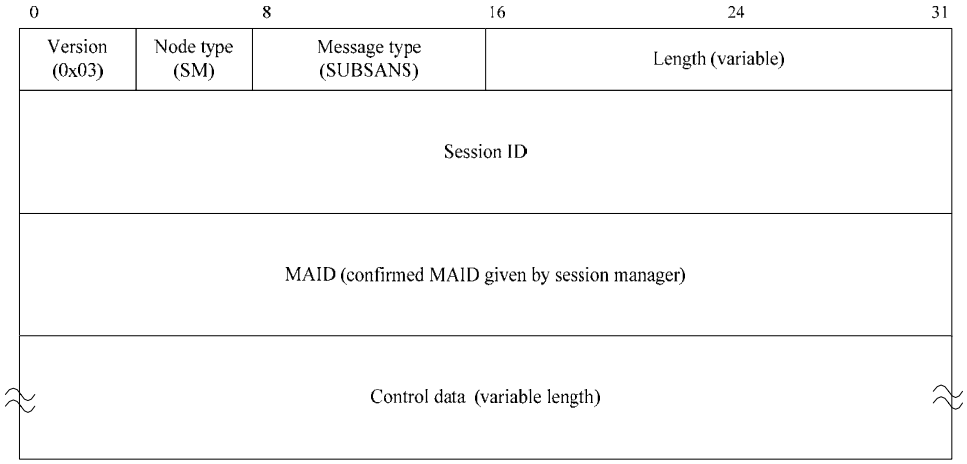

Figure 48 – SUBSANS control message format

Each field has the following meaning and value:

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for SM in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x02 (see Table 2);

d) *Length* – denotes the total length of the SUBSANS control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the confirmed MAID of the subscriber. (Confirmed by SM);

g) *Control data* – may include the following information:

8.3.2.2 Following table shows the control data types which can be used within the SUBSANS message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
RESULT	The result of the subscription request.	M
NEIGHBORLIST	A list of MAIDs for performing the neighbor discovery	M
DATAPROFILE	A description of the requirements for forwarding data.	O
AUTH	Authentication information for verifying the sender.	O

8.3.3 **PPROBREQ**

8.3.3.1 This is used to perform the Neighbor discovery procedure for determining the actual network condition and for exploring neighbors as well. PPROBREQ control message is also used to check whether its counterpart is still alive. Figure 49 illustrates the format of the PPROBREQ control message.

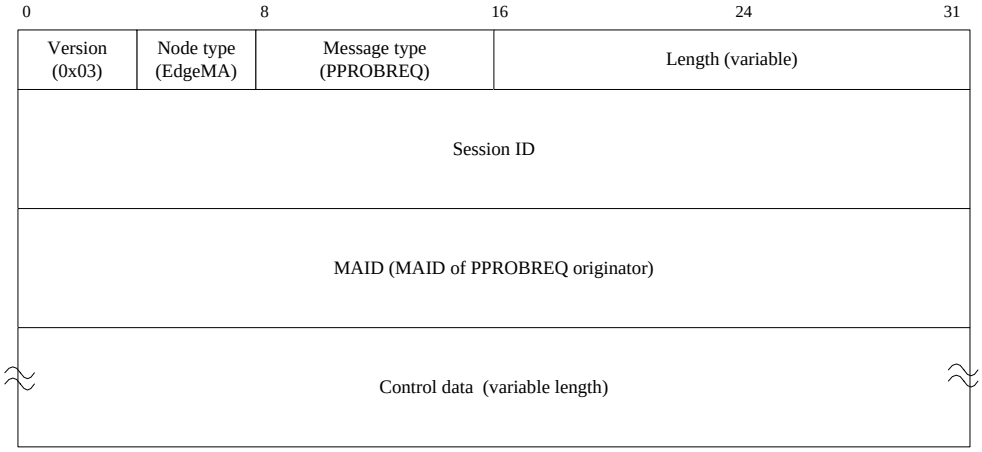

Figure 49 – PPROBREQ control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x03 (see Table 2);

d) *Length* – denotes the total length of the PPROBREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the PPROBREQ control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.3.2 Following table shows the control data types which can be used within the PPROBREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
TIMESTAMP	To measure of distance between sending and receiving MAs.	M
DATAPROFILE	A description of the requirements for forwarding data.	O

8.3.4 **PPROBANS**

8.3.4.1 PPROBANS is a response message to the PPROBREQ control message used in Neighbor discovery procedure to confirm available MA in the network. PPROBANS control message may contain the actual network condition value and a series of its neighbor information. Figure 50 illustrates the format of the PPROBANS control message.

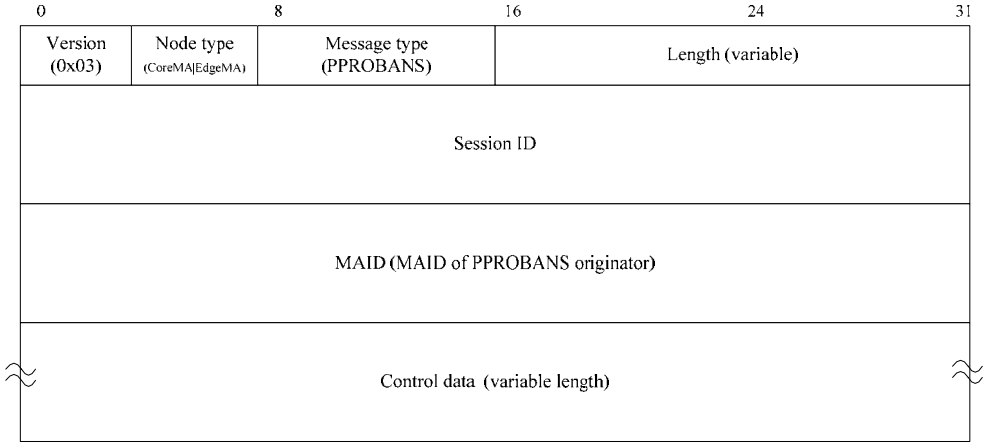

Figure 50 – PPROBANS control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
   - the coded value for CoreMA in Table 3;
   - the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x04 (see Table 2);

d) *Length* – denotes the total length of PPROBANS control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the PPROBANS control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.4.2 Following table shows the control data types which can be used within the PPROBANS message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
TIMESTAMP	To measure of distance between sending and receiving MAs.	M
NEIGHBORLIST	A list of MAs for performing the neighbor discovery.	M
ROOTPATH	A description of the path from CoreMA.	M
SYSINFO	A description of the system information of MA.	M
DATAPROFILE	A description of the requirements for forwarding data.	O

8.3.5 **HSOLICIT**

8.3.5.1 HSOLICIT control message is used to process self-organization in a local network. The purpose of this message is to find the HMA in the local multicast network. Figure 51 illustrates the format of HSOLICIT control message.

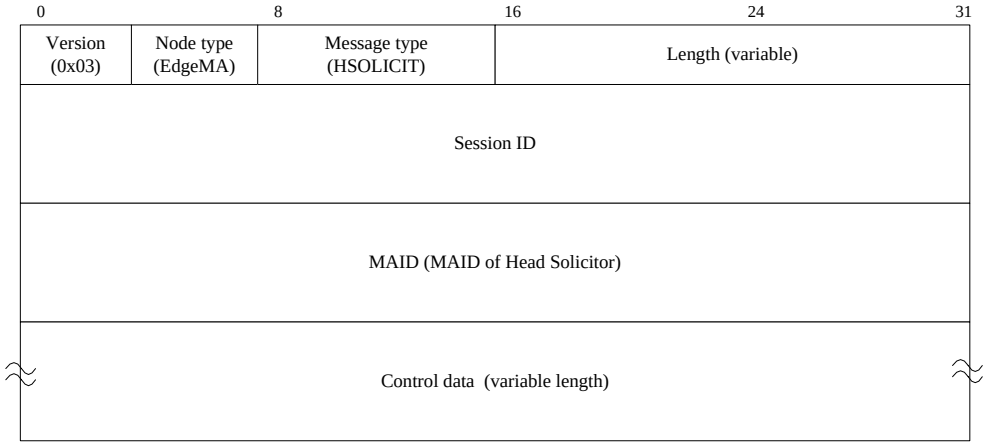

Figure 51 – HSOLICIT control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x05 (see Table 2);

d) *Length* – denotes the total length of HSOLICIT control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the HSOLICIT control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.5.2 Following table shows the control data types which can be used within the HSOLICIT message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
AUTH	Authentication information for verifying the sender.	M

8.3.6 **HANNOUNCE**

8.3.6.1 In response to HSOLICIT control message, HANNOUNCE control message is used to announce HMA's existence in the local multicast network. Figure 52 shows the format of this message.

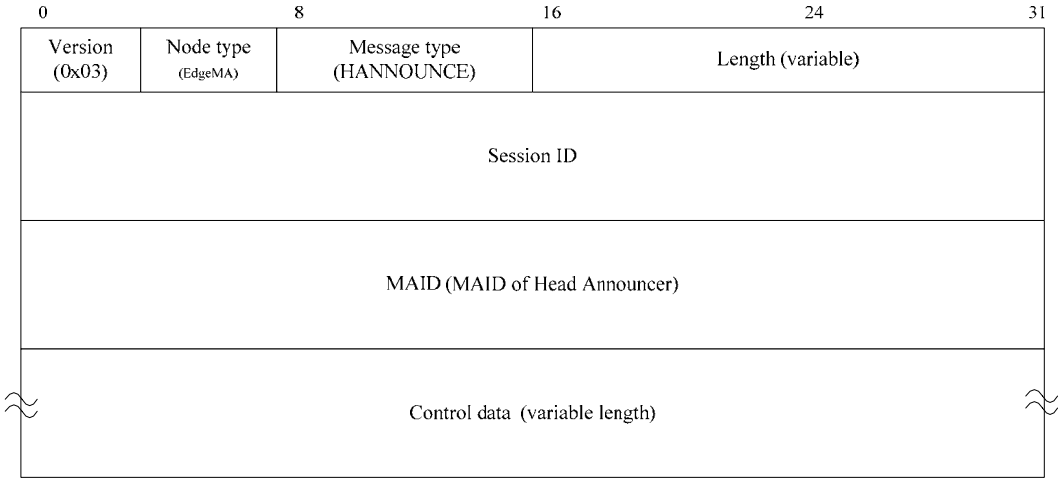

Figure 52 – HANNOUNCE control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message Type* – denotes the type of the message. The value shall be set to 0x06 (see Table 2);

d) *Length* – denotes the total length of the HANNOUNCE control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the HANNOUNCE control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.6.2 Following table shows the control data types which can be used within the HANNOUNCE message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
AUTH	Authentication information for verifying the sender.	M
SYSINFO	A description of the system information of MA.	M

8.3.7 **HLEAVE**

8.3.7.1 This is used to announce to its local network that HMA is leaving the RMCP-3 session. Figure 53 illustrates the format of this message.

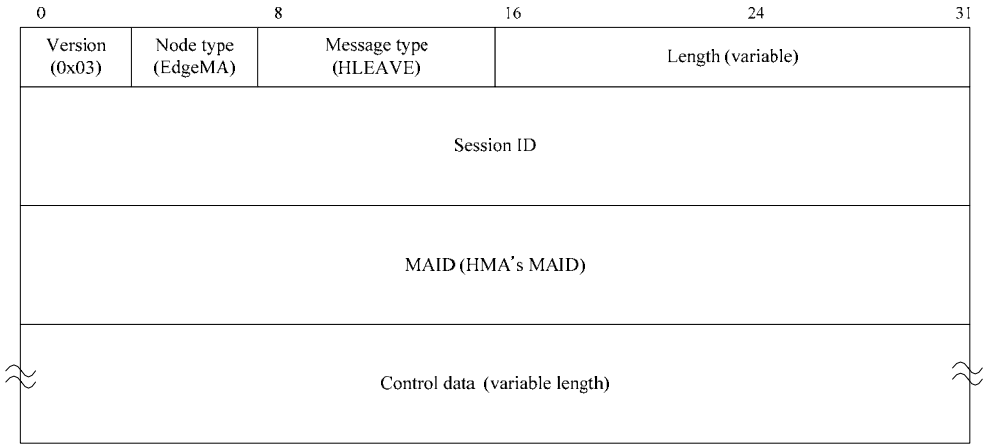

Figure 53 – HLEAVE control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of the node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x07 (see Table 2);

d) *Length* – denotes the total length of the HLEAVE control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the HLEAVE control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.7.2 Following table shows the control data types which can be used within the HLEAVE message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
NEIGHBORLIST	A list of MAs for performing the neighbor discovery.	M
ROOTPATH	A description of the path from CoreMA.	M
AUTH	Authentication information for verifying the sender.	M
REASON	A reason for leaving of MA.	M

8.3.8 **RELREQ**

8.3.8.1 This control message is used by CMA to request data forwarding from PMA. It may include a data profile to negotiate data. After relationship is established, MA can keep the relationship by periodical exchanging RELREQ and control message. Figure 54 depicts shows the format of this message.

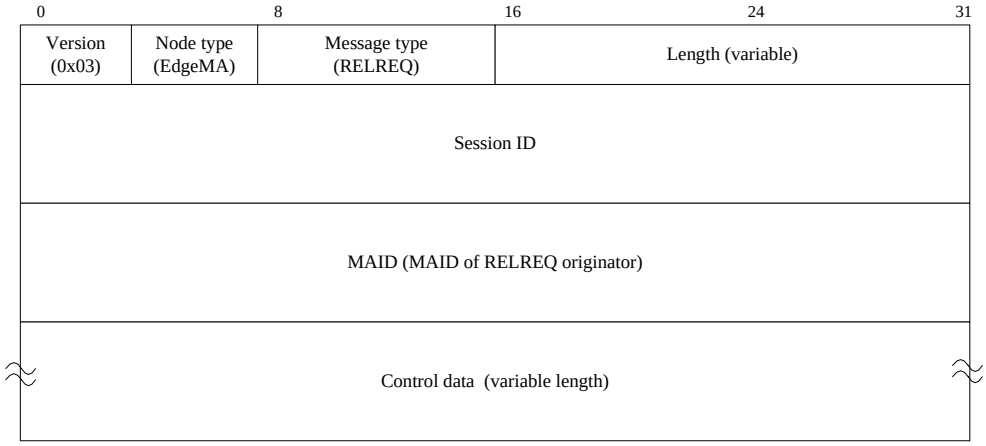

Figure 54 – RELREQ control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x08 (see Table 2);

d) *Length* – denotes the total length of the RELREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the RELRQ control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.8.2 Following table shows the control data types which can be used within the RELREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
NEIGHBORLIST	A list of MAs for performing the neighbor discovery.	M
DATAPROFILE	A description of the requirements for forwarding data.	O

8.3.9 **RELANS**

8.3.9.1 In response to RELREQ control message, RELANS control message is issued by PMA to CMA. The purpose of this message is to specify whether the relay request is allowed. It may also contain negotiated data profile. The message format of RELANS control message is shown in Figure 55.

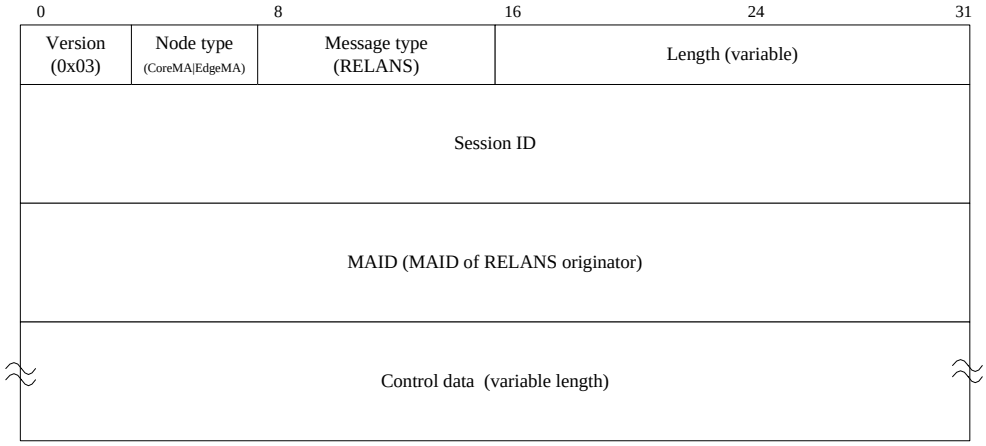

Figure 55 – RELANS control message format

a) *Version* – denotes the current version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
   - the coded value for CoreMA in Table 3;
   - the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x09 (see Table 2);

d) *Length* – denotes the total length of the RELANS control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of node sending the RELANS control message;

g) *Control data* – The Control data field may include the following information:

8.3.9.2 Following table shows the control data types which can be used within the RELANS message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
RESULT	A result of the relay request.	M
ROOTPATH	A description of the path from CoreMA.	O
DATAPROFILE	A description of the requirements for forwarding data.	O
SYSINFO	A description of the system information of MA.	O
NEIGHBORLIST	A list of MAs for performing the neighbor discovery	O

8.3.10 **STREQ**

8.3.10.1 STREQ control message is used for monitoring the status of MAs in the session. Figure 56 shows the format of this message.

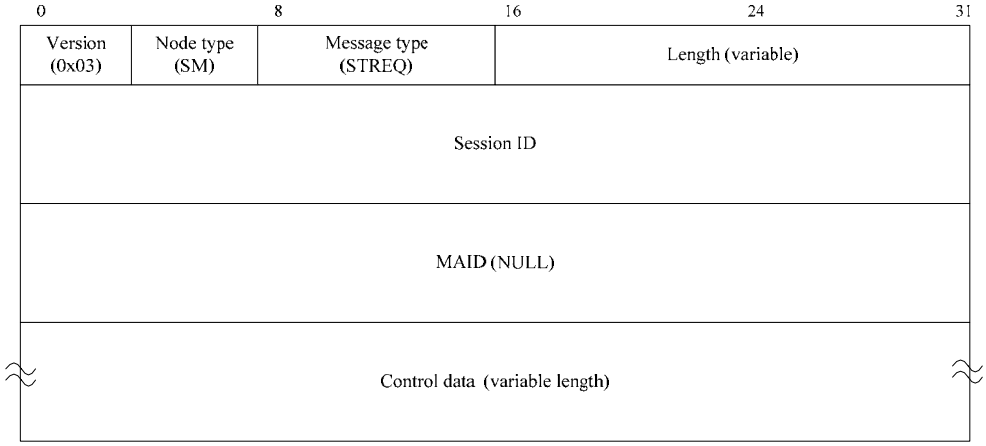

Figure 56 – STREQ control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to the coded value for SM in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x0A (see Table 2);

d) *Length* – denotes the total length of the STREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – should be set to zero because SM does not have a MAID;

g) *Control data* – The Control data field may include the following meaning and value:

8.3.10.2 Following table shows the control data types which can be used within the STREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
COMMAND	A description for requesting the specific information of MA.	M

8.3.11 **STANS**

8.3.11.1 This message is used for reporting the system information of MA. Figure 57 shows the format of the STANS control message.

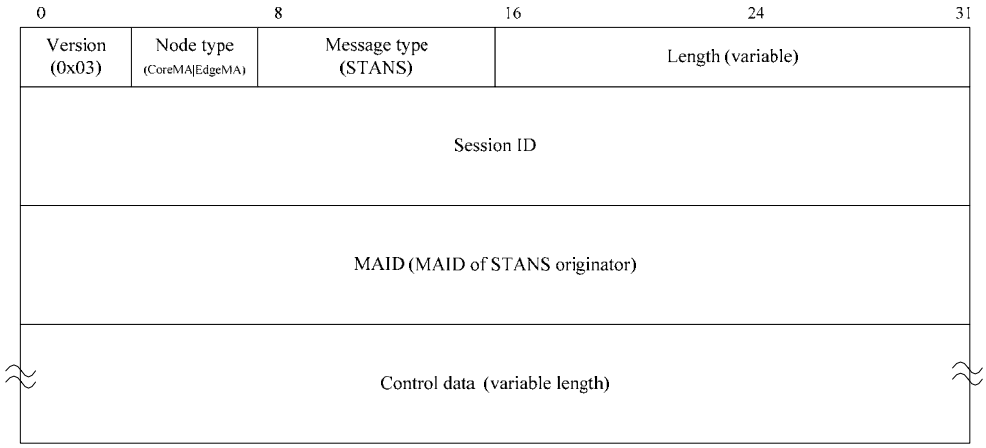

Figure 57 – STANS control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
- the coded value for CoreMA in Table 3;
- the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x0B (see Table 2);

d) *Length* – denotes the total length of the STANS control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to a the MAID of the STANS control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.11.2 Following table shows the control data types which can be used within the STANS message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
SYSINFO	A description of the system information of MA.	M

8.3.12 **LEAVREQ**

8.3.12.1 This control message is used for three different purposes, one of which is for leaving. When leaving the RMCP-3 session or its PMA for parent switching, MA sends LEAVREQ control message to the corresponding MAs based on the leaving procedure.

SM and PMA may use this control message to expel MA but their targets are different. The target of SM is an MA in the session; that of PMA is only its CMA.

Finally, this control message is used for terminating a session. When SMA leaves the session, this message should be forwarded to the endmost MA in the tree hierarchy. Figure 58 illustrates the format of the LEAVREQ control message.

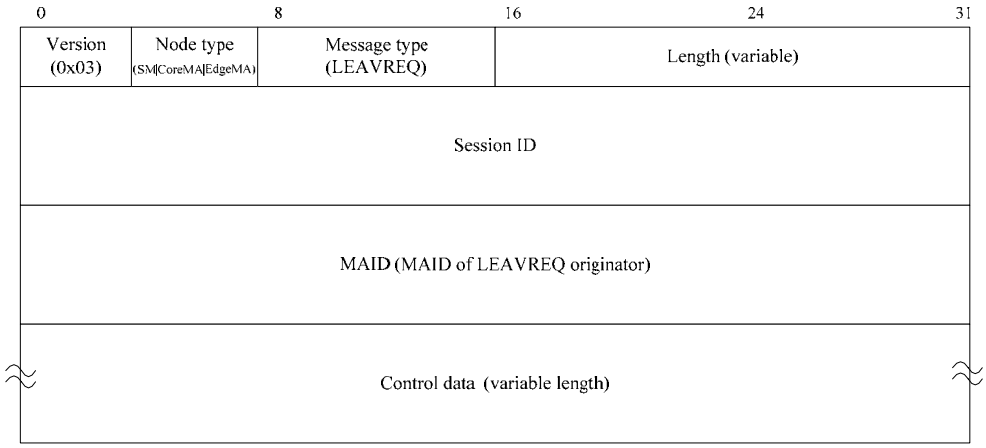

Figure 58 – LEAVREQ control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
- the coded value for SM in Table 3 for EdgeMA expulsion from the session;
- the coded value for CoreMA in Table 3 for EdgeMA expulsion from the requestor;
- the coded value for EdgeMA in Table 3 for notifying session leaving or parent switching;

c) *Message type* – denotes the type of the message. The value shall be set to 0x0C (see Table 2);

d) *Length* – denotes the total length of the LEAVREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the LEAVREQ control message originator (set to zero for SM);

g) *Control data* – The Control data field of the message may include the following information:

8.3.12.2 Following table shows the control data types which can be used within the LEAVREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
REASON	A reason for leaving of MA.	M

8.3.13 **LEAVANS**

8.3.13.1 As a confirmation of the LEAVREQ control message, LEAVANS control message is sent back by the MA receiving LEAVREQ control message. Figure 59 illustrates the format of the LEAVANS control message.

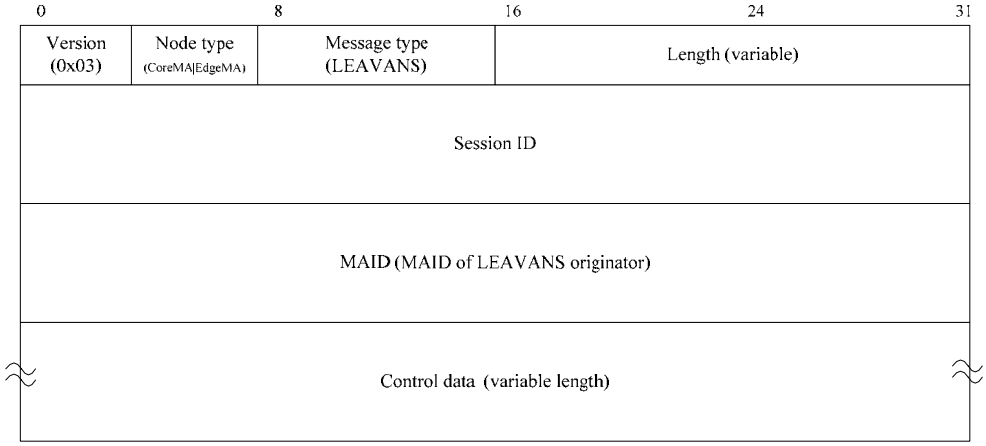

Figure 59 – LEAVANS control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) Node type – The message type of node sending the message. The value shall be set to
   - the coded value for CoreMA in Table 3;
   - the coded value for EdgeMA in Table 3;

c) Message type – The type of the message. The value shall be set to 0x0D (see Table 2);

d) Length – The total length of the LEAVANS control message including control data (in bytes);

e) Session ID – A 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) MAID – The MAID of the LEAVANS control message sender;

g) Control data – The Control data field of the message may include the following information:

8.3.13.2 Following table shows the control data types which can be used within the LEAVANS message. Details of following control data are described in Clause 8.4.

Control type	**Meaning**	**M/O**
RESULT	A result of leave request.	M

8.3.14 **TERMREQ**

8.3.14.1 TERMREQ control message is used to terminate an existing RMCP-3 session. It is issued by the SM and subsequently forwarded by CoreMA to the endmost MAs along the tree hierarchy. Figure 60 shows the format of the TERMREQ control message.

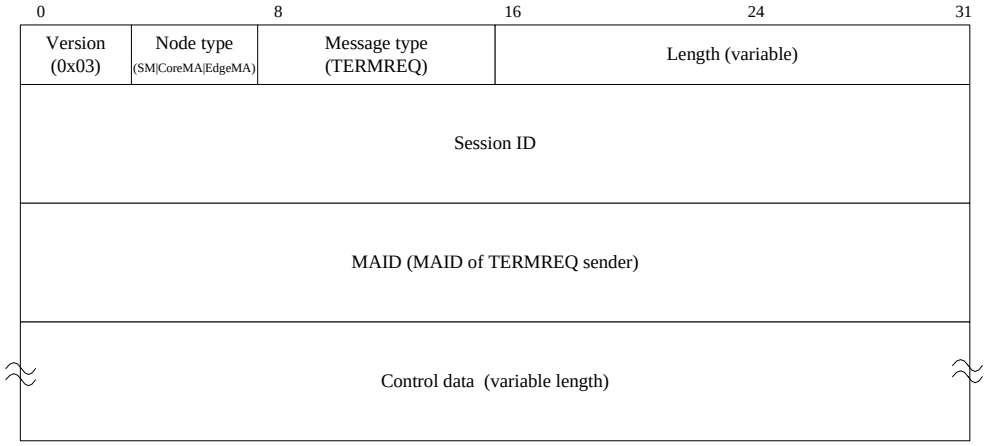

Figure 60 – TERMREQ control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
   - the coded value for SM in Table 3;
   - the coded value for CoreMA in Table 3;
   - the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x0E (see Table 2);

d) *Length* – denotes the total length of the TERMREQ control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as define in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the TERMREQ control message sender (set to zero for SM);

g) *Control data* – The Control data field may include the following information:

8.3.14.2 Following table shows the control data types which can be used within the TERMREQ message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
REASON	A reason for terminating the session.	M

8.3.15 **TERMANS**

8.3.15.1 Figure 61 illustrates the format of the TERMANS control message.

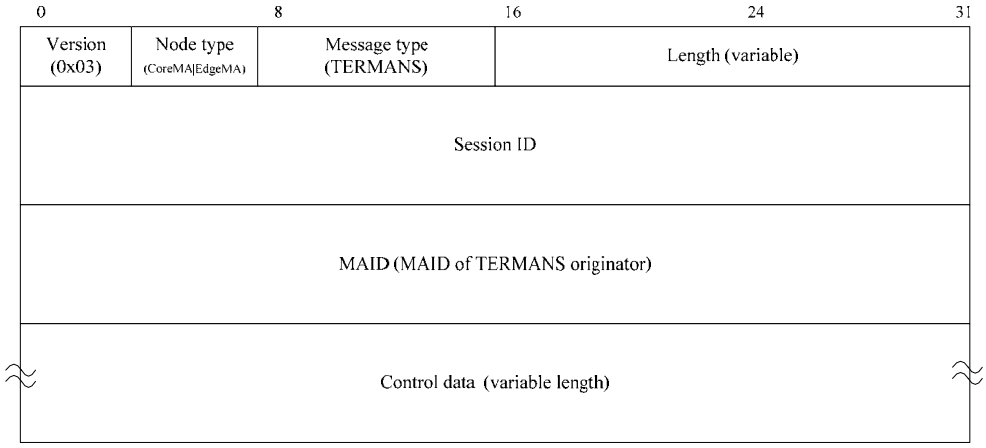

Figure 61 – TERMANS control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
- the coded value for CoreMA in Table 3;
- the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x0F (see Table 2);

d) *Length* – denotes the total length of the TERMANS control message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the TERMANS control message sender;

g) *Control data* – The Control data field may include the following information:

8.3.15.2 Following table shows the control data types which can be used within the TERMANS message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
RESULT	A result of the termination request.	M

8.3.16 **VHB**

8.3.16.1 VHB control message is issued by CoreMA periodically to give clock information through the RMCP-3 session. With VHB control message, each EdgeMA can diagnose the network condition. Figure 62 illustrates the format of the VHB control message.

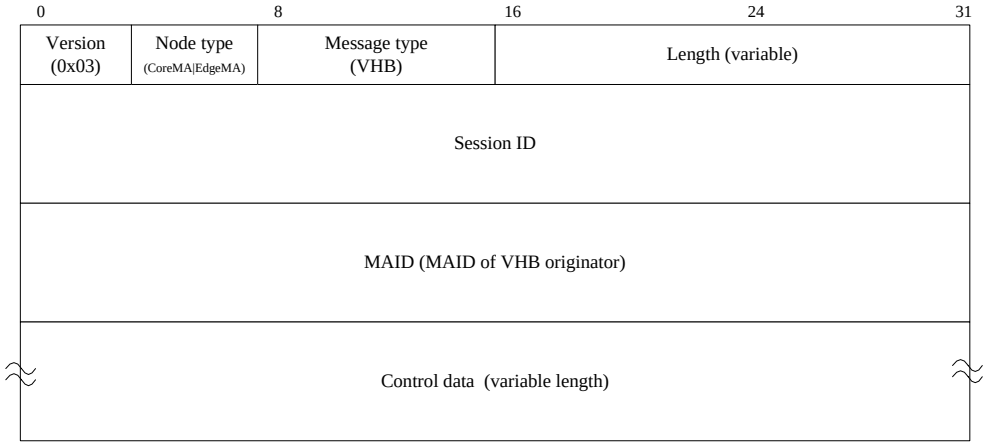

Figure 62 – VHB control message format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
   - the coded value for CoreMA in Table 3;
   - the coded value for EdgeMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x10 (see Table 2);

d) *Length* – denotes the total length of the VHB message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as defined in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the VHB originator. Although VHB is forwarded by PMA to CMA, this field is not changed by the intermediate relaying PMA;

g) *Control data* – The Control data field may include the following information:

8.3.16.2 Following table shows the control data types which can be used within the VHB message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
ROOTPATH	A description of the path from CoreMA. By using RP_PSEUDO sub-control data, the ROOTPATH control data can indicate that ROOTPATH in VHB is a pseudo ROOTPATH.	M
AUTH	Authentication information for verifying message sender.	O

8.3.17 **HHB**

8.3.17.1 Figure 63 shows the format of HHB control message which is used by CoreMA to share CoreRing information and to make the CoreRing robust.

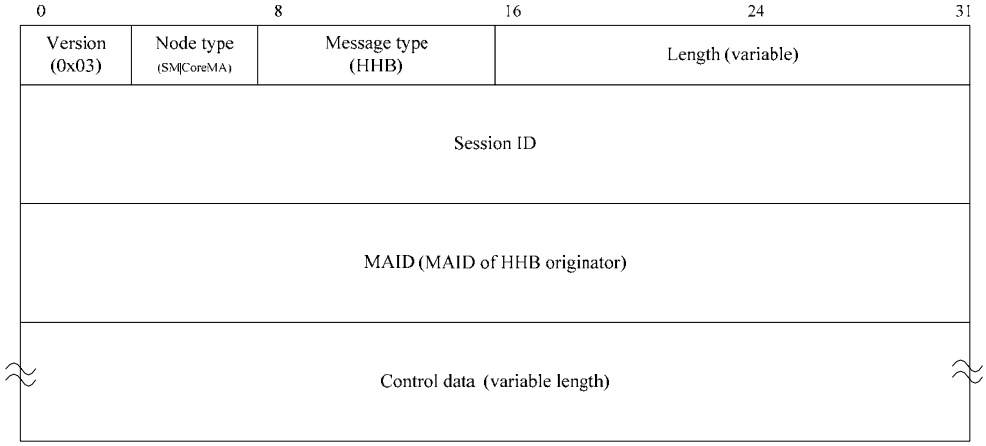

Figure 63 – HHB control message format

The meaning of each field and its values are:

a) *Version* – denotes the version of RMCP. Its value shall be 0x03;

b) *Node type* – denotes the type of node sending the message. The value shall be set to
- the coded value for SM in Table 3;
- the coded value for CoreMA in Table 3;

c) *Message type* – denotes the type of the message. The value shall be set to 0x11 (see Table 2);

d) *Length* – denotes the total length of the HHB message including control data (in bytes);

e) *Session ID* – shall be set to a 64-bit integer value that identifies a session as *defined* in Clause 9.1.1;

f) *MAID* – shall be set to the MAID of the HHB message originator (set to zero indicating SM);

g) *Control data* – The Control data field may include the following information:

8.3.17.2 Following table shows the control data types which can be used within the HHB message. Details of following control data are described in Clause 8.4.

Control type	Meaning	M/O
ROOTPATH	The description of the path from SM. By using RP_PSEUDO sub-control data, the ROOTPATH control data can indicate that ROOTPATH in HHB is a pseudo ROOTPATH.	M
SYSINFO	A description of the system information of CoreMA.	O

8.4 **RMCP-3 control data**

8.4.1 **SYSINFO control data**

8.4.1.1 This control data specifies the system information of MA, e.g. in/out bandwidth, controllable number of CMAs.

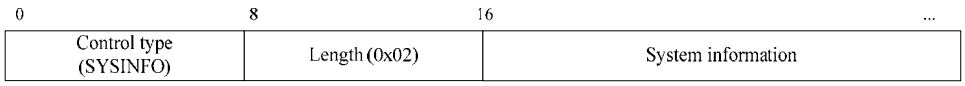

Figure 64 – Control data – SYSINFO

a) *Control type* – denotes the type of the control data. The value shall be set to 0x08 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x02 which means 2-byte;

c) *System information* – may include the following sub-control data.

The sub-control data that may follow the SYSINFO control data are shown in Figure 65 through Figure 79.

8.4.1.2 Figure 65 shows the SI_POS_BW sub-control data format.

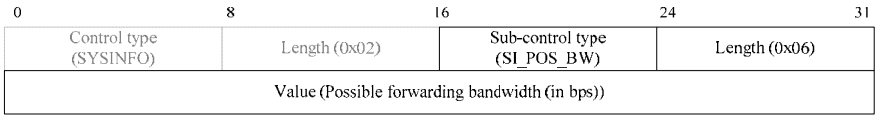

Figure 65 – Sub-control data – SI_POS_BW

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x25 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byte;

c) *Value* – shall be set to the possible forwarding bandwidth that MA can offer.

8.4.1.3 Figure 66 shows the SI_IP sub-control data. Each field has the following meaning and value:

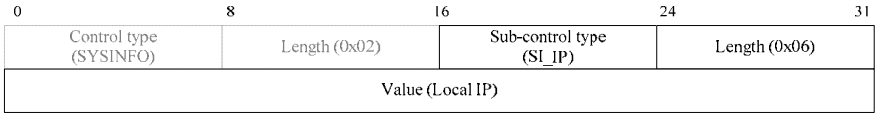

Figure 66 – Sub-control data – SI_IP

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x11 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byte;

c) *Value* – shall be set to the IP address of local host.

8.4.1.4 The SI_UPTIME sub-control data is shown in Figure 67. It can be used as the lifetime of HMA or the report on the system uptime since MA joined the session. Each field has the following meaning and value:

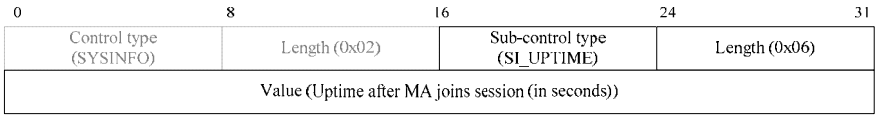

Figure 67 – Sub-control data – SI_UPTIME

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x12 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byte;

c) *Value* – shall be set to the time after the node joins the RMCP-3 session (in seconds).

8.4.1.5 The SI_DELAY sub-control data is shown in Figure 68. It can be used to tell the delay as perceived by EdgeMA from root node of EdgeTree which the EdgeMA belongs. Each field has the following meaning and value:

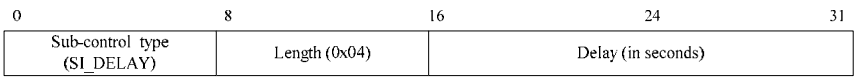

Figure 68 – Sub-control data – SI_DELAY

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x13 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Value* – shall be set to the delay value as perceived by EdgeMA from root node of EdgeTree which the EdgeMA belongs (in seconds).

8.4.1.6 Figure 69 shows the SI_ROOM_CMA sub-control data. It can be used to report the room for CMAs. Each field has the following meaning and value:

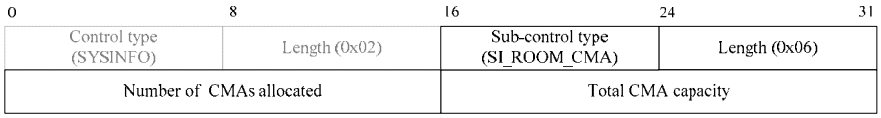

Figure 69 – Sub-control data – SI_ROOM_CMA

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x14 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byes;

c) *Number of CMAs allocated* – shall be set to the number of allocated rooms for the CMAs;

d) *Total CMA capacity* – shall be set to the total CMA capacity. Thus the available number of rooms for CMA will be the difference between the number of CMAs allocated and total CMA capacity.

8.4.1.7 Figure 70 shows the report on the bandwidth that can be provided by a system. Each field has the following meaning and value:

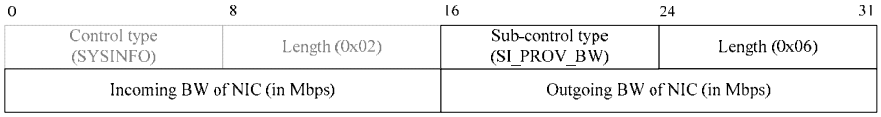

Figure 70 – Sub-control data – SI_PROV_BW

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x15 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byte;

c) *Incoming BW of NIC* – shall be set to the maximum incoming bandwidth of network interface card;

d) *Outgoing BW of NIC* – shall be set to the maximum outgoing bandwidth of network interface card.

8.4.1.8 The SI_SND_BW sub-control data is shown in Figure 71. It provides the information about the total bandwidth consumed by MA to serve its CMAs. Each field has the following meaning and value:

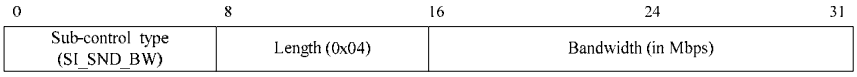

Figure 71 – Sub-control data – SI_SND_BW

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x35 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Bandwidth* – shall be set to the total bandwidth consumed by MA to serve its CMAs (in Mbps).

8.4.1.9 The SI_SND_PACKET sub-control data is shown in Figure 72. It tells the total number of packets sent by MA from startup. Each field has the following meaning and value:

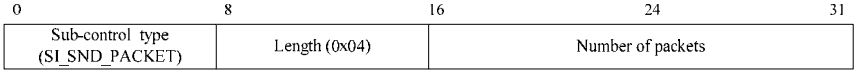

Figure 72 – Sub-control data – SI_SND_PACKET

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x36 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Number of packets* – shall be set to the total number of packets sent by the MA from startup.

8.4.1.10 The SI_SND_BYTES sub-control data is shown in Figure 73. It can be used to tell the total number of bytes sent by MA from startup. Each field has the following meaning and value:

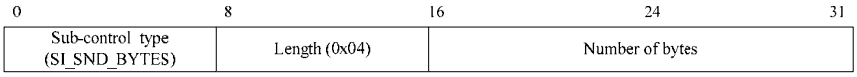

Figure 73 – Sub-control data – SI_SND_BYTES

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x37 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Number of bytes* – shall be set to the total number of bytes sent by the MA from startup.

8.4.1.11 The SI_RCV_BW sub-control data is shown in Figure 74. It can be used to tell the bandwidth perceived by MA between its PMA. Each field has the following meaning and value:

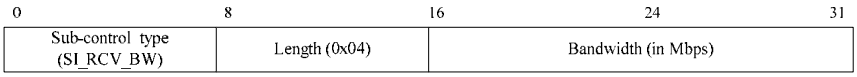

Figure 74 – Sub-control data – SI_RCV_BW

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x45 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Bandwidth* – shall be set to the bandwidth perceived by MA between its PMA (in Mbps).

8.4.1.12 The SI_RCV_PACKET sub-control data is shown in Figure 75. It tells the total number of packets received by MA from startup. Each field has the following meaning and value:

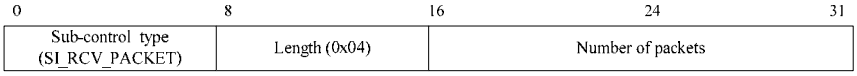

Figure 75 – Sub-control data – SI_RCV_PACKET

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x46 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Number of packets* – shall be set to the total number of packets received by MA from startup.

8.4.1.13 The SI_RCV_BYTES sub-control data is shown in Figure 76. It can be used to tell the total number of bytes received by MA from startup. Each field has the following meaning and value:

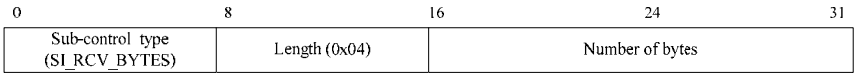

Figure 76 – Sub-control data – SI_RCV_BYTES

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x47 (see Figure 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x04 which means 4-byte;

c) *Number of bytes* – shall be set to the number of bytes received by MA from startup.

8.4.1.14 Figure 77 shows SI_REL_BYTES sub-control data. It can be used within the HHB control message to report the number of bytes of the relayed data. Each field has the following meaning and value:

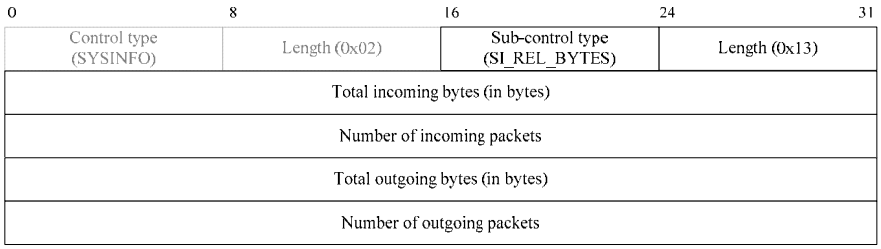

Figure 77 – Sub-control data – SI_REL_BYTES

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x57 (see Table 5);

b) *Length* – denotes the length of sub-control data. The value shall be set to 0x13 which means 18-byte;

c) *Total incoming bytes* – shall be set to the total bytes of incoming data;

d) *Number of incoming packet* – shall be set to the total number of incoming packets;

e) *Total outgoing bytes* – shall be set to the total bytes of outgoing data;

f) *Number of outgoing packet* – shall be set to the total number of outgoing packets.

8.4.1.15 Figure 78 shows the report on the status of tree PMA and CMAs of EdgeMA. Each field has the following meaning:

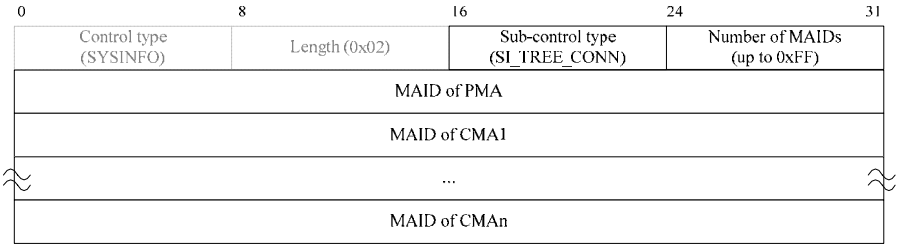

Figure 78 – Sub-control data – SI_TREE_CONN

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x68 (see Table 5);

b) *Number of MAIDs* – denotes the number of MAIDs in the list. The value shall be set to n+1 in hexadecimal. Since the length of this field is 8-bit, maximum value of this field is 0xFF which means that 255 MAIDs, one for PMA and 254 MAIDs for CMAs, are included in the SI_TREE_CONN sub-control data;

c) *MAID of PMA* – shall be set to the MAID of directly attached PMA;

d) *MAID of CMA n* – shall be set to the MAID of n-th directly attached CMA.

8.4.1.16 Figure 79 shows the report on the member of tree. Each field has the following meaning:

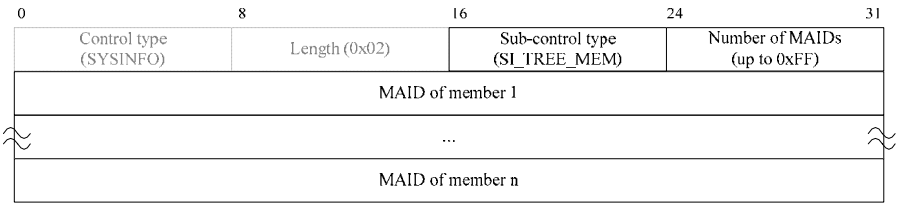

Figure 79 – Sub-control data – SI_TREE_MEM

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x69 (see Table 5);

b) *Number of MAIDs* – denotes the number of MAIDs in the list. The value shall be set to n in hexadecimal. Since the length of this field is 8-bit, maximum value of this field is 0xFF which means that there are 255 members in the tree;

c) *MAID of member n* – shall be set to the MAID of n-th tree member.

8.4.2 DATAPROFILE control data

8.4.2.1 DATAPROFILE control data delivers the controllable data profile of each MA. DATAPROFILE control data may be used within control messages for negotiating data channel.

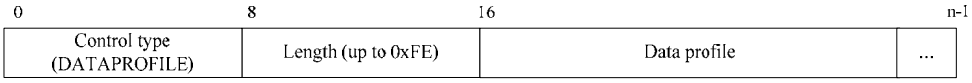

Figure 80 – Control data – DATAPROFILE

Each field has the following meaning and information:

a) *Control type* – denotes the type of control data. The value shall be set to 0x03 (see Table 4);

b) *Length* – denotes the length of the control data. The value shall be set to n/8 in hexadecimal which means the total length of the DATAPROFILE control data in byte. Since the length of this field is 8-bit, maximum value of this field is 0xFF which means the length of the DATAPROFILE control data is 255-byte including 253-byte of the "Data profile" field. But, since the sum of the length of the Data profile field and the length of padding field is aligned to multiple of 4-byte, maximum value of the Length field can be 0xFE;

c) *Data profile* – denotes the data profile that MA wants to use. Data profile is the description of the characteristics of the data channel. It follows the SDL-like encoding scheme;

d) *Zero or more padding* – Since Data profile consists of a text-based variable message, the size may vary. To align a length of 4 bytes, each data profile pads zero or more 1-byte zero padding.

8.4.3 AUTH control data

8.4.3.1 Authentication information is delivered using AUTH control data. The authentication algorithm used is defined in AUTH_ALG field. AUTH_ALG code is defined in Table 8.

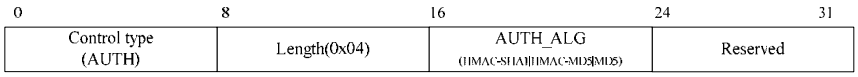

Figure 81 – Control data – AUTH

Each field has the following meaning and information:

a) *Control type* – denotes the type of control data. The value shall be set to 0x01 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x04 which means 4-byte;

c) *AUTH_ALG* – denotes the type of authentication algorithm. The value shall be set to
   - the coded value for HMAC-SHA1 in Table 8;
   - the coded value for HMAC-MD5 in Table 8;
   - the coded value for MD5 in Table 8;

d) *Reserved* – reserved for the further use.

8.4.4 RESULT control data

8.4.4.1 This control message specifies whether MA's request is successful or not. If MA's request is successful, the OK code is includes within the Result code field. Otherwise, an appropriate error code is given. Figure 82 shows the format of RESULT control data.

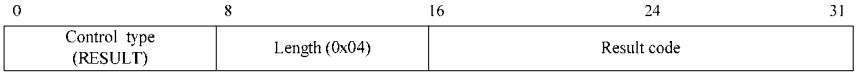

Figure 82 – Control data – RESULT

Each field has the following meaning and value:

a) *Control type* – denotes the type of control data. The value shall be set to 0x05 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x04 which means 4-byte;

c) *Result code* – denotes the result of the request. The codes and their meaning are listed in Table 11.

8.4.5 NEIGHBORLIST control data

8.4.5.1 If a subscription is successful, SM gives neighbor lists which include MAID of whole active CoreMAs back to the subscriber. The NEIGHBORLIST control data can be used as bootstrap information by each subscriber. Figure 83 shows the format of NEIGHBORLIST, note that it only delivers MAID.

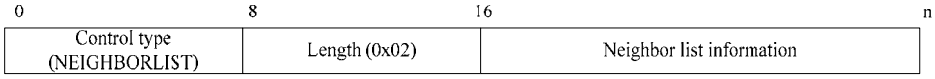

Figure 83 – Control data – NEIGHBORLIST

a) *Control type* – denotes the type of control data. The value shall be set to 0x04 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x02 which means 2-byte;

c) *Neighbor list information* – denotes the series of information on MAIDs. The following are the usage and format:

8.4.5.2 Figure 84 shows the sub-control data that follows NEIGHBORLIST control data. Each field has the following meaning and value:

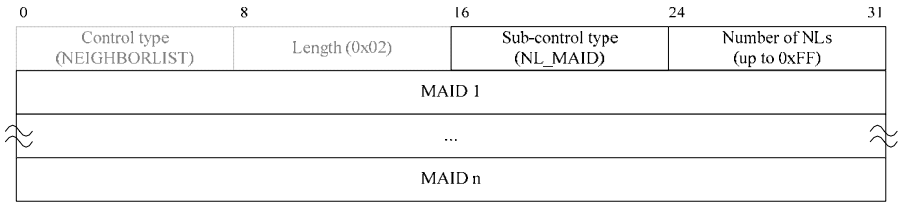

Figure 84 – Sub-control data – NL_MAID

a) *Sub-control type* – denotes the type of sub-control data. The value shall be set to 0x01 (see Table 6);

b) *Number of NLs* – denotes the number of MAIDs. Since the length of the Number of NLs field is 8-bits, maximum value of this field is 0xFF which means that there are 255 MAID of neighbors;

c) *MAID n* – shall be set to the MAID of n-th neighbor.

8.4.6 ROOTPATH control data

8.4.6.1 To prevent loop and solve the tri-angular problem, the probed MA must include its root path using ROOTPATH control data shown in Figure 85. Each field has the following meaning and value:

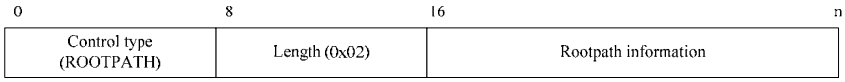

Figure 85 – Control data – ROOTPATH

a) *Control type* – denotes the type of control data. The value shall be set to 0x07 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x02 which means 2-byte;

c) *Rootpath information* – This field includes rootpath information. The following are the format and usage:

8.4.6.2 Figure 86 shows the sub-control data of ROOTPATH control data.

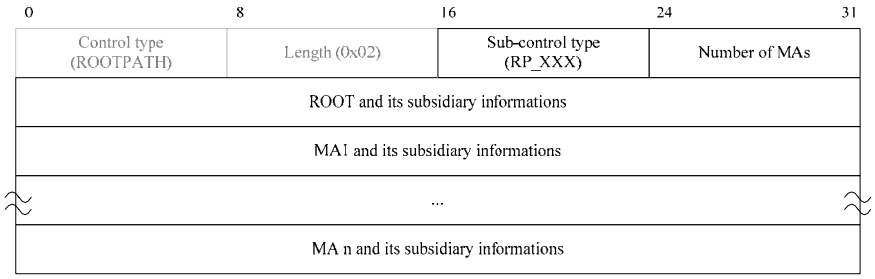

Figure 86 – Sub-control data – RP_XXX

Each field has the following meaning and value:

a) *Sub-control type* – denotes the type of sub-control data. The codes and their meaning are listed in Table 7;

b) *Number of MAs* – denotes the number of MAs on the rootpath;

c) *One or more information* – denotes the information about hop according to sub-control type. The size of each field is fixed and can be calculated by combination of each type length.

8.4.7 TIMESTAMP control data

8.4.7.1 Figure 87 shows the TIMESTAMP control data used to examine the distance between two MAs.

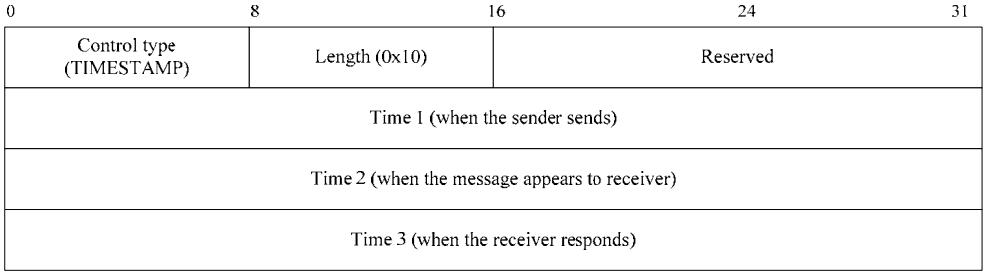

Figure 87 – Control data – TIMESTAMP

a) *Control type* – denotes the type of control data. The value shall be set to 0x09 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x10 which means 16-byte;

c) *Reserved* – Reserved for the further use;

d) *Time1* – shall be set to the time when the message is sent to its counterpart;

e) *Time2* – shall be set to the time when the message appears to the counterpart;

f) *Time3* – shall be set to the time when the receiver of the message sends the TIMESTAMP control data in response.

8.4.8 REASON control data

8.4.8.1 To specify the reason for leaving of MA, the LEAVREQ/HLEAVE control message must include REASON control data. TERMREQ control message must include REASON control data to specify the reason for terminating the session. Figure 88 shows the REASON control data format.

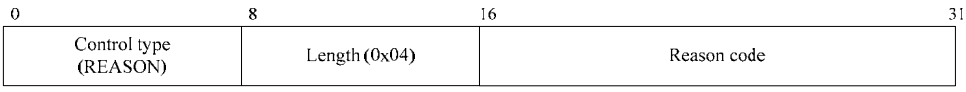

Figure 88 – Control data – REASON

a) *Control type* – denotes the type of control data. The value shall be set to 0x05 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x04 which means 4-byte;

c) *Reason code* – denotes an integer value to indicate the specific reason for leaving. The encoded value and its meaning follow the codes specified in Table 9 and Table 10

8.4.9 COMMAND control data

8.4.9.1 COMMAND control data is used to request the specific information of MA. The STREQ control message should include COMMAND control data to specify what status report it needs. Figure 89 shows the COMMAND control data format.

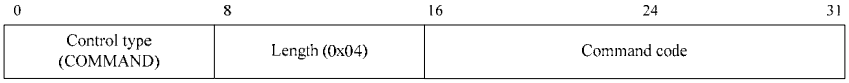

Figure 89 – Control data – COMMAND

a) *Control type* – denotes the type of control data. The value shall be set to 0x02 (see Table 4);

b) *Length* – denotes the length of control data. The value shall be set to 0x04 which means 4-byte;

c) *Command code* – denotes the value to indicate the specific command. The encoded value and its meaning are same as specified in Table 5.

9 Parameters

9.1 RMCP-3 identifiers

9.1.1 Session ID

Session ID (SID) is generated with a combination of the local IP address of the Session Manager (SM) and the group address of the session. The SM allocates a group address to a new session when it is requested to create a session. The group address is created as a unique value for the session without duplication with any session it manages.

Figure 90 illustrates the format of SID in RMCP-3.

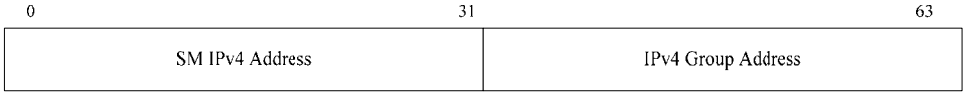

Figure 90 – Format of Session ID

9.1.2 MAID

MAID consists of the local IP address, port number, and serial number as Figure 91 shows. The local IP address is the IP address of MA. An MA in a RMCP-3 session may have to open several ports for the session. The port number used for generation of its MAID is a listening port number opened when the MA starts to run RMCP-3 in order to receive control messages from SM or other MAs.

Each MA can be identified by its port number in a multi-user system. It is, however, not possible to identify each MA inside of a Network Address Translation (NAT) based network, where it may show the same IP address for multiple MAs to the communication peer outside of the network. To handle this case, SM generates a unique MAID as it fills in a unique value in the serial number field when it receives a NAT address from an MA, and returns the ID to the MA.

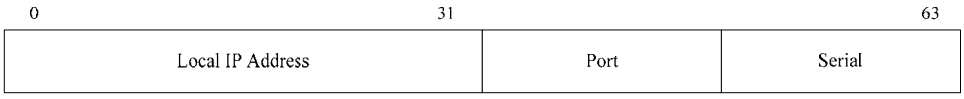

Figure 91 – Format of MAID

The following Figure 92 is the simple algorithm that the current version of RMCP-3 uses to generate a unique MAID.

If the IP address in the received MAID is a NAT address

```
        Search for its NAT_address_list;
        if there already exists the same address
                serial_number++;
            else
                add the list into NAT_address_list
                serial_number++;
        MAID = IP_address + port_number + serial_number;
return MAID;
```

Figure 92 – Simple algorithm to generate a unique MAID

9.2 Parameters used in RMCP-3

This clause defines encoding of the following:

a) RMCP-3 control message types
b) RMCP-3 node types
c) Control data type
d) Sub-control data type

9.2.1 RMCP-3 control message types

Table 2 lists the types of RMCP-3 control message and the corresponding encoded values.

Table 2 – RMCP-3 control message types

Message type	Code (8 bits)
SUBSREQ	0x01
SUBSANS	0x02
PPROBREQ	0x03
PPROBANS	0x04
HSOLICIT	0x05
HANNOUNCE	0x06
HLEAVE	0x07
RELREQ	0x08
RELANS	0x09
STREQ	0x0A
STANS	0x0B
LEAVREQ	0x0C
LEAVANS	0x0D
TERMREQ	0x0E
TERMANS	0x0F
VHB	0x10
HHB	0x11

9.2.2 Node types

Table 3 lists the RMCP-3 nodes and corresponding encoded values.

Table 3 – RMCP-3 node types

Node type	Code (4 bits)
SM	0x1
CoreMA	0x2
EdgeMA	0x3

9.2.3 Control data types

Table 4 lists the codes of RMCP-3 control data.

Table 4 – RMCP-3 control data types

Control data type	Code (8 bits)
AUTH	0x01
COMMAND	0x02
DATAPROFILE	0x03
NEIGHBORLIST	0x04
REASON	0x05
RESULT	0x06
ROOTPATH	0x07
SYSINFO	0x08
TIMESTAMP	0x09

9.2.4 Sub-control data types

A single control data may include zero or more sub-control data. This clause defines codes of RMCP-3 sub-control data.

9.2.4.1 SYSINFO control data is used for describing information related to MA. When SM sends STREQ message, sub-control data of SYSINFO control data can also be used by COMMAND control data for requesting specific information. Table 5 lists the possible sub-control data type and its encoded value and meaning. The four most significant bits of the encoded code specify the category of the information, with the lowest four bits specifying the detailed items such as bandwidth, packets, and bytes.

Table 5 – RMCP-3 sub-control data types (SYSINFO)

Type	Code (8 bits)	Meaning
SI_IP	0x11	IP address of MA.
SI_UPTIME	0x12	Time of MA’s uptime.
SI_DELAY	0x13	Status of delay as perceived by EdgeMA from root node of EdgeTree which the EdgeMA belongs.
SI_ROOM_CMA	0x14	The room for CMAs.
SI_PROV_BW	0x15	Maximum incoming / outgoing bandwidth of MA’s network interface card.
SI_POS_BW	0x25	The possible forwarding bandwidth that MA can afford.
SI_SND _BW	0x35	Total bandwidth consumed by MA to serve its CMAs.
SI_SND_PACKET	0x36	Total number of packets sent by MA from startup.
SI_SND _BYTES	0x37	Total number of bytes sent by MA from startup.
SI_RCV_BW	0x45	Bandwidth perceived by MA between its PMA.
SI_RCV_PACKET	0x46	Number of packets received by MA from startup.

SI_RCV_BYTES	0x47	Number of bytes received by MA from startup.
SI_REL_BYTES	0x57	The number of bytes of the relayed data.
SI_TREE_CONN	0x68	PMA and CMA(s) of MA.
SI_TREE_MEM	0x69	List of tree members.

9.2.4.2 NEIGHBORLIST control data is used for describing information related to the RMCP-3 neighbors. Table 6 lists the possible sub-control data type and its encoded value and meaning.

Table 6 – RMCP-3 sub-control data types (NEIGHBORLIST)

Type	Code (8 bits)	Meaning
NL_MAID	0x01	List of MAs; MA includes both CoreMA and EdgeMA

9.2.4.3 ROOTPATH control data is used to describe the path between two end-points. The path consists of MAs passing through between the two end-points along the RMCP-3 Hybrid Tree. Table 7 lists the possible sub-control data types and their encoded value and meaning.

Table 7 – RMCP-3 sub-control data types (ROOTPATH)

Type	Code (8 bits)	Meaning
RP_ID	0x11	The following ROOTPATH contains only the MAID of each hop (8 bytes each).
RP_ BW	0x12	The following ROOTPATH contains only the bandwidth by hop (4 bytes each).
RP_ DL	0x14	The following ROOTPATH contains only the delay perceived by each hop (4 bytes each).
RP_ID _BW	0x13	The following ROOTPATH contains the MAID and bandwidth of each hop (12 bytes each).
RP_ID _DL	0x15	The following ROOTPATH contains the MAID and corresponding delay of each hop (12 bytes each).
RP_ID_BW_DL	0x17	The followed ROOTPATH contains the MAID, bandwidth, and delay of each hop (16 bytes each).
RP_PSEUDO	0x10	The following ROOTPATH is a pseudo-ROOTPATH for fault recovery (8 bytes each).

9.3 Encoding rules to represent values used in RMCP-3

This clause defines various codes representing reasons, results, and authentication algorithm.

9.3.1 Authentication algorithm

AUTH control data is used to specify the authentication algorithm to be used. Table 8 lists the possible authentication algorithms for RMCP-3 and their encoded value and reference.

Table 8 – Authentication algorithm for RMCP-3 (AUTH_ALG)

Type	Code (8 bits)	Reference
HMAC-SHA1	0x01	IETF RFC 2104
HMAC-MD5	0x02	IETF RFC 2104
MD5	0x03	IETF RFC 1321

9.3.2 Reason for leaving

Table 9 lists the various reasons for leaving of MA. The four most upper bits specify the main cause of leaving, with the four lowest bits specifying the detailed reasons for leaving. Through the code for the reason for leaving, MA can express the reason for leaving explicitly.

Table 9 – Code for reason for leaving

Category	Value	Code (8 bits)	Meaning
Leave	MA_LEAVE	0x10	MA's own leaving
	SMA_LEAVE	0x11	SMA leaving
Kick out	SM_KICKOUT	0x20	SM kick out
	PMA_KICKOUT	0x21	PMA kick out
Parent switching	PA_SWITCH	0x40	Parent switching by MA

9.3.3 Reason for termination

Table 10 lists the reason for session termination. The four most significant bits specify the main reason for session termination, with the four lowest bits specifying the reasons.

Table 10 – Code for reason for termination

Category	Value	Code (8 bits)	Meaning
Normal session termination	NORM_TERM	0xE0	Session is terminated normally.
Abnormal session termination	NOREA_TERM	0xF0	Session is terminated abnormally for no reason.
	USER_TERM	0xF1	Session is terminated abnormally by user request.

9.3.4 Result code

Table 11 lists the results. These codes are included in the return message to specify the result of a specific request.

Table 11 – Result codes

Value	Code (8 bits)	Meaning
RE_OK	0x10	OK
RE_SYSPROB	0x20	System problem
RE_ADMPROB	0x30	Administrative problem

9.4 Timers and their parameters

Manipulating the RMCP-3 Hybrid Tree requires RMCP-3 to keep several timers and their related parameters. This clause defines various timers and parameters.

9.4.1 Parameters for neighbor discovery

Every MA in RMCP-3 performs a neighbor discovery procedure during the RMCP-3 session; each MA periodically exchanges PPROBREQ and PPROBANS control messages with its neighboring MAs for EdgeTree refinement. The following timer and number are related to the neighbor discovery procedure which is described in Clause 7.3.2:

Table 12 – Parameter for neighbor discovery

Value	Meaning	Default value	Description
T_PPROBE	Parent probe timer	45	This timer should be kept by each EdgeMA to issue a periodic PPROBREQ control message. At every assigned time period, it reminds EdgeMA to issue a PPROBREQ control message. The default value for the T_PPROBE is 45 seconds, although it can be changed arbitrarily by each EdgeMA.
N_PPROBE	Number of parent probing	1	This parameter limits the maximum number of PPROBREQ messages that can be sent by each EdgeMA simultaneously to prevent PPROBREQ implosion. The default value for N_PPROBE is 1, but it can be changed arbitrarily for quick completion of parent probing.

9.4.2 Parameters for heartbeat

RMCP-3 uses the heartbeat mechanism to maintain the resilience of RMCP-3 Hybrid Tree. This is because the heartbeat mechanism can detect any network fault such as loops and partitions. The RMCP-3 heartbeat mechanism uses both HHB and VHB -- HHB for CoreRing, VHB for EdgeTree. RMCP-3 defines the following parameters to support the heartbeat mechanism:

Table 13 – Parameter for heartbeat

Value	Meaning	Default value	Description
T_HB	Heartbeat timer	15	This timer is required by SM, CoreMA, and EdgeMA. The T_HB for SM is used to issue the HB control message (including VHB and HHB) and to detect network fault; that for CoreMA and that for EdgeMA are used to detect network fault. The default value for T_HB is 15 seconds. The HB timer should be synchronized according to every RMCP-3 entity.
N_HB	Number of heartbeat	3	This value is used to check whether the EdgeTree or CoreRing is partitioned. If it does not receive the HB control message (including VHB and HHB) for a specific time (N_HB * T_HB), MA can detect partition. The default value of N_HB is 3.

9.4.3 Parameters for report

STREQ and STANS control messages are used to monitor the RMCP-3 session. To prevent SM from waiting for the STANS control message infinitely, RMCP-3 limits the waiting time. RMCP-3 defines the following parameter for limiting the STANS control message waiting time:

Table 14 – Parameter for report

Value	Meaning	Default value	Description
T_REPORT	Report timer	15	This timer is required by SM to limit the waiting time for STANS control message. After sending STREQ control message to a specific MA, SM starts this timer. SM assumes that MA does not operate any more if STANS control message does not arrive until the timer expires. The default value for T_REPORT is 15 seconds, but it can be changed arbitrarily.

9.4.4 Parameters for HMA selection

The following parameters are used to support the HMA mechanism (HMA in RMCP-3 is a very important entity for using the IP multicast data transmission in a local multicast-enabled network):

Table 15 – Parameter for HMA selection

Value	Meaning	Default value	Description
T_HSOLICIT	HSOLICIT timer	2	This defines the time period of the HSOLICIT control message. An EdgeMA in the local multicast domain sends the HSOLICIT control message at every T_HSOLICIT. The default value for T_HSOLICIT is 2 seconds.
N_HSOLCIT	Number of HSOLICIT	3	This defines the maximum number of HSOLICIT control message generation attempts as a non-HMA. After the N_HSOLICIT times of HSOLICIT control message issuance, EdgeMA tries to become the new HMA in the local multicast area. The default value for N_HSOLICIT is 3.
W_ HSOLICIT	HSOLICIT wait timer	T_HSOLICIT * N_HSOLICIT	This defines the maximum waiting time for HSOLICIT control message. If HSOLICIT control message does not arrive until W_HSOLICIT becomes zero, HMA realize that there is no member in the local network. Thus HMA stop multicasting received data to the local network. The default value for W_HSOLICIT is T_HSOLICIT * N_HSOLICIT.
T_HANNOUNCE	HANNOUNCE timer	1	This timer is used when HMA election is occurred by HMA leave. To prevent HANNOUNCE control message flooding during HMA election, every non-HMA has a different time value for sending HANNOUNCE message. The default value for T_HANNOUNCE is 1 second and can be changed arbitrarily.

9.4.5 **Parameters for maintenance of EdgeTree**

To maintain established connection for data relay, each CMA periodically sends a RELREQ control message to its PMA. The following parameters are used to support the maintenance of EdgeTree:

Table 16 – Parameter for maintenance of EdgeTree

Value	Name	Default value	Description
T_RELREQ	Relay request timer	6	This timer is used to generate a periodic RELREQ control message. PMA and CMA should both keep the same T_RELREQ. The initial value of T_RELREQ is 6 seconds.
N_RELREQ	Number of RELREQ	3	This is used to check whether the CMA is still alive. If the node does not receive the RELREQ control message from a specific CMA for the N_RELREQ times (until the RELREQ timer has expired), it considers that the CMA has left the session abruptly. The default value for N_ RELREQ is 3.

9.4.6 **Parameters for session leave**

RMCP-3 allows EdgeMA to leave the session before the session ends. For soft tree reconfiguration, the leaving EdgeMA should not leave the session abruptly but should wait for a certain period. The following parameters are used to support elegant session leave of EdgeMA:

Table 17 – Parameter for session leave

Value	Name	Default value	Description
T_LEAVE	Leave timer	10	This timer is used by each EdgeMA to wait for finding a new PMA of its CMA; leaving EdgeMA should relay session data to its CMA until this timer expires. The default value for T_LEAVE is 10 seconds and can be changed arbitrarily.

9.5 Data profile

RMCP-3 defines the data profile as a profile describing the requirements for forwarding data between a PMA and its direct CMA. The data profile is used to negotiate the data channel in terms of the type of data delivered during the session. When multiple types of data are simultaneously transmitted in a session, the information of each data stream is defined through negotiation. Following table shows possible parameters and their value that can be used in the data profile. The value that is used in data profile negotiation is in the text-mode, thus the value itself is the meaning and the value.

Table 18 – Parameters for data profile

Parameter	Value	Description
Protocol	TCP	Data channel will be established using TCP.
	UDP	Data channel will be established using UCP.
	SCTP	Data channel will be established using SCTP.
Listening address	*IPv4 address:port number*	Listening address and port number of MA.
Data stream type	REALTIME	Data stream type is real-time.
	RELIABLE	Data stream type is reliable. In this case, RMCP-3 encapsulates data using defined encapsulation format. See Appendix B.
Extension	Additional parameters can be defined. Examples are shown in Appendix A & B.	

Annex A

N-plex real-time data delivery scheme

(This annex does not form an integral part of this Recommendation | International Standard)

A.1 Overview

This annex defines the N-plex real-time data delivery scheme that can be used in the RMCP-3 protocol. The RCMP-3 protocol does not define the actual data delivery scheme due to being too implementation-dependent. The data delivery is dependent on the application characteristics. The data delivery scheme defined here is only a suggested example. More suitable data delivery scheme should be defined for the actual implementation of RMCP-3.

This annex uses the example model defined in the protocol operation clause of this Recommendation which is shown in Figure A-1.

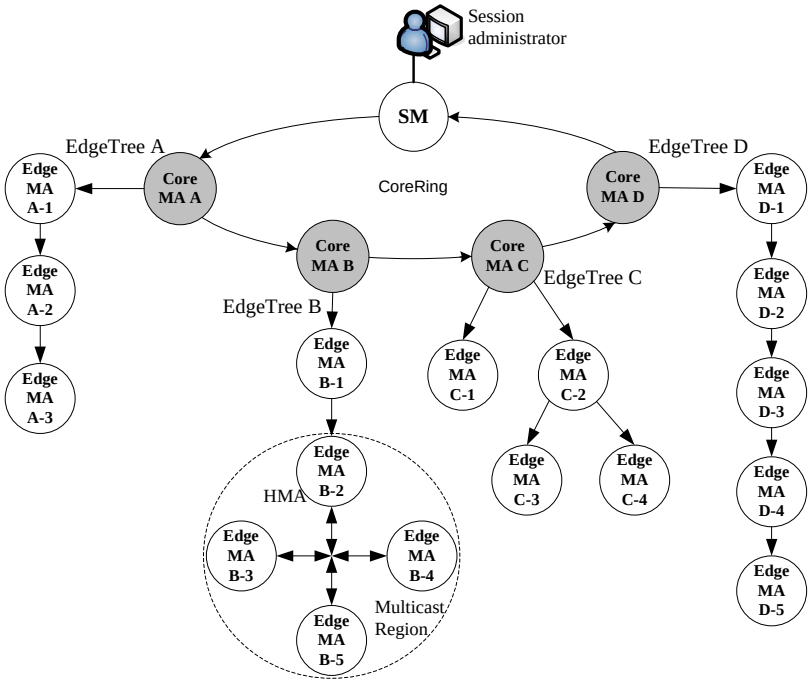

Figure A-1 – RMCP-3 topology

A.2 Example of real-time data delivery

In this clause, two examples of real-time data delivery are described. A first example describes data delivery from EdgeMA in the multicast region and a second example describes data delivery from EdgeMA in the unicast area.

Figure A-2 shows an example of real-time data delivery from HMA. In the example, EdgeMA B-2, HMA of multicast region, sends real-time data. The EdgeMA B-2 will multicast the data to its local network and sends the data to the CoreMA that is the root node of the EdgeTree B.

Upon receiving the data, CoreMA B checks whether the data is from its direct CMA. The direct CMA would be EdgeMA B-1. Since the EdgeMA B-1 is not the originator of the data, CoreMA B sends the data to EdgeMA B-1. The EdgeMA B-1 checks if the originator is its CMA, which is EdgeMA B-2 to deliver the data along the EdgeTree. Since, the EdgeMA B-2 is the originator, EdgeMA B-1 will not send the data to EdgeMA B-2.

The CoreMA B also delivers the data to CoreMA A and CoreMA C which are neighboring CoreMAs in the CoreRing. Each CoreMA delivers received data to its own EdgeTree and its neighbor CoreMA(s). It should be noted that the data path in the CoreMA can be bi-directional. The data path does not have to follow the control path.

It is, also, important not to deliver the data to the CoreMA which has already receives the data. In the example, CoreMA C should not deliver the received data to the CoreMA B. The CoreMA C should only deliver the data to CoreMA D in

the CoreRing. This example uses connectivity of the CoreRing to deliver RMCP-3 data. To provide loop-free delivery model, the SM can be used as an end-point to the ring topology. The SM does not need to receive multicast data. CoreMA will know that the neighboring node is a SM and does not deliver the received multicast data. In this example, CoreMA D and CoreMA A should not deliver the data received to SM. It is possible that a loop is prevented by using these methods.

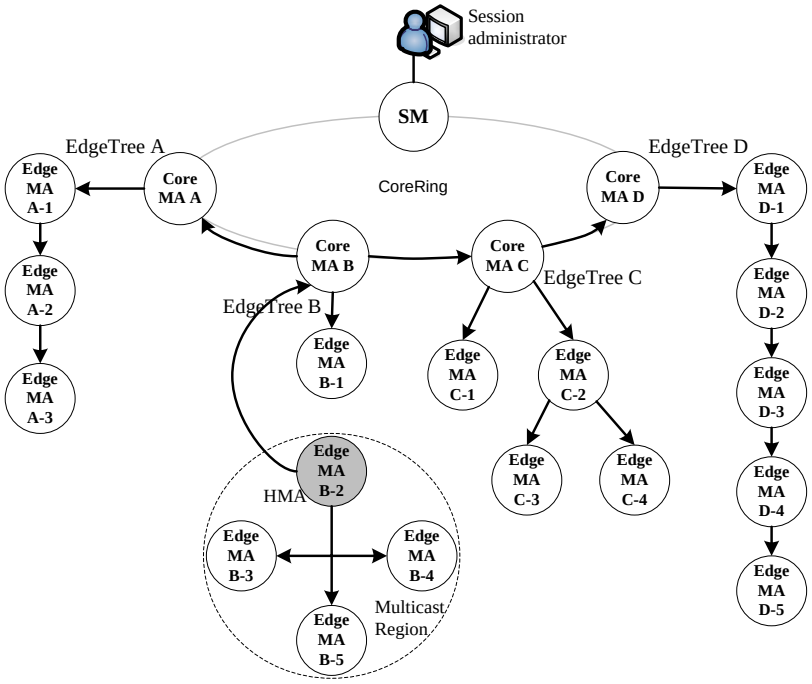

Figure A-2 – Real-time data delivery path (HMA sends the data)

Figure A-3 shows an example of real-time data delivery from EdgeMA in unicast area. In the example, EdgeMA C-2 sends real-time data. The EdgeMA C-2 sends the data to its CMAs and to the CoreMA C that is root node of EdgeTree C.

Upon receiving the data, CoreMA C checks whether its CMA is the originator of received data. Since EdgeMA C-2 is the originator of the data, CoreMA C does not send the data to EdgeMA C-2.

CoreMA C delivers the data to CoreMA B and D which are connected to CoreMA C on the CoreRing. Each CoreMA delivers received data to its own EdgeTree and neighboring CoreMA(s).

Both CoreMA D and CoreMA A deliver the data to its own EdgeTree but do not deliver the data to SM.

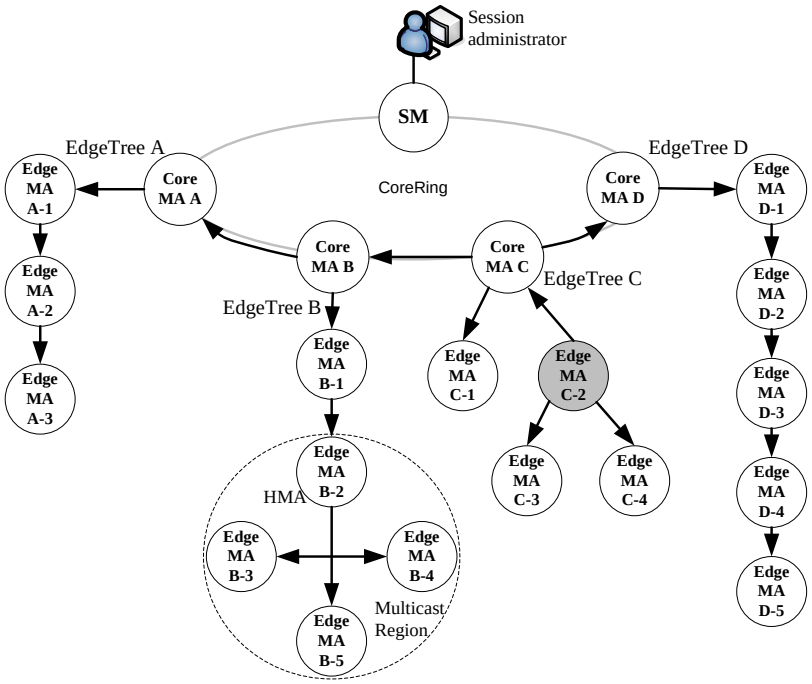

Figure A-3 – Real-time data delivery path (EdgeMA within unicast area sends the data)

For data relaying, both CoreMA and EdgeMA should maintain a forwarding table which may consist of *incoming_from* (*from* in figure A-4) field and *outgoing_to* (*to* in figure A-4) field. Figure A-4 shows examples of forwarding table maintained by each MA. The *incoming_from* field represents neighboring MA in which the MA has received data from and the *outgoing_to* field represents neighboring MA in which the MA has to transmit data to. The *incoming_from* field with *lo0* represents the MA being the originator. Note that the CoreMA does not have *lo0* in the *incoming_from* field, since it is not the participating MA. The table of EdgeMA C-1 in figure A-4 has "C-2 (!=src)" meaning EdgeMA C-1 would only send data to EdgeMA C-2 if the originator of the data is not EdgeMA C-2. The RMCP-3network can configure a simple forwarding rule to transmit multicast data as in figure A-4.

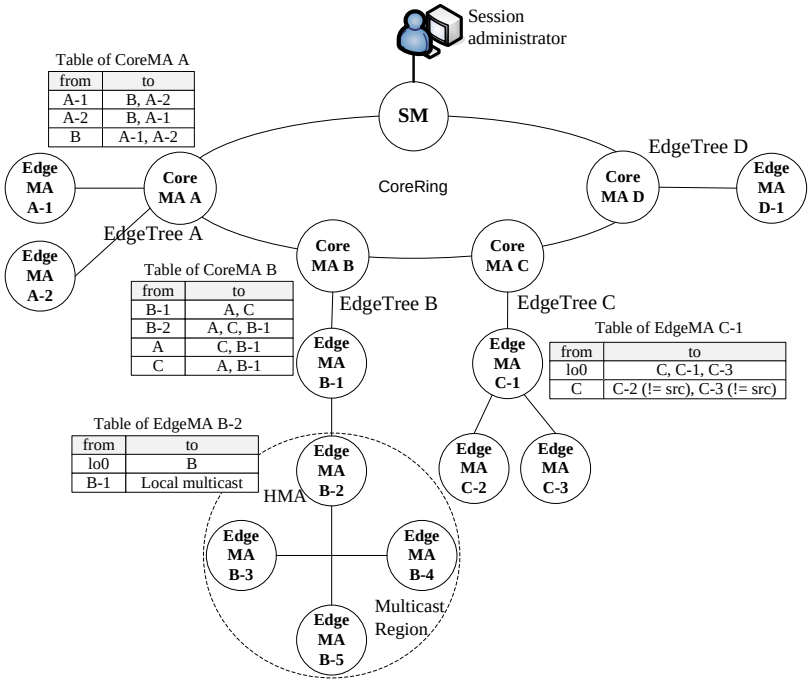

Figure A-4 – Examples of forwarding table

For relaying user data, the RMCP-3 may use IP-IP tunneling scheme. Using the IP-IP tunneling scheme, address of data originator can be preserved as the data is transmitted in the RMCP-3 network. The source EdgeMA, i.e. originating

EdgeMA, captures full IP multicast packet using a raw socket. Next, it forwards the IP multicast packet to MAs according to the forwarding table. The receiving MA also forward the received data through the raw socket in which the IP multicast application can receive the original packet. This tunneling scheme can be used for source-specific application. Figure A-5 shows an example of datagram flow when using IP-IP tunneling.

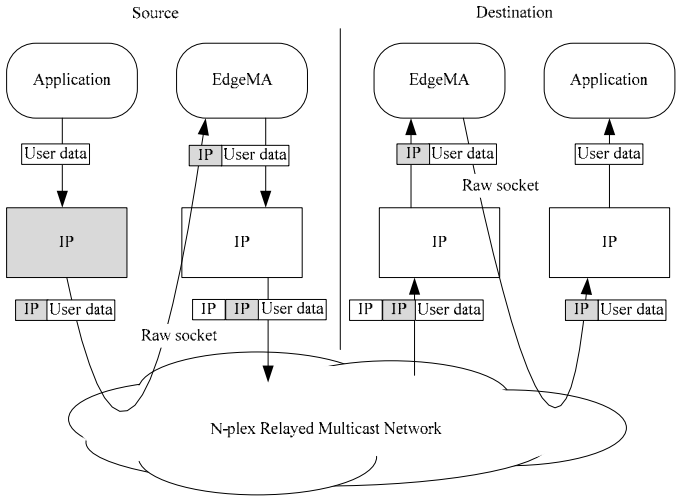

Figure A-5 – Example of datagram flow – using IP-IP scheme

Table A-1 defines parameters that are used in the data profile to negotiate real-time data stream using IP-IP encapsulation. The protocol that can be used is UDP or SCTP. The value for the encapsulation parameter is IP-IP indicating the use of IP-IP tunneling.

Table A-1 – Parameters for data profile – IP-IP scheme

Parameter	Value	Description
Protocol	UDP	Data channel will be established using UCP.
	SCTP	Data channel will be established using SCTP.
Listening address	*IPv4 address:port number*	Listening address and port number of MA.
Data stream type	REALTIME	Data stream type is real-time.
Encapsulation	IP-IP	Use of IP-IP encapsulation scheme

If an application does not have to receive data from a specific source, IP-IP tunneling may be not used. In this case, source address of packet is changed while packet is relayed through the RMCP-3 network. The source EdgeMA, i.e. originating EdgeMA, would receives user data of IP multicast. The source EdgeMA insert RMCP-3 header indicating the originating EdgeMA. The RMCP-3 header would prevent the originating EdgeMA to receive duplicate data from its PMA. The RMCP-3 header can consist of only source EdgeMA ID along with other information that may be needed to transfer to other MAs in the RMCP-3 network. The receiving EdgeMA forwards the received user data with RMCP-3 header according to the forwarding table. The EdgeMA also needs to send user data to application with RMCP-3 header discarded. Figure A-6 shows an example of datagram flow when IP-IP tunneling is not used.

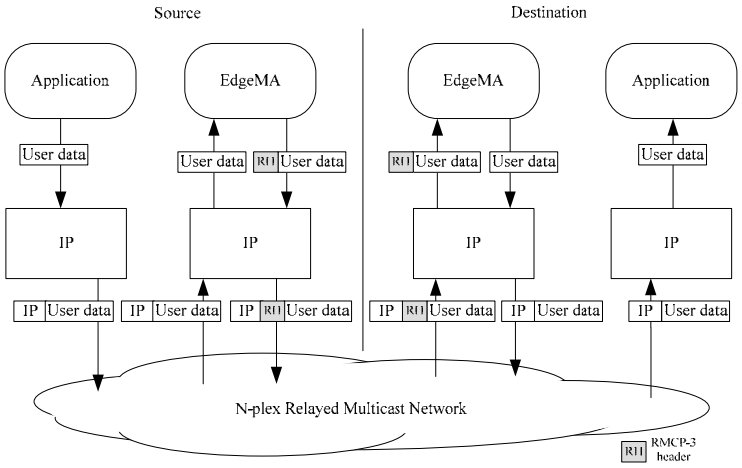

Figure A-6 – Example of datagram flow – not using IP-IP scheme

Table A-2 defines parameters that are used in the data profile to negotiate real-time data stream using the proposed p method of non-IPIP scheme. The proposed non-IPIP scheme defines RMCP-3 header with containing source address of the source EDGEMA.

Table A-2 – Parameters for data profile – IP-IP scheme

Parameter	Value	Description
Protocol	UDP	Data channel will be established using UCP.
	SCTP	Data channel will be established using SCTP.
Listening address	*IPv4 address:port number*	Listening address and port number of MA.
Data stream type	REALTIME	Data stream type is real-time.
RMCP-3 Header	SOURCE_EDGEMA	Use of RMCP-3 header to indicate Source EdgeMA

Annex B

N-plex reliable data delivery scheme

(This annex does not form an integral part of this Recommendation | International Standard)

B.1 Reliable data delivery using data profile

B.1.1 Overview

This scheme uses bi-directional TCP channel for hop-by-hop data delivery. It would be desirable to use TCP to deliver reliable data than using other different application-level data transferring because it guarantees the ordered sequence, reliable transmission, and built-in congestion control. Moreover, TCP is universally implemented and does not raise any questions of fairness.

However, just relaying data stream with TCP channel may result in following problems:

a) Any local network congestion can make the overall network performance drop instantly. That results from the fact that intermediate nodes, MAs are blocked from receiving upstream while streaming down user data with TCP channel. So, the local congestion affects intermediate nodes successively to SMA(s) in short time.

b) Parent switching may result in message loss.

c) Parent switching may result in duplicated data flow.

To resolving these problems, this scheme provides following functions:

a) Data buffering

b) Parent switching to support successive data transferring

c) Detecting and handling duplicated data

These functions are explained in the following section.

B.1.2 Data Buffering

MAs have local buffers to flow user data through a series of data channel. Local network congestion can be covered by the nearest MA as far as its buffer permits. So, this scheme really works well if the buffer is sufficiently large. However, long-lasting network congestion can make a local buffer reach nearly full. Before that case happens, the intermediate node should lead the bottle neck node to move on as shown as Figure B-1.

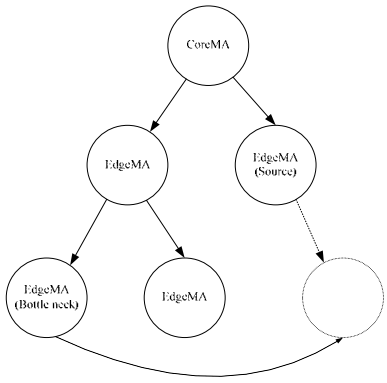

Figure B-1 – Example of switching action before buffer reaches full

B.1.3 Parent switching to support successive data transferring

Suppose that link throughput between EdgeMA 5 and EdgeMA 9 in Figure B-2 has dropped down. The throughput between EdgeMA 5 and EdgeMA 9 does not affect any other link throughput due to data buffering of application level in EdgeMA 5. If the buffer nearly reaches full, EdgeMA 5 notifies bottlenecked EdgeMA 9 to switch its parent with

another node using LEAVREQ message. The bottle neck EdgeMA 9 tries to switch its parent to other EdgeMA which is determined by network condition and node information from candidate nodes. The parent switch will be successful when candidate parent node can allows EdgeMA 9 to be its child. During parent switch procedure, EdgeMA 9 can request which it have to receive for successive data transferring. This method will be support a successive data transferring. Figure B-2 shows an example of parent switching that EdgeMA 9 tries to change its parent from EdgeMA 5 to EdgeMA 2.

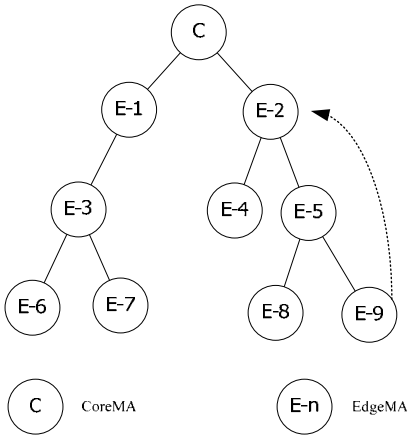

Figure B-2 – Example of parent switching

B.1.4 Detecting and handling duplicate data

B.1.4.1 Data sequence

Data sequence is also used to identify each data flow with Sender MAID. MAs should deliver user data successively after parent switching. To make it, data stream is relayed in service data unit (SDU) and the sequence number is allocated, sequentially, by each sending MA. The Sender MAID and the Data sequence are important information to detect duplicated data and provide criteria to choose new peer for parent switching.

B.1.4.2 Discarding duplicated data

Dynamic intermediate nodes organize and change network configuration by themselves. Then, data path can be piled on another and be looped for a while as illustrated in Figure B-3. This results in network congestion. So, it is desirable each MA which detects data duplication should discard the duplicated data. As described above, the Sender MAID and the Data sequence provide identification of data flow.

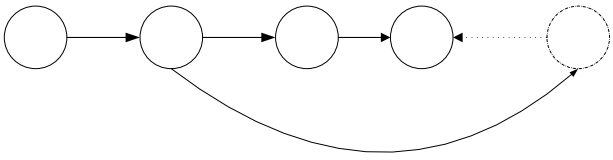

Figure B-3 – Duplicated data with parent switching

B.1.5 Protocol Operation

Protocol operation consists of three phases: connection establishment, data transfer and connection termination.

B.1.5.1 Connection establishment

Procedure for connection establishment is illustrated in Figure B-4.

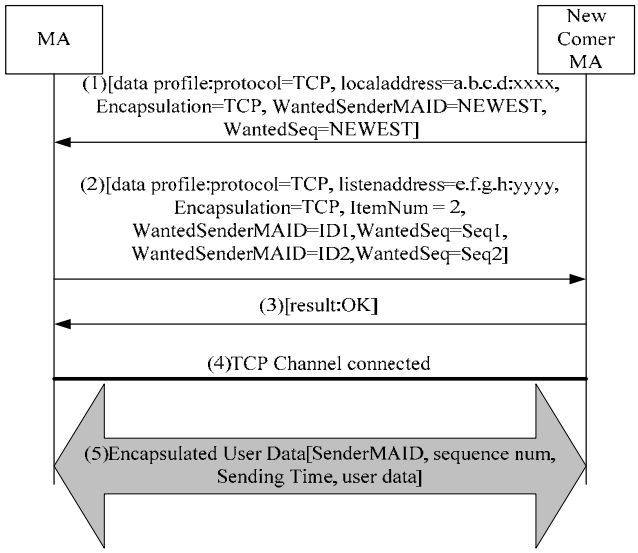

Figure B-4 – Procedure for Connection establishment

a) New comer MA sends MA a data profile containing its own local address, wanted sender MAID and wanted sequence number from which it expects to receive. If the sender MAID and the sequence number are not fixed, they can be set NEWEST which requires future sequence number from any sender.

If the MA which received data profile allows data relaying, it responses with another data profile containing listening address and a list of wanted sender MAID and wanted sequence number from which it expects to receive.

a) If the new comer MA which received the data profile allows data relaying, it responses with ok.

b) Two MAs which have exchanged channel information establish bi-directional TCP connections between them.

c) The two MAs relay user data from other channels through newly connected data channel. User data are encapsulated with the sender MAID, sequence number, sending time. They relay user data according to the negotiated data profiles. For example, the MA relays new coming data from any sender to the new comer MA. On the other hand, the new comer MA sends ID1's message from sequence Seq1.

B.1.5.2 Data Profile

Table B-1 defines parameters that are used in the data profile to negotiate reliable data stream using the proposed method. The proposed reliable scheme defines ItemNum, WantedSenderMAID, and WantedSeq parameters to be negotiated.

Table B-1 – Parameter of data profile for reliable data transmission

Parameter	Value	Description
protocol	TCP	Data channel will be established using TCP.
localaddress	*IPv4 address:port number*	Local address and port number of requesting MA
listenaddress	*IPv4 address:port number*	Listening address and port number of MA.
Encapsulation	TCP	Data stream type is reliable. RMCP-3 encapsulates data using defined encapsulation format.
ItemNum	*Number of relaying data which are received from different senders.*	MA can specify the number of streams which are received from different senders. This value means that MA has to receive streams as many as value of ItemNum.
WantedSenderMAID	*Sender MAID*	To receive from a specific sender, MA can specify MAID of sender for each stream.

WantedSeq	*Sequence number*	By specifying the sequence number of data expected to receive, MA can request specific data for each stream.

B.1.5.3 Data Transfer

Every node in session can invoke user data stream. According to the sending node, the direction of data delivery path changes as shown in Figure B-5 and Figure B-6.

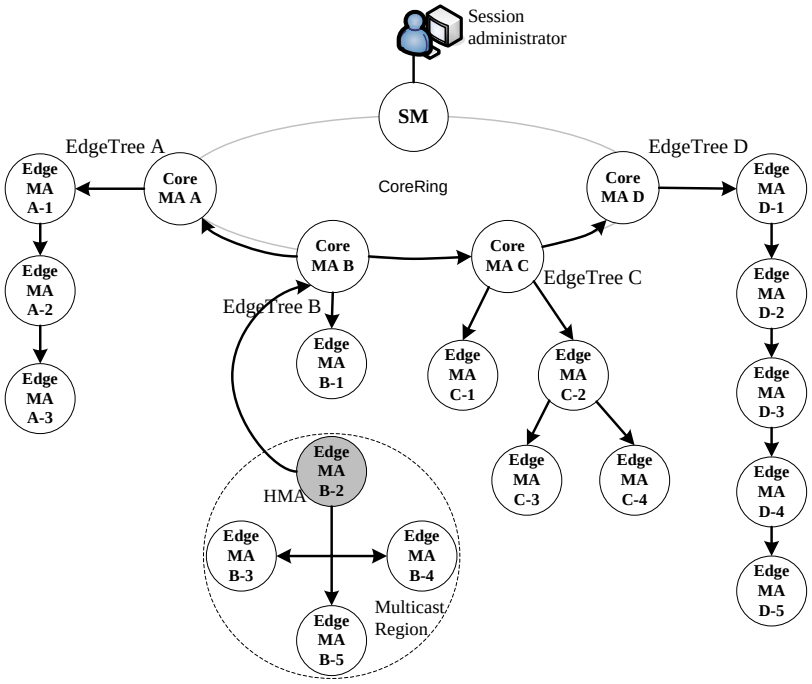

Figure B-5 – Example for data delivery (HMA sends the data)

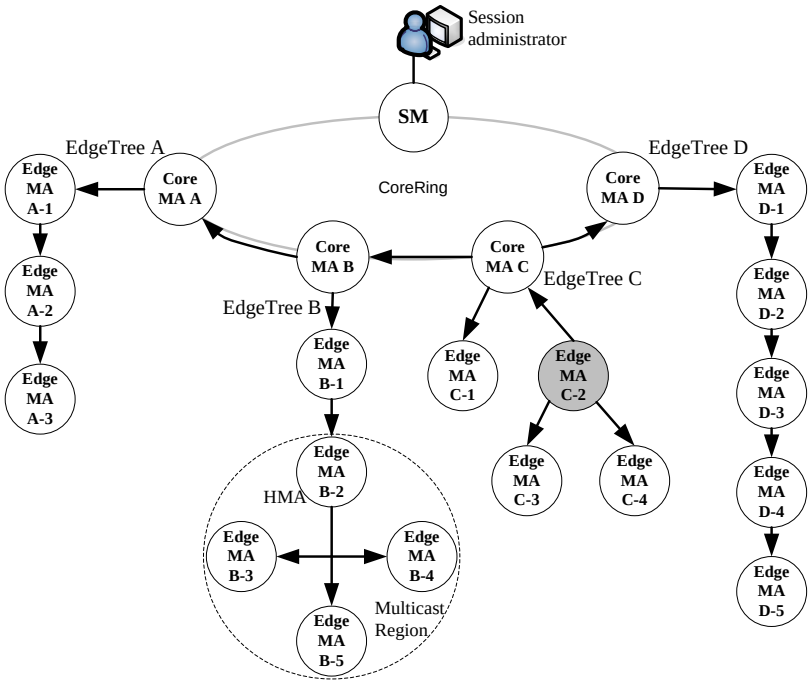

Figure B-6 – Example for data delivery (EdgeMA within unicast area sends the data)

However, each intermediate node needs not know the whole data delivery path. Instead, receiving stream from a peer node, it relays stream to the other peer(s). This method may result in data duplication, consequently, network congestion in case of parent switching. So, MA which detects data duplication should discard the duplicated data.

B.1.5.4 Parent switching

Figure B-7 shows MA's parent switching procedure.

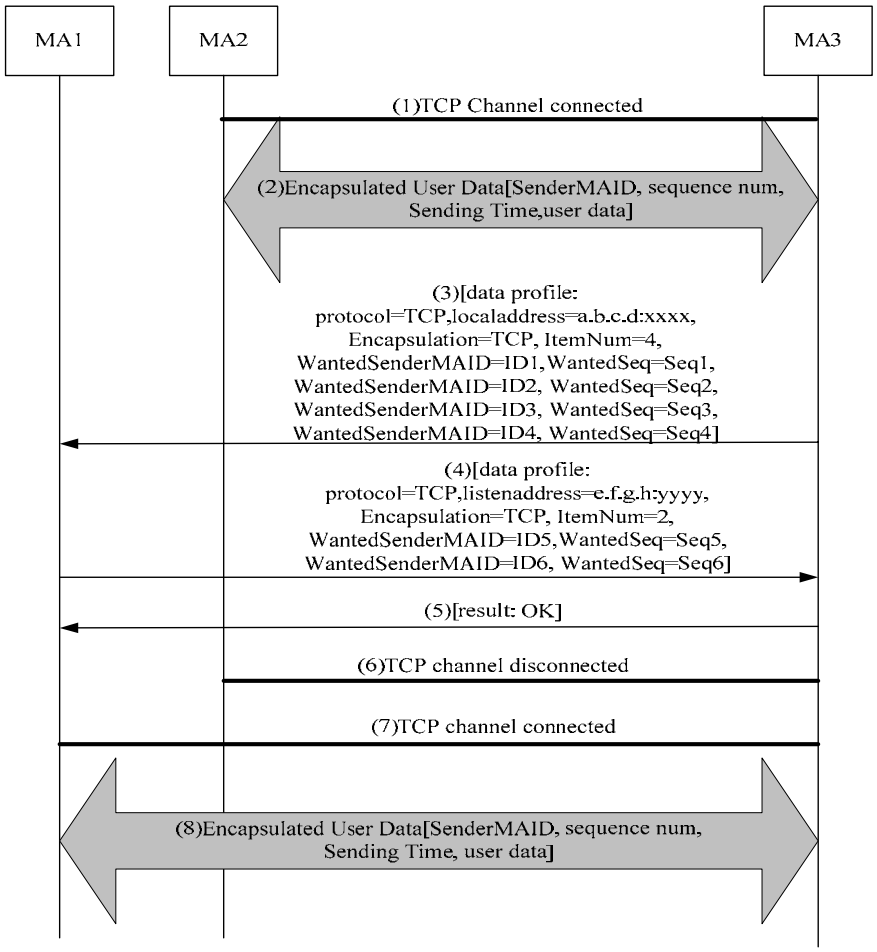

Figure B-7 – Procedure for parent switching

a) TCP connections between MA2 and MA3 are established

b) MA2 and MA3 exchange encapsulated data with the sender MAID, sequence number and sending time.

c) MA3 sends a new Peer MA, MA1, a data profile containing local address of itself, a list of wanted sender MAID, wanted sequence number from which it expects.

d) If the MA1 which received data profile allows data relaying, it responses with another data profile containing listening address, a list of wanted sender MAID, wanted sequence number from which it expects to receive observing the ITEM POLICY

e) If the MA3 which received the data profile allows data relaying, it responses with ok.

f) TCP channel between MA2 and MA3 is disconnected.

g) MA1 and MA3 establish bi-directional TCP connections between them.

h) MA1 and MA3 relay user data from other channels through newly connected data channel according to the negotiated data profiles. For example, MA3 sends ID5's message from sequence Seq5 to MA1.

B.1.5.5 Connection termination

Procedure for connection termination is illustrated in Figure B-8.

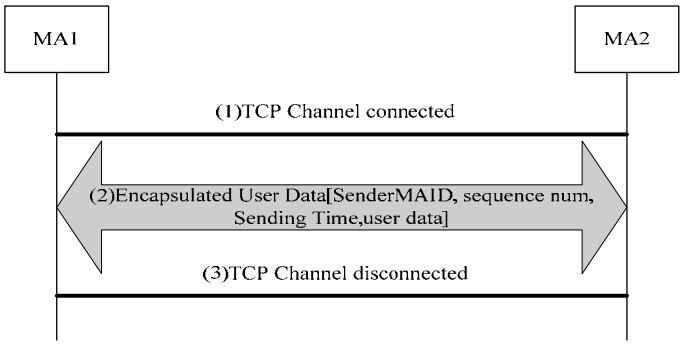

Figure B-8 – Procedure for connection termination

a) TCP connections between two MAs are established

b) MAs send encapsulated user data with a Sender MAID, a sequence number and sending time.

c) Either MA can call TCP close to terminate TCP channel.

B.1.6 Data Encapsulation Format

User data message consists of the following fields, in Figure B-9.

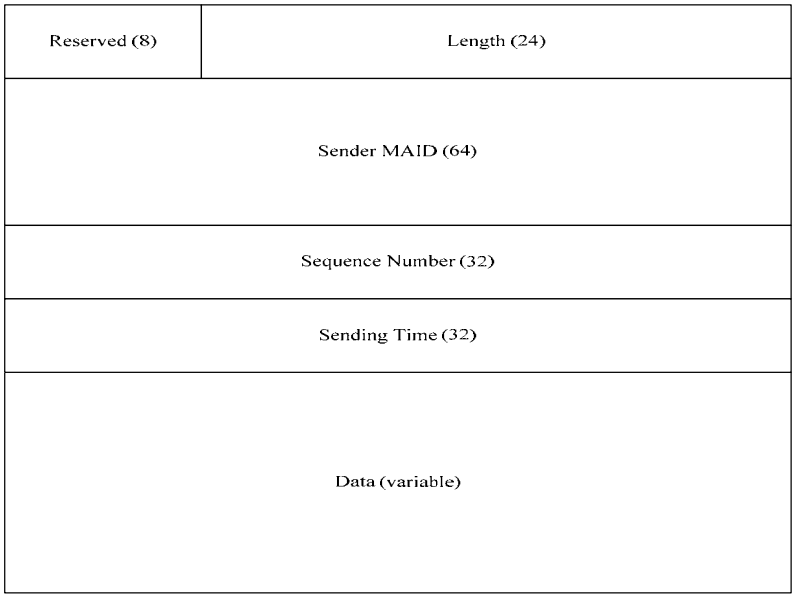

Figure B-9 – Data Encapsulation Format

a) *Reserved* – Reserved for the further use (should be set to zero now);

b) *Length* – denotes the total byte length of current message;

c) *Sender MAID* – denotes the MAID of sending MA;

d) *Sequence Number* – denotes the sequence number of current Service Data Unit (SDU). Sequence number can be allocated globally round-robin by each SMA;

e) *Sending Time* – denotes the time when sending MA sends data. It is based on a specific system's clock

f) *Data* – User Data

B.2 Reliable data delivery for source-specific application

In this section, an example of reliable data delivery that can be used for source-specific application is introduced in this clause.

B.2.1 Overview

There are some applications which want to receive data from specific source. Since source address of data is changed while data is relayed, a mechanism to preserve source address is needed for source-specific application. This section defines another method in which to deliver reliable data and to preserve original source address of the original source application.

B.2.2 IP-IP tunneling scheme and data encapsulation

To preserve the source address, IP-IP tunneling scheme which is already introduced in Annex A and data encapsulation are used. Figure B-10 shows an example data flow that provide IP-IP tunneling scheme. The RMCP-3 header is different from the RMCP-3 header in Annex A. The details of the RMCP-3 header will be described in clause B.2.5.

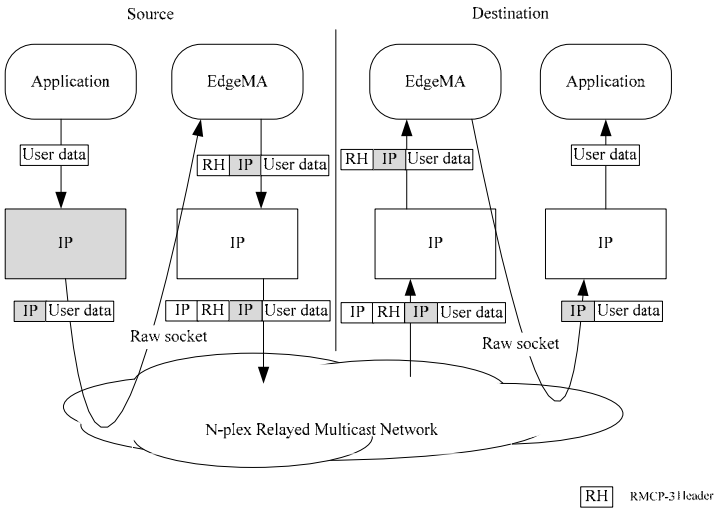

Figure B-10 – Example of data flow (reliable delivery preserving source address)

B.2.3 Data buffering and retransmission

The defined method uses data buffering, acknowledgement, and retransmission for reliable data delivery. The multicast data are partially buffered in the CoreMA. The EdgeMA periodically send acknowledgement to the CoreMA, it the CoreMA receives acknowledgement from every EdgeMA, it can remove the stored data in its buffer. If the EdgeMA finds out that it has lost some data, it can request for missing data to the CoreMA. If the CoreMA does not have the data requested from EdgeMA, then EdgeMA may request the missed data of original data sender.

B.2.4 Examples of data flow

In this section, three examples of data flow are shown. All examples in B.2 clause follow the topology shown in Figure B-11.

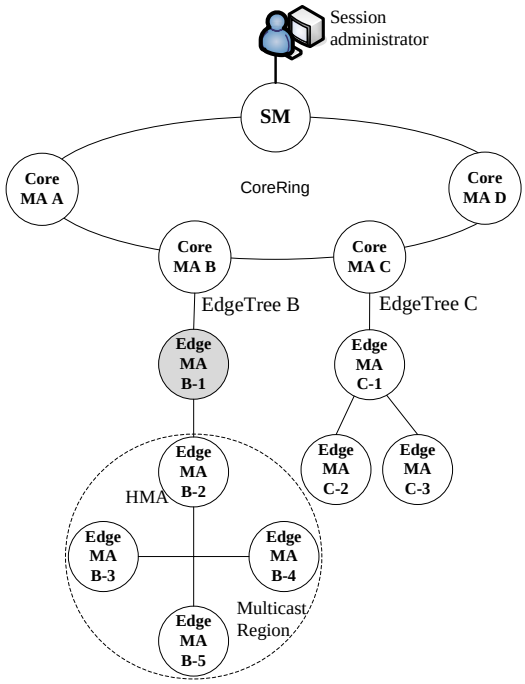

Figure B-11 – Example topology of RMCP-3

B.2.4.1 Reliable data flow at the start of data transmission

Each CoreMA stores received data and forwards the data to its direct CMA(s) and neighbor CoreMA(s). The data is delivered along the RMCP-3 Hybrid Tree. To prevent implosion of acknowledgement from EdgeMAs, each EdgeMA should not send acknowledgement for each received data. Instead, EdgeMA should send acknowledgement when the number of received data reaches the threshold value. The threshold values should be defined by SM . In this example, the threshold value is set to 10 which mean that EdgeMA sends acknowledgement message to CoreMA for every 10 data.

Figure B-12 shows data flow when EdgeMA B-1 starts transmitting multicast data. The EdgeMA B-1 sends data to CoreMA B and the CoreMA B will eventually forward received data to CoreMA C. The CoreMA B part is omitted in this example. The CoreMA C would buffer the data from EdgeMA B-1 and also delivered the data to its EdgeTree. The EdgeMA C-1, C-2, C-3 would receive data from CoreMA C. The EdgeMAs would send acknowledgement message to the CoreMA C after receiving threshold number of data (which is 10 in this example). The CoreMA would know that the EdgeMAs in its EdgeTree have successfully received first 10 data. The CoreMA can delete first 10 data, but it does not delete the data immediately since it may be needed for retransmission. CoreMA C deletes first 10 data after receiving acknowledgement for 20. Acknowledgement method may be used or timing method may be used in deleting data in the CoreMA buffer. In the timing method, CoreMA can empty the buffer after certain period of time.

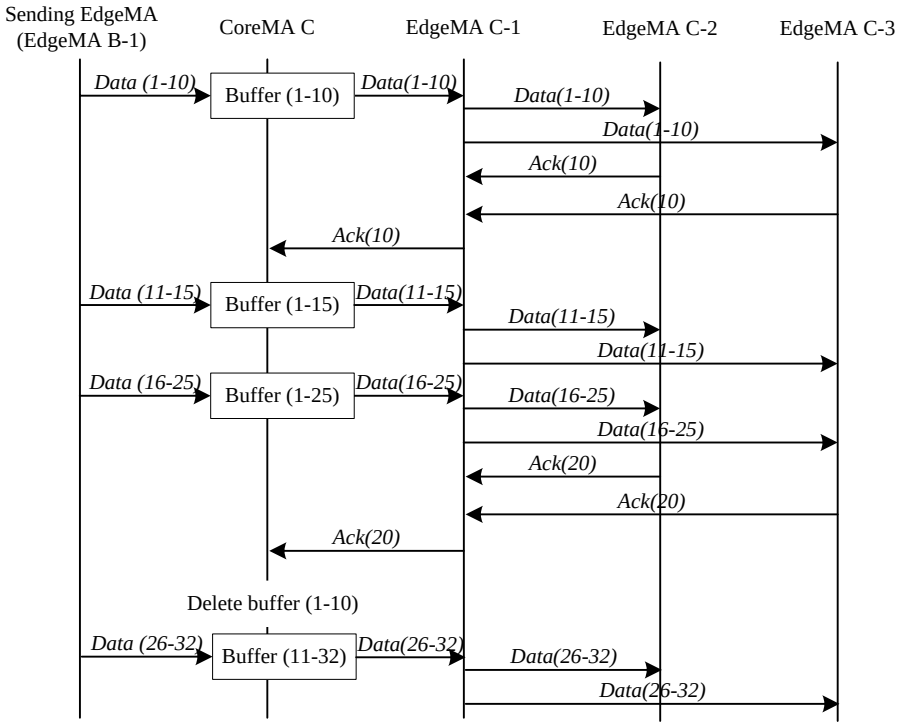

Figure B-12 – Data flow at the start of data transmission

B.2.4.2 Parent switching within same EdgeTree

This clause defines data flow in case in which the EdgeMA C-3 performs parent switching to CoreMA C. During the parent switching procedure, the EdgeMA C-3 did not receive data from 11 to 15. Upon receiving the next data which is data 16, EdgeMA C-3 will know that it has lost data 11 to 15. The EdgeMA will continue to receive data after 16, since it must continue to participate in multicast transmission. But, the EdgeMA C-3 does not send any acknowledgement message before receiving the lost data 11 to 15. If the EdgeMA C-3 receives 11 to 15 along with data 16 to 25, then it would send acknowledgement message to the CoreMA. Note, that for simplicity, the EdgeMA C-3 does not send acknowledgement message of data 25, since the acknowledgement message is defined to be transmitted to every 10 data.

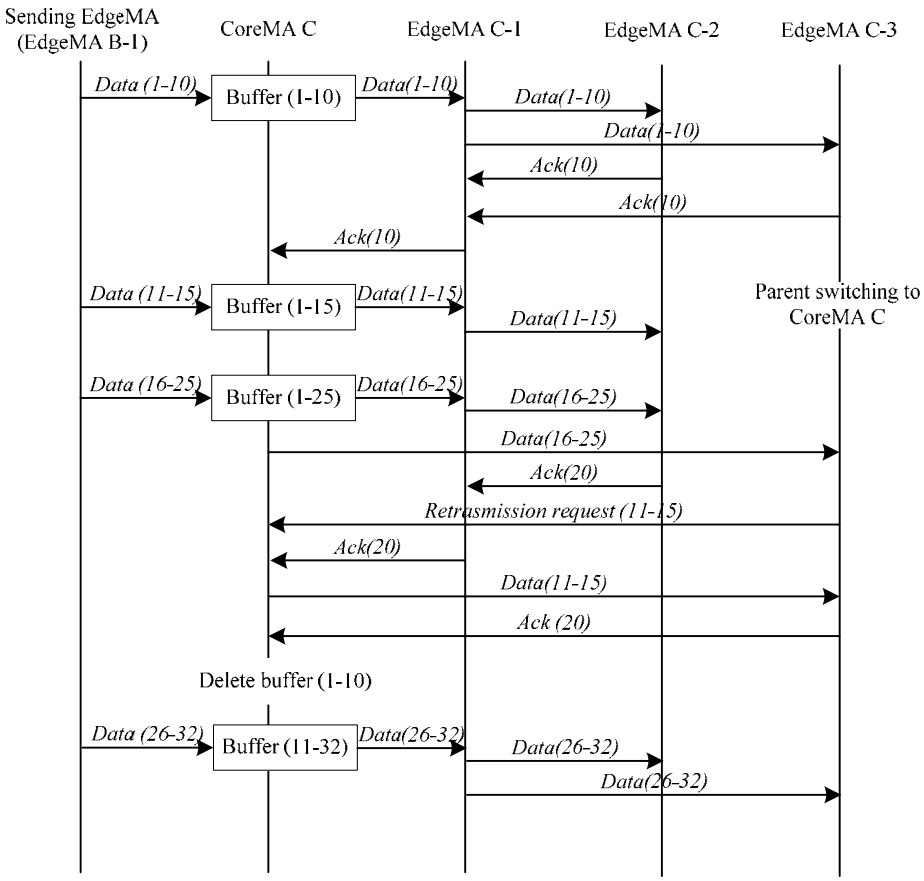

Figure B-13 – Data flow after parent switching within same EdgeTree

B.2.4.3 Parent switching between EdgeTrees

In this example, EdgeMA C-3 performs parent switching to CoreMA B which is root of EdgeTree B. Although, this case is not usual, it may happen. Same as above, EdgeMA C-3 requests to retransmit missed data of its CoreMA. After retransmission is complete, EdgeMA C-3 sends acknowledgement for 30.

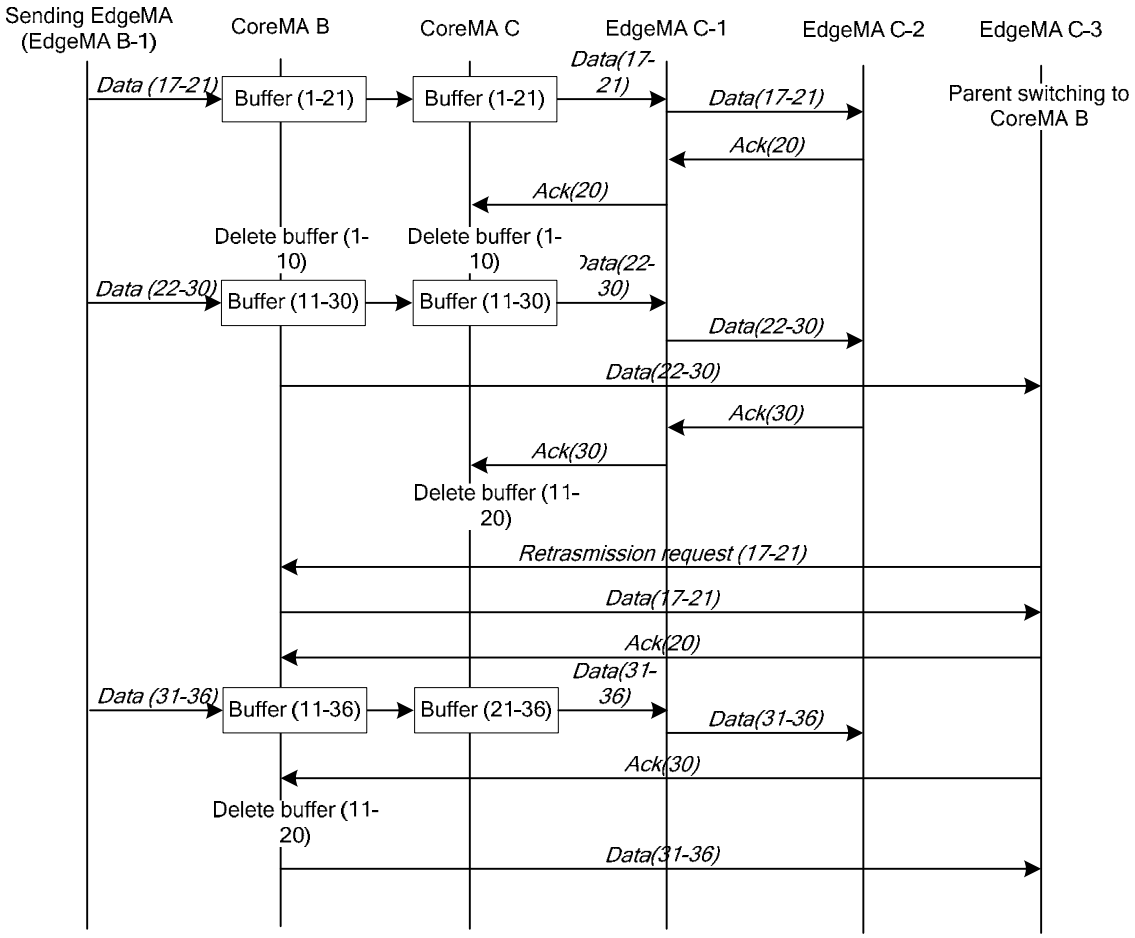

Figure B-14 – Data flow after parent switching between EdgeTree

If a new EdgeMA joins the RMCP-3 multicast in the middle of a session and wishes to receive data that was delivered before joining the session, then it can ask for retransmission from its CoreMA. Since the CoreMA does not keep full data, it would transmit every data in its buffer. The new EdgeMA may ask the sending EdgeMA for retransmission for full data. Although, it would be application-dependent, but if the sending EdgeMA can transmit every data from the beginning, then it may send every data that was transmitted in the session.

B.2.5 Data encapsulation and message format

Data encapsulation format may be same as Figure B-15. Each field has the following meaning and value:

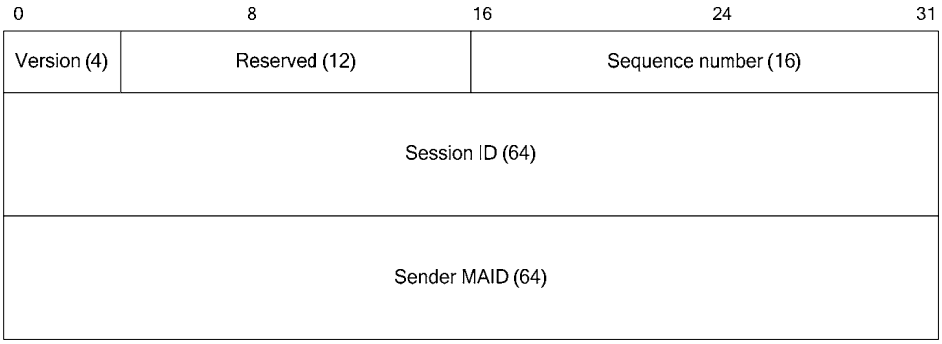

Figure B-15 – Data encapsulation format

a) *Version* – denotes the version of RMCP. Its value shall be set to 0x03;

b) *Reserved* – Reserved for further use;

c) *Sequence number* – denotes the sequence number of current Service Data Unit (SDU). Sequence number can be allocated globally round-robin by each SMA;

d) *Session ID* – denotes a 64-bit integer value that identifies a session;

e) *Sender MAID* – denotes a 64-bit unique value used to identify sender MA for a certain session.

Acknowledgement may be a control message of RMCP-3. Figure B-16 shows an example of acknowledgement control message format.

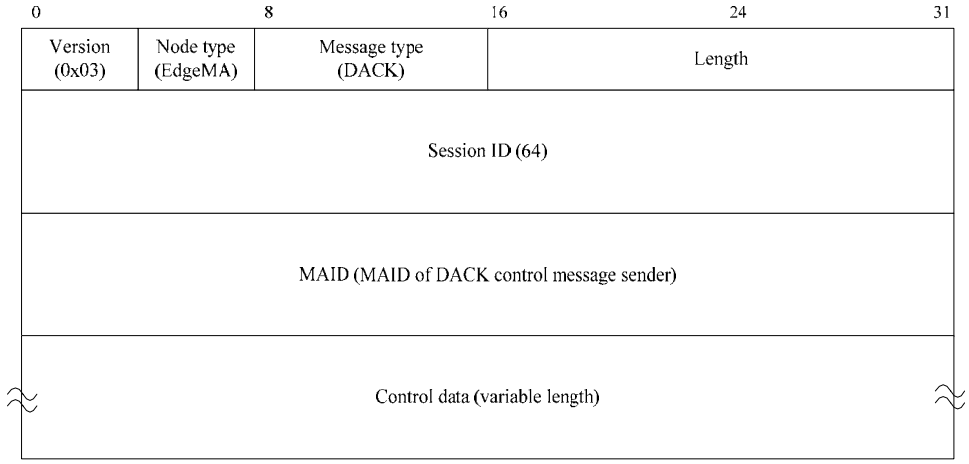

Figure B-16 – Example of DACK control message format

To indicate which data is received, acknowledgement message should express the sequence number of acknowledging data. Figure B-17 shows an example of control data and sub-control data format of DACK control message.

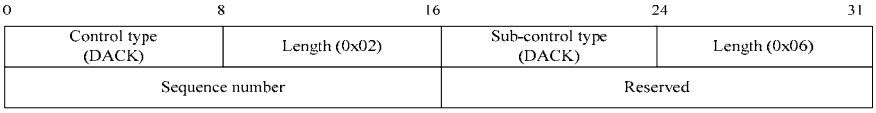

Figure B-17 – Example of DACK control and sub-control data format

Each field has the following meaning and value:

a) *Control type* – denotes the type of control data. Its value shall be set to the code for DACK;

b) *Length* – denotes the length of control data. Its value shall be set to 0x02 which means 2-byte;

c) *Sub-control type* – denotes the type of sub-control data. Its value shall be set to the code for DACK;

d) *Length* – denotes the length of sub-control data. Its value shall be set to 0x06 which means 6-byte;

e) *Sequence number* – shall be set to the sequence number of acknowledging data;

f) *Reserved* – Reserved for further use.

To request for retransmission and to specify which data it needs, retransmission request message may be used. Figure B-18 shows an example of RETREQ control message format.

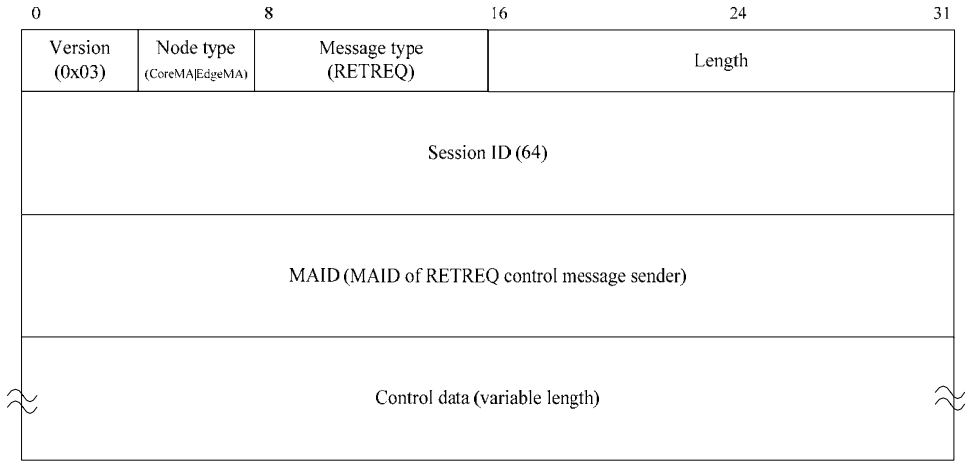

Figure B-18 – Example of RETREQ control message format

Figure B-19 shows an example of control data and sub-control data format of RETREQ control message. Each field has the following meaning and value:

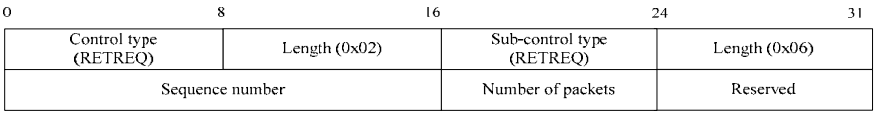

Figure B-19 – Example of RETREQ control data and sub-control data format

a) *Control type* – denotes the type of control data. The value shall be set to the code for RETREQ;

b) *Length* – denotes the length of control data. The value shall be set to 0x02 which means 2-byte;

c) *Sub-control type* – denotes the type of sub-control data. The value shall be set to the code for RETREQ;

d) *Length* – denotes the length of sub-control data. The value shall be set to 0x06 which means 6-byte;

e) *Sequence number* – denotes the sequence number of missed data. If number of missed data is more than two, value of this field means first sequence number of missed data;

f) *Number of packet* – denotes the number of data required to retransmit;

g) *Reserved* – Reserved for further use.

B.2.6 Data profile

Table B-2 defines parameters that are used in the data profile to negotiate real-time data stream using the proposed method. The proposed method uses IP-IP encapsulation scheme and newly defined RMCP-3 header. Since, there is no appropriate name for the proposed scheme, thus the value for the RMCP-3 header is defined as APPENDIX_B.2.

Table B-2 – Parameters for data profile

Parameter	Value	Description
Protocol	TCP	Data channel will be established using TCP.
	SCTP	Data channel will be established using SCTP.
Listening address	*IPv4 address:port number*	Listening address and port number of MA.
Data stream type	RELIABLE	Data stream type is reliable.
Encapsulation	IP-IP	Use of IP-IP Encapsulation
RMCP-3 Header	APPENDIX_B.2	Use of RMCP-3 header defined in appendix B.2