International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.603.1

(03/2010)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

OSI networking and system aspects – Networking

Information technology – Relayed multicast protocol: Specification for simplex group applications

Recommendation ITU-T X.603.1

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	**X.600–X.629**
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems management framework and architecture	X.700–X.709
Management communication service and protocol	X.710–X.719
Structure of management information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, concurrency and recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	X.1000–X.1099
SECURE APPLICATIONS AND SERVICES	X.1100–X.1199
CYBERSPACE SECURITY	X.1200–X.1299
SECURE APPLICATIONS AND SERVICES	X.1300–X.1399
CYBERSECURITY INFORMATION EXCHANGE	X.1500–X.1598

For further details, please refer to the list of ITU-T Recommendations.

INTERNATIONAL STANDARD ISO/IEC 16512-2
RECOMMENDATION ITU-T X.603.1

Information technology – Relayed multicast protocol: Specification for simplex group applications

Summary

This Recommendation | International Standard describes an application-layer protocol which constructs multicast tree for data delivery from a sender to multiple receivers over Internet where IP multicast is not fully deployed. The specified relayed multicast protocol consists of multicast agent and session manager. This Recommendation | International Standard specifies a series of functions and procedures of multicast agent to construct one-to-many relayed data path and to relay simplex data. It also specifies the operations of session manager to manage multicast sessions. This protocol can be used for applications that require one-to-many data delivery services, such as multimedia streaming service, file dissemination service, etc.

Amendment 1 describes the security functionalities of an application-level relayed multicast protocol for one-to-many group applications. The protocol provides various security facilities to fulfil general as well as specific security requirements. Some detailed functions that can operate with a variety of standardized security mechanisms are provided. This amendment enforces the existing RMCP protocol security.

Amendment 2 revises messages and code values.

History

Edition	Recommendation	Approval	Study Group	
1.0	ITU-T X.603.1	2007-02-13	17	
1.1	ITU-T X.603.1 (2007) Amend.1	2009-11-13	11	
1.2	ITU-T X.603.1 (2007) Amend. 2	2010-03-01	11	
2.0	ITU-T X.603.1	2010-03-01	11	Integrated publication of above editions

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Introduction

Relayed MultiCast Protocol Part 2 (RMCP-2) is an application-layer relayed multicast protocol for simplex group applications. RMCP-2 can construct an optimized and robust one-to-many relayed multicast delivery path over a unicast network with the help of RMCP entities defined by Rec. ITU-T X.603 | ISO/IEC 16512-1.

An RMCP-2 session consists of one SM and one or more MAs; SM initiates and terminates RMCP-2 session and manages RMCP-2 session and participated MAs; MA configures an RMCP-2 tree to deliver group data by exchanging a series of RMCP-2 control messages.

Along the relayed multicast delivery path, several types of data delivery channels can be constructed according to the requirement of application services.

INTERNATIONAL STANDARD
RECOMMENDATION ITU-T

Information technology – Relayed multicast protocol: Specification for simplex group applications

1 Scope

This Recommendation | International Standard specifies the Relayed MultiCast Protocol for simplex group applications (RMCP-2), an application-layer protocol, which constructs a multicast tree for data delivery from one sender to multiple receivers over the Internet where IP multicast is not fully deployed.

Clauses 5-8 define a basic RMCP-2 protocol without security features, and clauses 9-12 define a secure RMCP-2 protocol that adds security features to the basic protocol. Both protocols specify a series of functions and procedures for multicast agents to construct a one-to-many relayed data path and to relay simplex data. They also specify the operations of the session manager to manage multicast sessions.

These protocols can be used for applications that require one-to-many data delivery services, such as multimedia streaming services or file dissemination services.

Annex E defines a membership authentication procedure for use with the secure RMCP-2 protocol. Annexes A-D provide informative material related to these protocols. Annex F contains an informative bibliography.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this 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, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

– Recommendation ITU-T X.603 (2004) | ISO/IEC 16512-1:2005, *Information technology – Relayed multicast protocol: Framework.*

2.2 Additional references

– ISO/IEC 9797-2:2002, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function*.

– ISO/IEC 9798-3:1998, *Information technology – Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques*.

– ISO/IEC 18033-2:2006, *Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers*.

– ISO/IEC 18033-3:2010, *Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers*.

– ISO/IEC 18033-4:2005, *Information technology – Security techniques – Encryption algorithms – Part 4: Stream ciphers*.

– IETF RFC 2094 (1997), *Group Key Management Protocol (GKMP) Architecture*.

– IETF RFC 3546 (2003), *Transport Layer Security (TLS) Extensions*.

– IETF RFC 3830 (2004), *MIKEY: Multimedia Internet KEYing*.

– IETF RFC 4279 (2005), *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS).*

– IETF RFC 4346 (2006), *The Transport Layer Security (TLS) Protocol Version 1.1*.

– IETF RFC 4535 (2006), *GSAKMP: Group Secure Association Key Management Protocol*.

3 Definitions

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

3.1 multicast: A data delivery scheme where the same data unit is transmitted from a single source to multiple destinations over a single invocation of service.

3.2 IP multicast: A multicast scheme in an IP network supported by multiple multicast-enabled IP routers.

3.3 relayed multicast: A multicast data delivery scheme that can be used in unicast environments; the scheme is based on intermediate multicast agents that relay multicast data from a media server to media players over a tree hierarchy.

3.4 relayed multicast protocol (RMCP): A protocol that supports and manages the relayed multicast data transport.

3.5 RMCP-2 session: An MA set that uses the RMCP to configure the data delivery path.

3.6 multicast agent (MA): An intermediate data transport entity used to relay the multicast application data. Depending on the deployment, an MA may be installed in the same system as a receiving client.

3.7 sender multicast agent (SMA): The MA attached to the sender in the same system or local network.

3.8 receiver multicast agent (RMA): The MA attached to the receiver in the same system or local network.

3.9 head multicast agent (HMA): A representative of the MA inside a local network where the multicast is enabled.

3.10 session manager (SM): An RMCP entity that is responsible for the overall RMCP operations; it may be located in the same system as the media server or located separately from the media server.

3.11 parent multicast agent (PMA): The next upstream MA in the RMCP-2 data delivery path.

3.12 child multicast agent (CMA): The next downstream MA in the RMCP-2 data delivery path.

3.13 RMCP-2 protocol: A relayed multicast protocol for simplex group applications.

3.14 basic RMCP-2 protocol: The relayed multicast protocol for simplex group application defined in clauses 5-8.

3.15 secure RMCP-2 protocol: The relayed multicast protocol supporting security features for simplex group applications defined in clauses 9-12.

3.16 dedicated multicast agent (DMA): An intermediate MA pre-deployed as a trust server by the Session Manager (SM) in an RMCP session.

3.17 security policy: The set of criteria for the provision of security services, together with the set of values for these criteria, resulting from agreement of the security mechanisms defined in 10.1.4.

3.18 TLS_CERT mode: A mode of the TLS defined in IETF RFC 4346 for the authentication of MAs using a certificate.

3.19 TLS_PSK mode: A mode of the TLS defined in IETF RFC 4279 for the authentication of MAs using a pre-shared key for the TLS key exchange.

3.20 relayed multicast region; RM region: A management zone defined by the use of the session key Ks.

3.21 member multicast region; MM region: A management zone defined by the use of one or more group keys Kg.

3.22 member multicast group; MM group:

1) (in a multicast disabled area) a group consisting of one DMA and multiple RMAs sharing the same group key Kg.
2) (in a multicast enabled area) a group consisting of one HMA, multiple RMAs together with one or more candidate HMAs sharing the same group key Kg.

3.23 candidate HMA: A DMA that is able to assume the role of an HMA, should the original HMA leave or be terminated from a multicast-enabled MM group.

3.24 group attribute (GP_ATTRIBUTE): An attribute that defines whether or not the Content Provider controls the admission of RMAs to the secure RMCP-2 session.

3.25 closed group: An MM group in which all the RMAs have been allocated a service user identifier from the Content Provider before subscribing to the secure RMCP-2 session.

3.26 **open group**: An MM group in which none of the RMAs require a service user identifier before subscribing to the secure RMCP-2 session.

3.27 **regular HB message**: An HB message that is relayed without interruption along the path of the RMCP-2 tree from the SMA to the receiver of the message. The originator of a regular HB message is the SMA.

3.28 **pseudo-HB message**: An HB message that indicates a fault in the delivery path of the RMCP-2 tree. The originator of a pseudo-HB message is the MA that discovers this fault.

4 Abbreviations

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

ACL	Access Control List
AUTH	Authentication
CEK	Contents Encryption Key
CMA	Child Multicast Agent
CP	Content Provider
DMA	Dedicated Multicast Agent
HANNOUNCE	HMA announce message
HB	Heartbeat message
HLEAVE	HMA leave message
HMA	Head Multicast Agent
HRSANS	Head Required Security Answer
HRSREQ	Head Required Security Request
HSOLICIT	HMA solicit message
IP-IP	IP in IP
KEYDELIVER	Key Delivery
LEAVANS	Leave answer message
LEAVREQ	Leave request message
MA	Multicast Agent
MAID	Multicast Agent Identification
PMA	Parent Multicast Agent
PPROBANS	Parent probe answer message
PPROBREQ	Parent probe request message
RELANS	Relay answer message
RELREQ	Relay request message
RMA	Receiver Multicast Agent
RMCP	Relayed MultiCast Protocol
SDP	Session Description Protocol
SECAGANS	SECurity AGreement ANSwer
SECAGREQ	SECurity AGreement REQuest
SECALGREQ	SECurity ALgorithms REQuest
SECLIST	Selected sECurity LIST
SID	RMCP-2 Session Identification
SMA	Sender Multicast Agent
STANS	Status report answer message
STCOLANS	Status report collect answer message
STCOLREQ	Status report collect request message
STREQ	Status report request message

SUBSANS	Subscription answer message
SUBSREQ	Subscription request message
T/TCP	TCP extensions to Transactions
TCP	Transmission Control Protocol
TERMANS	Termination answer message
TERMREQ	Termination request message
TLS	Transport Layer Security
UDP	User Datagram Protocol

5 Overview

The RMCP-2 is an application-level protocol that uses multicast agents (MAs) and a session manager (SM) to support and manage a relayed multicast data transport over a unicast-based Internet. With the help of the SM, the RMCP-2 begins by constructing a relayed multicast control tree that consists of MAs. Consequently with the preconfigured control tree, each MA connects appropriate data channels with each other.

The RMCP-2 entities for a simplex delivery model are described in clause 5.1.

5.1 RMCP-2 entities

The RMCP-2 entities are the same as those described in RMCP Part 1. As shown in Figure 1, each RMCP-2 session constructs a relayed multicast data delivery model with the following entities:

a) one SM;

b) one sender multicast agent (SMA) per sender application;

c) one or more receiver multicast agents (RMAs);

d) one or more sending or receiving group applications.

An SM, which can handle one or multiple sessions simultaneously, can be implemented separately or as a part of other entities in an RMCP-2 session.

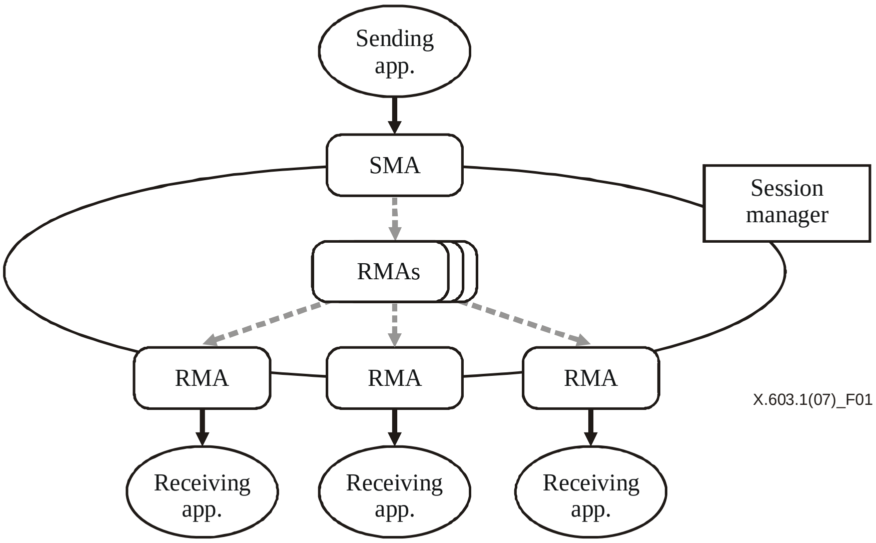


Figure 1 – RMCP-2 service topology

An SM can provide the following functionalities:

a) session initialization;

b) session release;

c) session membership management;

d) session status monitoring.

An MA, which refers to both the SMA and the RMA, constructs a relayed multicast delivery path from one sender to many receivers and then forwards data along the constructed path, can provide the following functionalities:

a) session initialization;

b) session join;

c) session leave;

d) session maintenance;

e) session status reporting;

f) application data relay.

5.2 RMCP-2 protocol block

An SM should exchange control messages with other MAs to control and manage RMCP-2 session. The control messages used by SM should be delivered reliably; otherwise, RMCP-2 session becomes unrecoverable. Figure 2 shows a protocol stack of an SM.

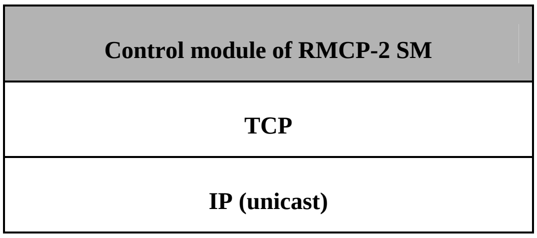


Figure 2 – Protocol stack of SM

An MA, which refers to both the SMA and the RMA, constructs a relayed multicast delivery path from one sender to many receivers and then forwards data along the constructed path. An MA consists of an *RMCP-2 control module* and a *data transport module*. The control module establishes the relayed data delivery path. The data transport module sets up a data channel along the path constructed by the control module and then relays data through the channel.

The MA's control module configures the control tree from the SMA to every leaf MAs by exchanging control messages with other MAs. Also the control module is used for session control and management by SM. Figure 3 shows the protocol stack of an MA's control module.

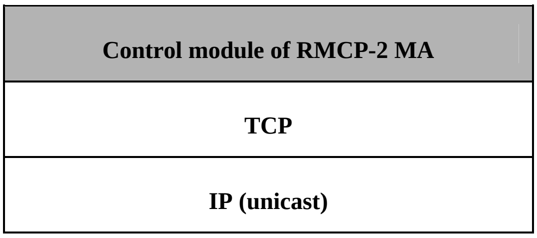


Figure 3 – Protocol stack of MA's control module

The MA's data module relays application data along the tree configured by the control module. Figure 4 shows the protocol stack of RMCP-2 data module. Any kind of transport mechanism can be inserted, if needed, because RMCP-2 imposes no restrictions on the type of application data to be delivered.

To ensure that RMCP-2 can adopt any kind of data transport mechanism, two MAs (namely, the parent multicast agent (PMA) and the child multicast agent (CMA)) construct a data delivery path on the control tree by exchanging the data profiles described later.

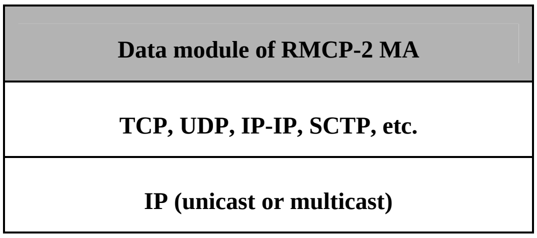


Figure 4 – Protocol stack of RMCP-2 data module

The topologies of the two paths for control and data delivery are usually the same, because a data delivery path is constructed along the RMCP-2 control tree. Along the data delivery path, the application data from the SMA can be delivered to each leaf MAs. For more information, Annexes B and C present two feasible real-time and reliable data delivery schemes.

5.3 Simplex delivery model of RMCP-2

The target services of RMCP-2 are *simplex broadcasting services*, such as Internet live TV and software dissemination. In those service models, building an optimal data delivery path from a sender to multiple receivers is important. RMCP-2 can support a simplex data delivery model by using the MA's control and data module.

The data delivery path that RMCP-2 considers is a *per-source relayed multicast tree*. Along the per-source relayed multicast path, a *unidirectional real-time or reliable data channel* can be constructed. Figure 5 shows one of the possible relayed multicast trees configured by RMCP-2 for *simplex real-time or reliable applications*.

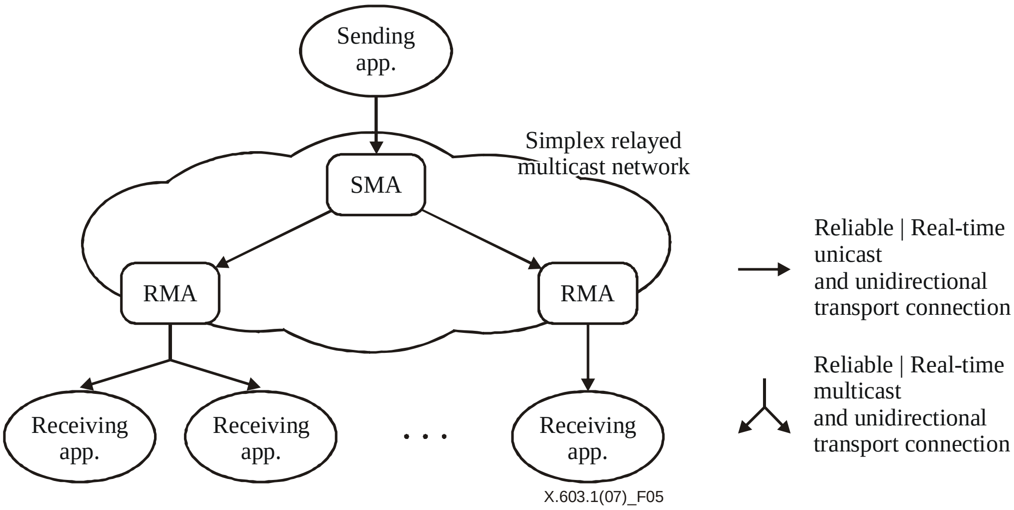

Sending app.
Simplex relayed multicast network
SMA
RMA
RMA
Receiving app.
Receiving app.
. . .
Receiving app.
Reliable | Real-time unicast and unidirectional transport connection
Reliable | Real-time multicast and unidirectional transport connection
X.603.1(07)_F05

Figure 5 – Relayed multicast tree configured by RMCP-2

5.4 Types of RMCP-2 messages

To construct and maintain a relayed multicast tree, several control messages are exchanged between RMCP-2 peers in a *request-and-answer* manner. Table 1 lists the RMCP-2 control messages according to the appropriate functions.

Table 1 – RMCP-2 messages

Messages	Descriptions	RMCP operations
SUBSREQ	Subscription request	Session initialization
SUBSANS	Subscription answer	
PPROBREQ	Parent probe request	MAP discovery
PPROBANS	Parent probe answer	
HSOLICIT	HMA solicit	HMA election
HANNOUNCE	HMA announce	
HLEAVE	HMA leave	

Table 1 – RMCP-2 messages

Messages	Descriptions	RMCP operations
RELREQ	Relay request	Data delivery
RELANS	Relay answer	
STREQ	Status report request	Session monitoring
STANS	Status report answer	
STCOLREQ	Status collect request	
STCOLANS	Status collect answer	
LEAVREQ	Leave request	Session leave
LEAVANS	Leave answer	
HB	Heartbeat	Session heartbeat
TERMREQ	Termination request	Session termination
TERMANS	Termination answer	

6 Protocol operation

This clause describes the RMCP-2 protocol functions and their operations in details. All the components described in this clause follow the definitions of Rec. ITU-T X.603 | ISO/IEC 16512-1.

6.1 SM's operation

6.1.1 Session initiation

To make the SM create a new session, a content provider (CP) should provide a session profile, which includes details to create a session such as the session name, media characteristics, and the group address. To distinguish the sessions from each other, the SM creates a globally unique session identification (SID). After a successful session creation, the SM returns the SID to the CP. The CPs may announce the session creation by using a web server or email. But the way of session announcement is out of scope this Specification.

After the successful session creation, the SM waits for a subscription request from the MAs. When the SM receives a subscription request from an MA, the SM decides whether to accept the subscription request.

6.1.2 Admission control

On receiving MA's subscription request, firstly the SM checks the SID in the request message, and then determines whether the request is acceptable according to the session policy. RMCP-2 session can be operated privately as well as publicly with some extra information such as system information.

When the SID in the MA's SUBSREQ is valid, then the SM checks proposed MAID and proposed data profile. If the MAID proposed by MA has null or duplicated value, then the SM proposes a unique one; otherwise, the proposed MAID will be used during the session. If the proposed data profile cannot be supported, the SM should reject the request with a reason. Otherwise, the SM can negotiate for the most effective data profile and sends back with the negotiated one.

When the MA's SUBSREQ is granted, then the SM responds with a confirmed MAID, NL and session dependent information.

To kick out a specific MA, the SM starts the discard procedure by sending a leave request (LEAVREQ) with a reason code Kicked-Out (KO) and then updates its session member list. Upon receiving SM's LEAVREQ message, MA leaves the session promptly. Figure 6 illustrates the procedure, where the SM sends a LEAVREQ message with the reason code KO and then the MA B leaves the session with notifying its PMA and CMAs of the expulsion.

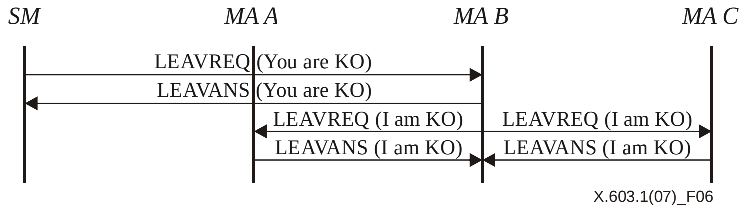


Figure 6 – When MA is kicked out by SM

6.1.3 Session monitoring

The SM can fetch status information of a specific MA by exchanging a status request and answer messages with any specific MA. Upon receiving the status request message, the MA responds with a status answer message that contains the requested information. Figure 7 shows how the SM monitors a specific MA.

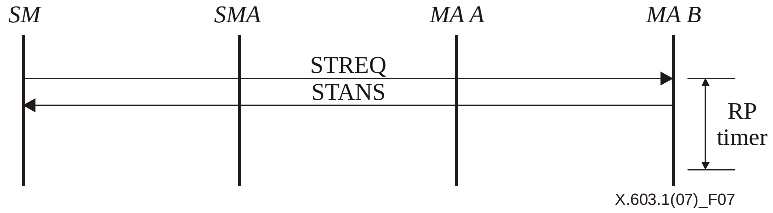


Figure 7 – Tree monitoring – Status report

SM can also collect status information of an entire or a part of a session. In this case, the SM sends a status collect request message to the top MA of the part. Upon receiving the status collect request message, the MA should send a status answer back to the SM with appropriate information on the MA and its children. When the session size is large, the use of this mechanism for the entire session may cause overloading the network and system resources. To limit the scope of the monitoring, the status collect message should contain an option for the depth.

6.1.4 Session termination

The SM's ongoing session may terminate due to one of the following two reasons:

1) administrative request; and
2) SMA's leave.

Figure 8 shows the SM's session termination procedure.

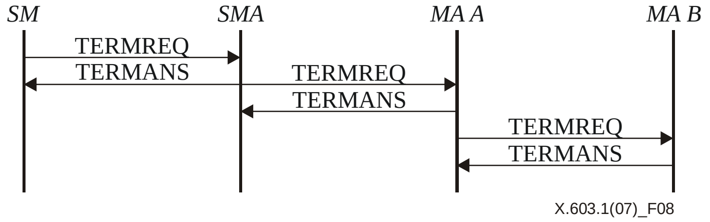


Figure 8 – Session termination issued by SM

Because a RMCP-2 session can continue only when the SMA is alive, the SMA must notify the SM when it leaves. Having been notified SMA's leave, the SM should terminate the session promptly. The session termination caused by SMA's leave is described in 6.2.4.4.

6.2 MA's operation

6.2.1 Session subscription

Subscription is the first stage for an MA to be enrolled in a RMCP-2 session. Each MA must subscribe to the session by sending a subscription request (SUBSREQ) to the SM. Note that the SMA must have finished its subscription before the other MAs and it should act as a root node in the tree hierarchy. At this stage, each MA needs to know details of the session profile, such as the address of the SM and the policy.

Figure 9 shows the procedure of RMCP-2 session subscription procedure. After SMA's successful subscription, RMCP-2 session can be initiated.

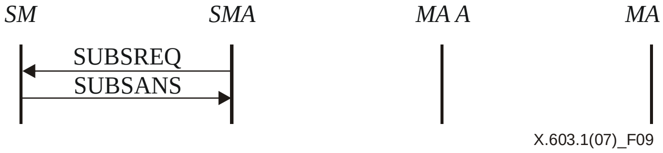


Figure 9 – SMA's subscription

Figure 10 shows the procedure of an MA subscription (for MA A and MA B). To subscribe an RMCP-2 session, each MA sends a SUBSREQ to the SM. Upon receiving SUBSREQ from the MA, the SM decides whether to accept the subscription request. If the request is accepted, the SM responds with a SUBSANS and bootstrapping information such as an NL. Otherwise, it responds with a SUBSANS with appropriate error reason code.

After receiving a successful SUBSANS from SM, the MAs (MA A and MA B) can complete the subscription phase.

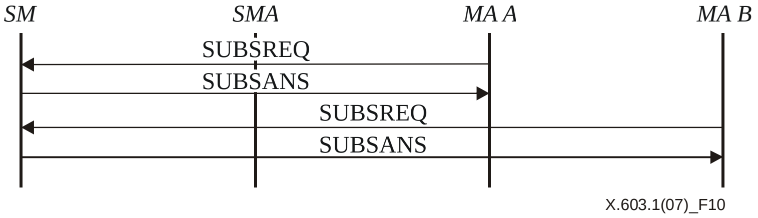


Figure 10 – MA's subscription

6.2.2 Map discovery

Since all MAs are logically interconnected, it would be difficult for a MA to know the entire network condition. However, by using map discovery procedures, each MA can explore the other MAs in the RMCP-2 network and measure the distance between itself and the other MAs. The map discovery mechanism consists of two steps. One is used in the multicast-enabled area, such as subnet LAN, and the other is used in the multicast-disabled area such as WAN.

6.2.2.1 Inside multicast-enabled area

It is desirable to assign the nearest node to its PMA. The network distance in RMCP-2 depends on the delay jitter, the hop count and the bandwidth.

Normally, an MA in the same network is closer than other MAs. Each MA looks for a candidate PMA in its local network by multicasting a head multicast agent solicit (HSOLICIT) to a specific pre-assigned address (aka, broadcast) at the beginning. If there is no answer, the MA becomes the HMA, which is a representative of the MA in the multicast-enabled network.

Once an MA becomes a HMA, the HMA announces its existence to the multicast-enabled network by sending periodic HANNOUNCE messages. The HMA sends a HANNOUNCE promptly on receiving HSOLICIT from the multicast-enabled area.

Upon receiving the HANNOUNCE from the HMA, each MA considers that a HMA already exists in the same network and then assumes the HMA as its primary PMA candidate. Figure 11 shows the HMA selection procedure.

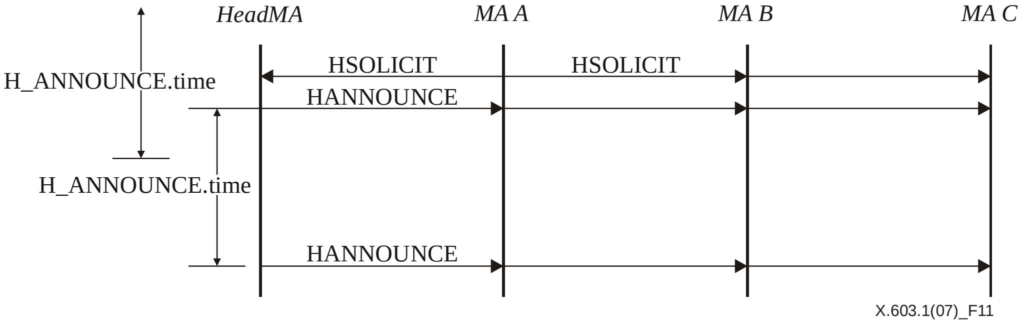


Figure 11 – HMA Solicit and its announcement

Figure 12 shows how an MA becomes a HMA. If there is no HANNOUNCE for a certain time (H_SOLICIT.time × N_SOLICIT), an MA becomes a new HMA and broadcasts a periodic HANNOUNCE every H_ANNOUNCE.time to the multicast-enabled area.

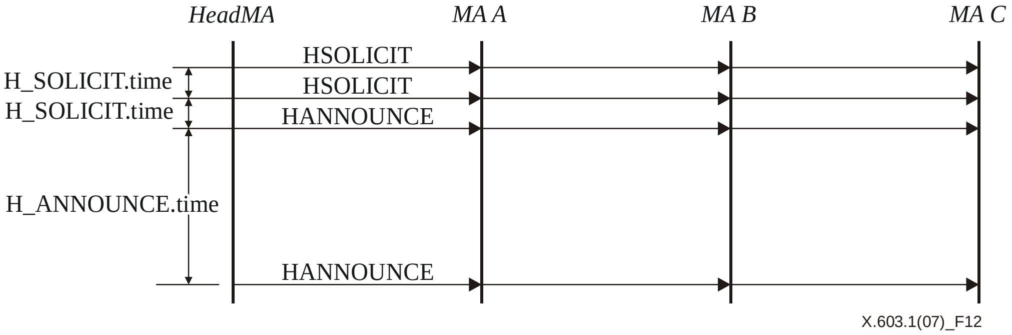


Figure 12 – An MA becomes a new HeadMA

Figure 13 shows how a HMA resumes. Once an MA becomes a HMA, it broadcasts a HANNOUNCE to the multicast-enabled network every H_ANNOUNCE.time.

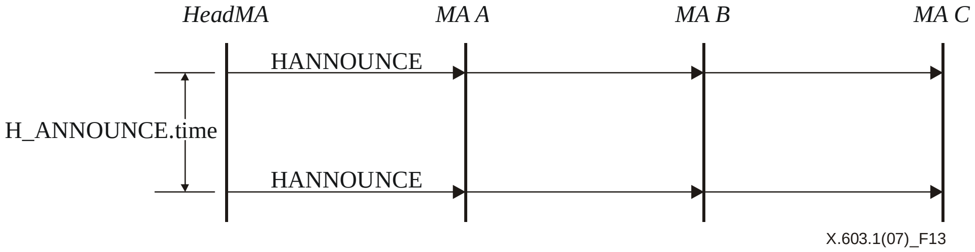


Figure 13 – Periodic head announce

Figure 14 shows how a new HMA is selected. If there is no HANNOUNCE for a certain time (H_ANNOUNCE.time × N_ANNOUNCE), the HMA waits for a HANNOUNCE for a random back-off time. If there is no HANNOUNCE, then the MA becomes the HMA of the multicast-enabled network. However, if there is a HANNOUNCE, then the MA discards the back-off time and selects the HMA as its primary PMA candidate. If there are more than two HANNOUNCE, the earliest HANNOUNCE sender becomes a HMA. If two or more HANNOUNCE have collided, then the HMA should follow the duplication suppression algorithm.

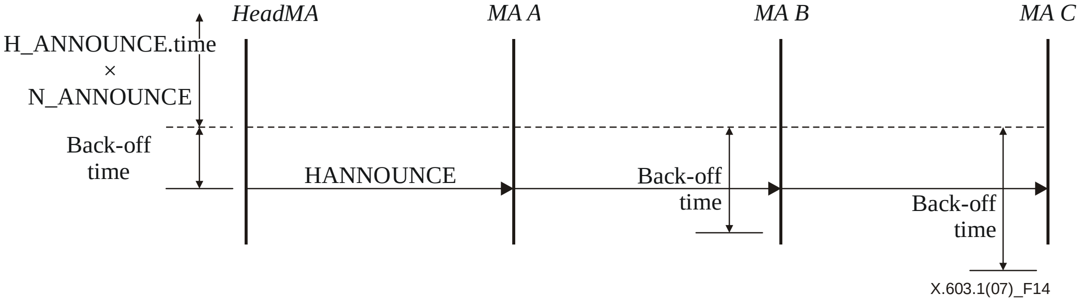


Figure 14 – New HMA selection

Because each MA in a multicast-enabled network can be elected as a HMA, each MA should also perform the map discovery mechanism for the outside network. The detailed procedure is discussed in the following subclause.

6.2.2.2 Outside multicast-enabled area

Each MA should start neighbour discovery procedure based on the initial bootstrapping information given by the SM. As shown in Figure 15, each MA can gradually learn the RMCP-2 tree topology by exchanging the tree information of each MA.

The basic map discovery mechanism is as follows: first, by using the PPROBREQ and PPROBANS, each MA can exchange a certain number of NLs at every interval (PPROBE.time). Because of the finite system resource of each MA, the maximum number of NLs to be exchanged should be bounded.

To prevent each MA suffered from PPROBREQ implosion, the maximum number of PPROBREQ messages for a certain period should be limited as N_MAX_PROBE.

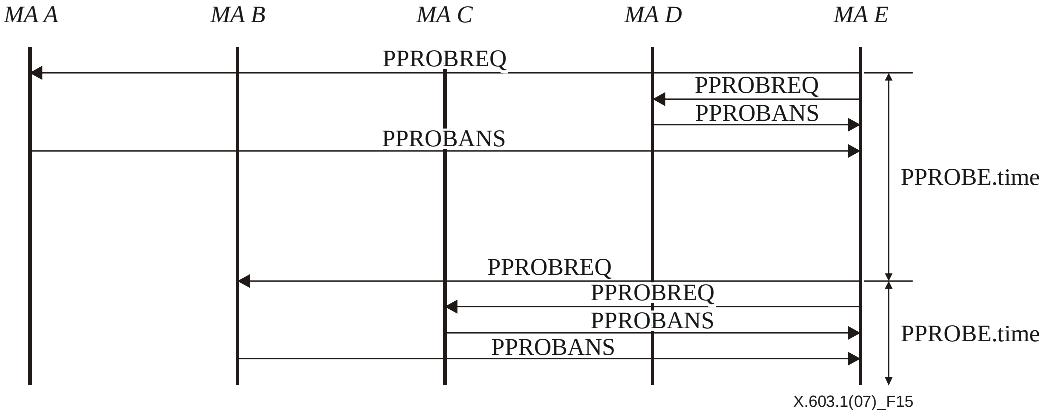


Figure 15 – Protocol sequence of map discovery

6.2.3 Tree join

Tree join procedure enables each MA to choose PMA inside a subscribed RMCP-2 session. Figure 16 shows how an MA selects its PMA based on the NL given by the SM. The joining MA (MA E) sends a PPROBREQ to one or more nodes listed in the NL (MA A, C, and D) and awaits a successful PPROBANS. Upon receiving a PPROBANS, the MA E can select the nearest MA. In Figure 16, the joining MA (node E) considers that the MA D is the best and then chooses the MA D as its PMA. After a PMA is selected, the joining MA (node E) will send to the MA D a RELREQ, which contains a proposed *data profile*.

If the RELREQ is acceptable, the MA D responds with a successful RELANS, which includes the negotiated *data profile* to be used. Otherwise, the MA D returns a reason code of the rejection.

Upon receiving a successful RELANS, data channel between the MA D and MA E is established according to the negotiated data profile. Otherwise, the MA E should try the second optimal PMA candidate.

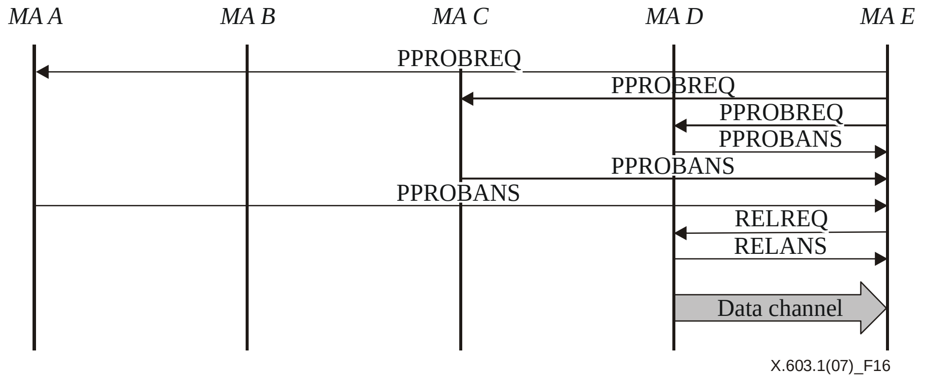


Figure 16 – Protocol sequence of successful tree join

If no MA wants to relay data to the joining MA, the joining MA can retry *tree join procedure* after a certain period. The retrial time can be set by the user, though this issue is beyond the scope of this Specification. Figure 17 shows when all the MAs listed in the NL given by the SM rejected node E's relay request. However MA E already learned about the existence of MA B during previous exchanges of PPROBREQ and PPROBANS, it can restart the joining procedure from MA B.

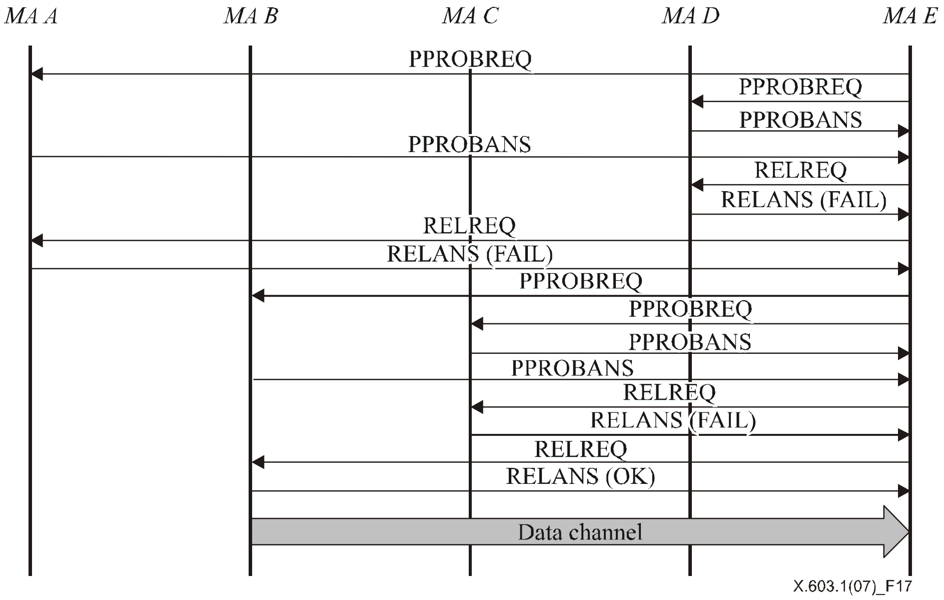


Figure 17 – Sequence of unsuccessful tree join and retrial

6.2.4 Leave

An RMCP-2 MA may leave a session during the session lifetime. To make a RMCP-2 tree robust, each MA should notify its departure to the PMA and CMAs. Upon receiving this notification, the PMA and each CMA should follow the appropriate procedure.

The RMCP-2 considers four types of departure. The first one refers to an MA that leaves the session at the request of a service user. The second one refers to an MA that leaves its PMA to switch parents. The third one refers to the expulsion of an MA from its PMA or SM. The final one refers to the departure of an SMA from a session. The detailed operations for the cases are described in the following subclauses.

6.2.4.1 When MA leaves a session

MAs may leave a session at any time during the session's lifetime. Before leaving, an MA must notify the PMA and CMAs of its departure. The PMA deletes the node from its CMA list and reserves a space for a new CMA.

a) *MA's leaving with multicast-disabled data delivery scheme*

To leave a session, an MA sends a LEAVREQ to its CMAs. Each CMA who receives the LEAVREQ should promptly start to connect to an alternative PMA by sending a RELREQ to the PMA candidate. If successful, each CMA sends its old PMA a LEAVANS.

Figure 18 shows how the MA C acts when the HMA leaves a session during which the multicast data delivery scheme is not used. MA C tries to leave the session by sending a LEAVREQ to MA D and MA E, which are the CMAs of MA C. On receiving the LEAVREQ, MA D and MA E each sends a RELREQ to their own PMA candidate.

After each MA has successfully attached to a new PMA (MA A and MA B), each MA (MA D and MA E) sends a LEAVANS to the current PMA (MA C). Upon receiving the LEAVANS from its CMAs, the MA C sends a LEAVREQ to its PMA (MA B). The PMA subsequently frees the MA from its CMA list. Any departing MA without CMA simply sends a LEAVREQ to its PMA.

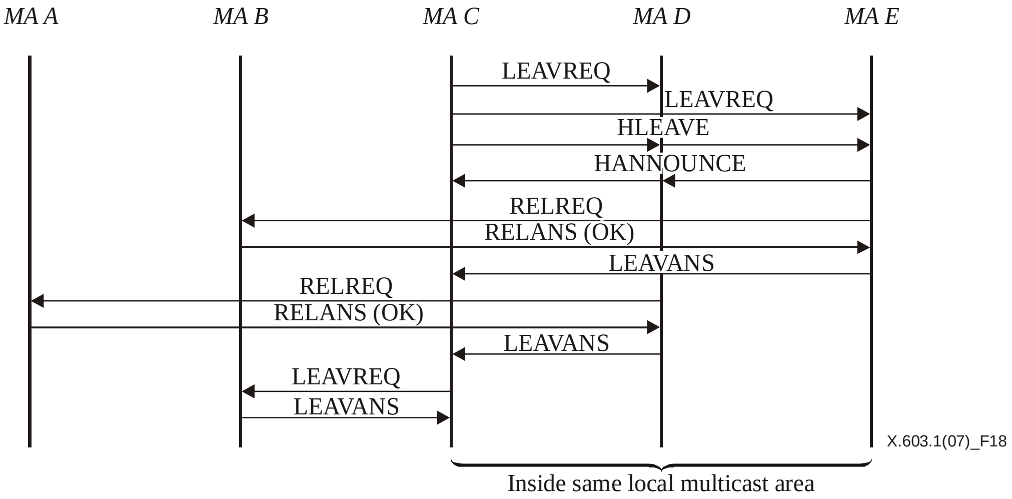


Figure 18 – HMA's leaving with multicast-disabled data delivery scheme

Figure 19 shows how a MA, which is not a HMA, leaves a session when a multicast-disabled data delivery scheme is used. In this scenario, the procedures of leaving for a non-HMA and the HMA are the same, except the HMA follows the HLEAVE exchanging sequence.

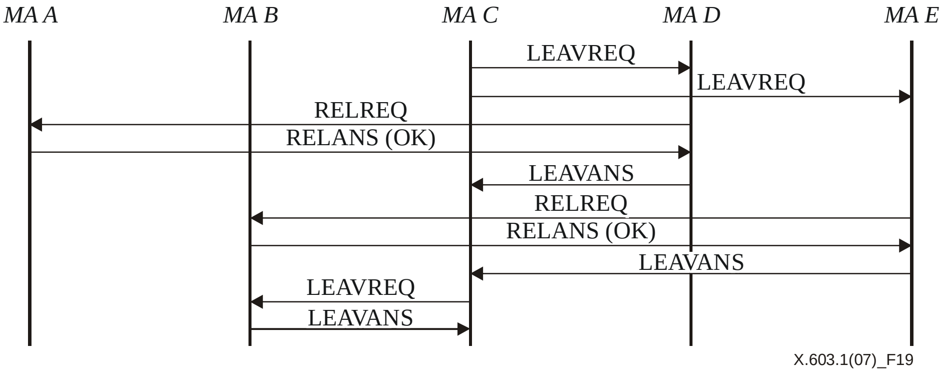


Figure 19 – Normal MA's leaving with multicast-disabled data delivery scheme

b) *MA's leaving with multicast-enabled data delivery scheme*

There are two cases of MA's leaving within a multicast-enabled area. The first case is of HMA's leaving and the other is of MA's leaving. Whenever the HMA of a multicast-enabled area wants to leave a session, it should notify its departure to the CMAs inside the local network as well as to the CMAs and the PMA outside the network.

Figure 20 shows how the MA C, which acts as HMA, leaves a session where the multicast data delivery scheme is used. The HMA (MA C) sends a LEAVREQ to its direct CMA (MA F) outside the local network. Upon receiving the LEAVREQ, MA F starts to switch parents and responds to MA C with a LEAVANS as well as multicasts a HLEAVE with an empty HMA candidate list to the local network. The HLEAVE message is used to announce the departure of the HMA.

Upon receiving the HLEAVE from HMA, both MA D and MA E from Figure 20 wait for a certain back-off time before multicasting the HANNOUNCE. The MA D sends the HANNOUNCE for the first time and becomes a new HMA. This step occurs because the MA D has a shorter back-off time than any other MA. Because the leaving MA C is a point which is connected to outside multicast-enabled network, the MA D should undertake the role of the MA C by connecting to the PMA outside of the network. Figure 20 shows how the MA D selects for its parent the MA B, which is the PMA of the MA C.

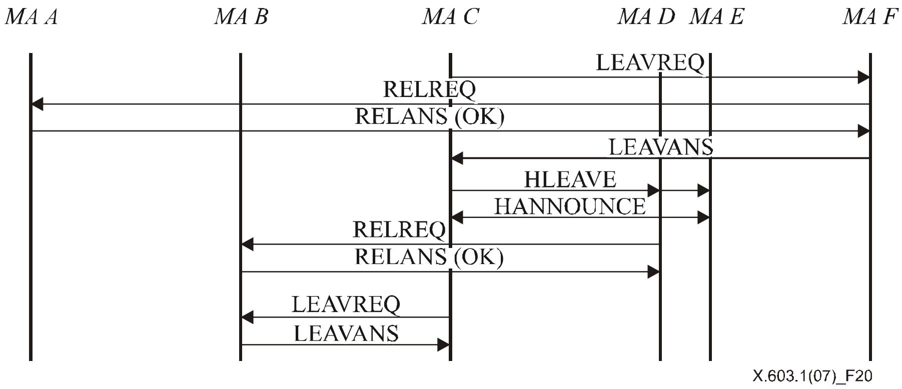


Figure 20 – MA's leaving with multicast-enabled data delivery scheme

Whenever any non-HMA of a multicast-enabled area wants to leave a session, it silently leaves the session. The MA D or MA E from Figure 20 does not need to notify other MAs of its departure.

6.2.4.2 When MA leaves from its PMA – for parent switching

An MA that wants to switch its PMA may leave its current PMA. In this case, the MA does not need to send a LEAVREQ to its CMAs. The CMAs do not need to know about the departure as long as they successfully receive data. To switch PMA, the MA sends a RELREQ to the other PMA candidate. An old PMA that receives a LEAVREQ with the reason code set to PS (parent switching) deletes the leaving MA from its CMA list but keeps the information of the departing MA in its NL because the leaving MA is still alive in the session.

Figure 21 shows how an MA switches its parents. Note that an MA can switch parents only when it receives a HB to keep tree unchanged. The HB mechanism is described in 6.2.5.1.

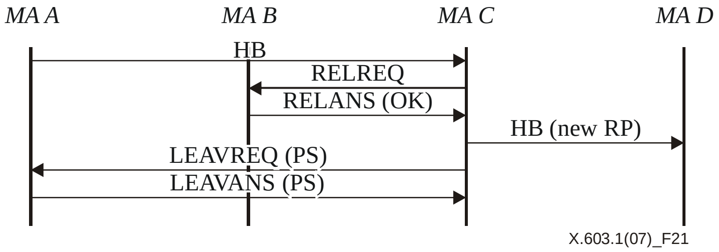


Figure 21 – MA's leaving for parent switching

6.2.4.3 When MA is kicked out

RMCP-2 has a mechanism for discarding certain MAs. For example, when a network manager wants the SM to discard a specific MA; and when an MA expels a CMA after it was aware that it cannot support more CMAs.

a) *Expulsion of an MA by its PMA*

A PMA can expel one of its CMAs when the PMA suffers from depleted system resources and can no longer feed its CMA, or when the PMA finds that one of its CMAs has depleted the system resources. An MA should find another PMA candidate, which would allow for a new CMA.

Figure 22 shows an example of a message flow. First, a PMA, namely the MA C, sends a LEAVREQ with a reason KO to expel MA D. The MA D searches other PMAs and sends a relay request. After switching parents, MA D transmits a LEAVANS to its old PMA.

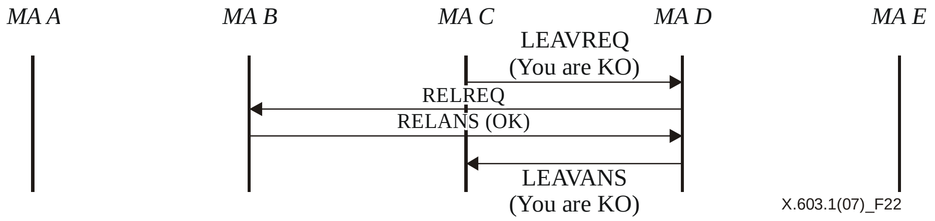


Figure 22 – When MA is kicked out by its PMA

b) *Expulsion of an MA by the SM*

The SM can discard any MA by sending a LEAVREQ with a reason kicked-out (KO). Upon receiving LEAVREQ from SM, an MA must leave the session promptly. After the expulsion, the SM should update its session member list.

In the message flow shown in Figure 23, the SM tells MA B to leave by sending a LEAVREQ with a reason KO. MA B must leave the session but, before leaving, MA B must notify its PMA and CMAs of its expulsion.

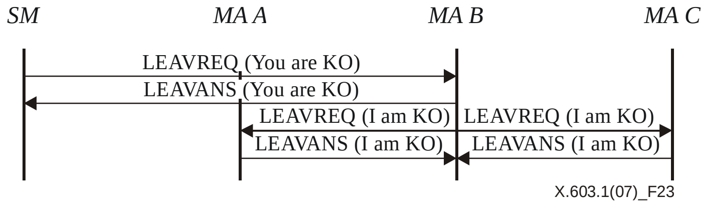


Figure 23 – When MA is kicked out by SM

6.2.4.4 When SMA leaves the session

Because an RMCP-2 session cannot exist without an SMA, an SMA never leaves a session before the session is terminated. In this case, when the SMA leaves the session, the session should be terminated.

Figure 24 shows the departure procedure of an SMA from a session. The SMA sends a LEAVREQ to the SM. Upon receiving the LEAVREQ from the SMA, the SM removes the session information and then replies with LEAVANS. Upon receiving the LEAVANS from the SM, the SMA sends a LEAVREQ with reason *SMA leave* to its direct CMAs. The LEAVREQ with reason *SMA leave* should be relayed downward promptly to make RMCP-2 session terminated.

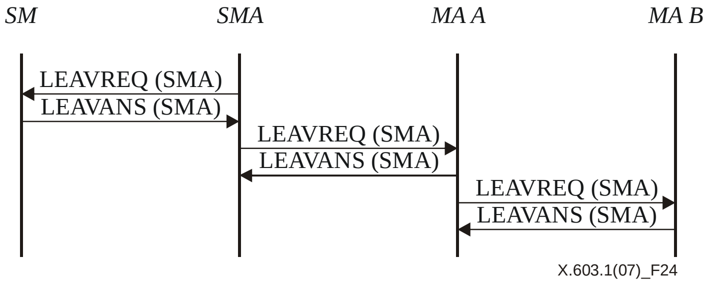


Figure 24 – SMA's leaving

6.2.5 Maintenance

6.2.5.1 Heartbeat

The purpose of the heartbeat is to keep the constructed RMCP-2 tree robust. The heartbeat, which gives unified synchronizing information to the session, helps each MA detect whether the session is currently alive. It also contains useful information on the data delivery path, named ROOTPATH. The ROOTPATH includes a relayed data path which follows the tree hierarchy.

Figure 25 shows the RMCP-2 heartbeat procedure. In this procedure, the SMA sends the HB, along the ROOTPATH, to its descendants; each descendant then appends the hop information, which may include MAID, per-hop network distance and system information such as in-and-out bandwidth, affordable number of CMA, etc. to the HB and forwards the modified HB to its descendants. Finally, the ROOTPATH contains all the MAs visited along the tree.

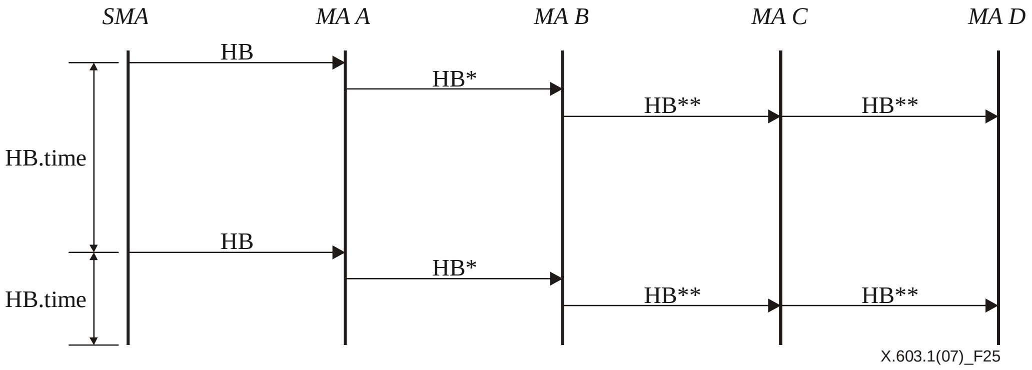


Figure 25 – Heartbeat

6.2.5.2 Monitoring

RMCP-2 has two types of monitoring mechanisms. The first one, which is shown in Figure 26, monitors a specific MA. The other one, which is shown in Figure 27, monitors a part of the tree through a specific MA.

Figure 26 shows how an SM monitors a specific MA. In this procedure, the SM sends an STREQ to MA B and requests one or more specific types of status information from MA B. In response, MA B sends the SM a STANS with the requested information.

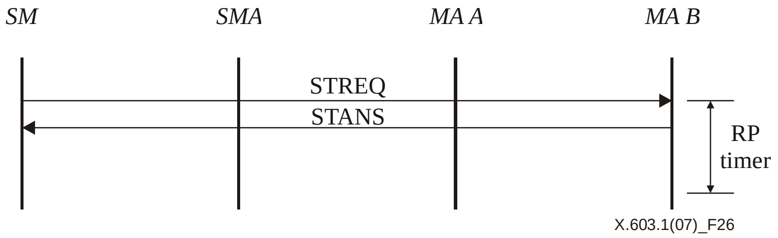


Figure 26 – Tree monitoring by status report

Figure 27 shows how the SM queries the scoped area of a tree. That is, the SM asks for merged information on the scoped area of a tree by sending an STREQ to a specific MA (SMA and MA A each) to collect status information for the scoped area.

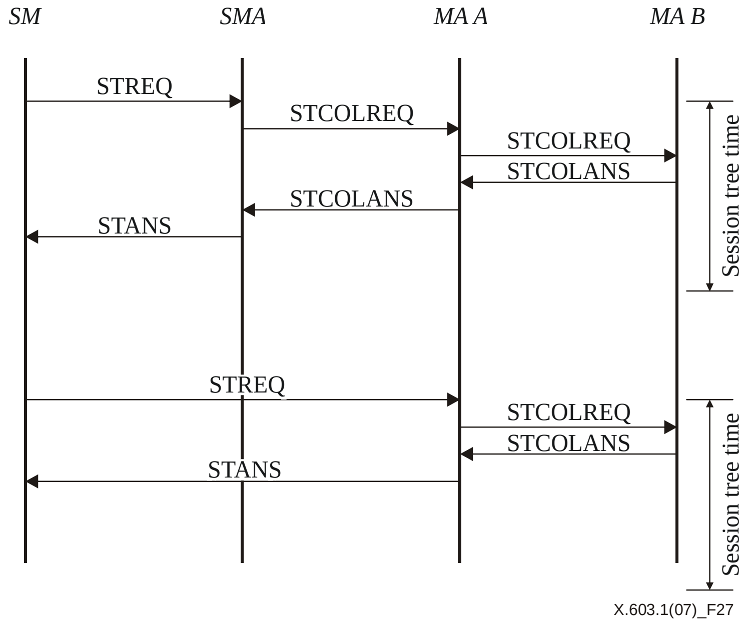


Figure 27 – Tree monitoring by collecting status report

6.2.5.3 Fault detection and recovery

This procedure is performed by each MA when each MA detects network faults and recovers from the problems to make the RMCP-2 tree robust. Network faults such as looping or partitioning are often caused by MA's frequent and careless movements. To detect and recover such network faults, RMCP-2 provides the following fault detection and recovery mechanisms.

a) *Loop detection and recovery*

A loop can be detected by checking the ROOTPATH contained in HB. Because the ROOTPATH gives the path track from the SMA to itself, the duplicated hop in the ROOTPATH means that a loop has formed. Whenever a loop occurs, each MA performs the following loop recovery mechanism: for the scenario described in Figure 28, MA Y examines the HB; MA Y then confirms the existence of a loop whenever it receives HB_{n+3} because MA Z, which is a CMA of MA Y, is already listed in the ROOTPATH twice. To recover from the loop, MA Y sends MA Z a LEAVREQ message to disconnect.

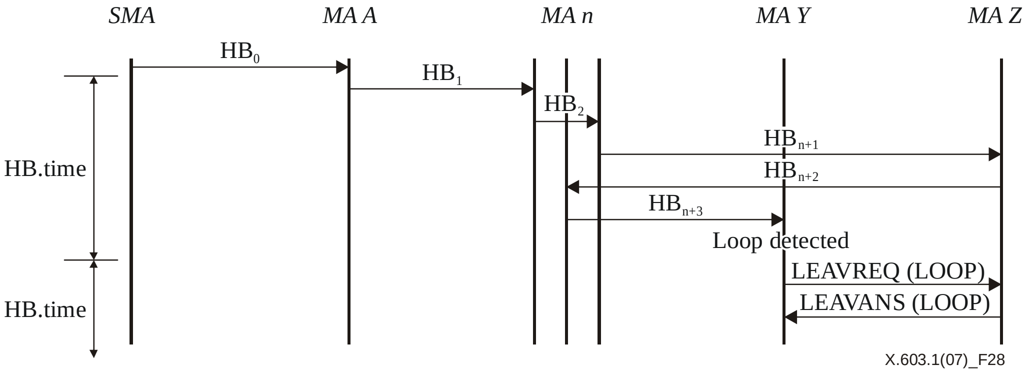


Figure 28 – Loop detection and recovery

b) *Network partitioning detection and recovery*

Whenever an MA fails to receive the HB message for a certain time, the MA assumes that it is partitioned from the tree. The time should be set for sufficient time to allow for a network delay. RMCP-2 defines the time as HB_TIME × MAX_PARTITION_CNT.

A partition can occur whenever one of the partition's associates fails. The MA detects the source of the partitioning by contacting its associates; the MA then solves the problem.

Figure 29 shows how MA Z detects tree partitioning: that is, a tree partition is detected whenever MA Z fails to receive the HB message for a certain period (HB_TIME × MAX_PARTITION_CNT). The failure to receive the HB message triggers the transmission of a number of PPROBREQ messages towards its associates. In Figure 29, MA Z receives a

PPROBANS message from MA A and MA B but no response from MA C, the current PMA of MA Z. MA Z detects that the partitioning occurs as a result of the failure of the direct PMA of MA Z; MA Z then tries to switch parents in order to recover from the partitioning.

During an MA's repairing the partition, the MA's descendants may also consider that the network has partitioned and they may start to repair the partition. As a result, an MA's failure in just one point can cause an entire tree to collapse. To prevent this problem, an MA, which is repairing a network fault, generates a pseudo HB message to its descendants to notify that the session is temporarily partitioned and being recovered.

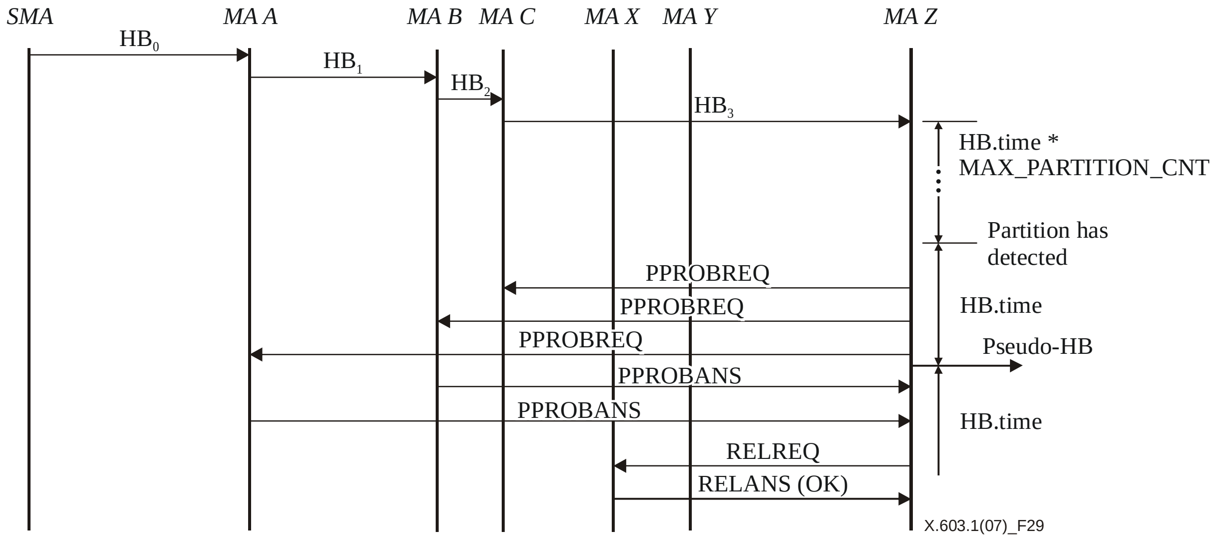


Figure 29 – Network partitioning detection and recovery

6.2.5.4 Tree improvement

Tree improvement procedure occurs when an MA finds one or more efficient PMA candidates and tries switching to the found one. By continuing the tree improvement procedure during the session, RMCP-2 tree can be improved gradually.

The procedure for finding better nodes follows the map discovery mechanism described in 6.2.2. At every turn of the map discovery, each MA compares the QoS parameters of its current PMA with those of the newly discovered node. When an MA finds a better MA than its current PMA, then the MA can switch its current PMA to a newly discovered MA according to the parent switching procedure described in 6.2.4.2.

While the tree is being improved, network faults such as a loop or partition can easily occur. In particular, network faults may occur in the following cases: when multiple MAs in the same branch may try to switch their PMAs at the same time and when multiple MAs along the branch may try to successively switch their PMAs.

To keep a tree from these hazards, RMCP-2 guarantees the atomic condition, in which each MA can switch a parent only after receiving a HB message with an unchanged ROOTPATH.

6.2.6 Termination

To terminate a session, the SM sends a TERMREQ to SMA as shown in Figure 30. An SMA (or MA) that receives a TERMREQ message from the SM (or PMA) sends the TERMANS message back to the SM (or PMA) and then forwards the TERMREQ message to its CMAs until it reaches the end nodes of the tree. Finally, the session is closed.

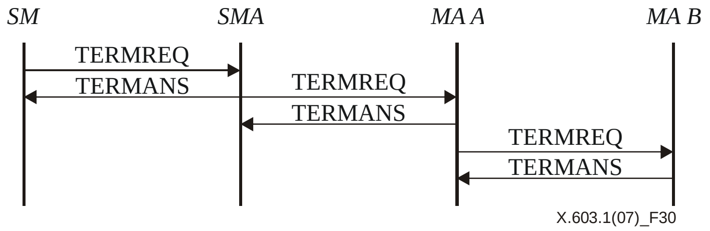


Figure 30 – Session termination issued by SM

7 RMCP-2 message format

This clause describes the formats and required information of the RMCP-2 messages. The corresponding value information of each message will be explained in clause 8.

7.1 Common format of RMCP-2 message

Figure 31 shows common RMCP-2 message format.

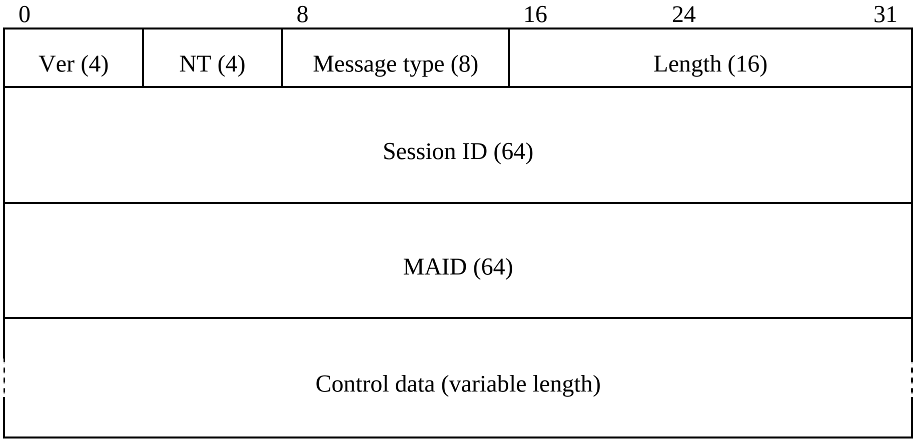


Figure 31 – Common RMCP-2 message format

The description of each field is as follows:

a) *Version* – It represents the current RMCP version. The default value for RMCP-2 is set to 0x2.

b) *NT (Node Type)* – It represents the type of node. It must be set to identify itself such as SM, SMA and MA.

c) *Message type* – It represents the type of the message.

d) *Length* – It represents the total length of the message in bytes including control data.

e) *Session ID* – It is a 64-bit integer that identifies a session.

f) *MAID* – It is a 64-bit unique value used to identify the MA for a certain session.

g) *Control data* – It contains control data used by each message as needed.

Session ID and MAID must be a unique value to identify the session and MA, respectively. RMCP-2 provides a generation rule of the ID value used for a session and MA.

7.1.1 Session ID

Session ID (SID) is a combination of the local IP address of the Session Manager (SM) and the group address of the session. The group address for a new session can be allocated by SM when the SM is requested to create a session. By doing this, the SID can be guaranteed as globally unique. Figure 32 illustrates the RMCP-2 SID format.

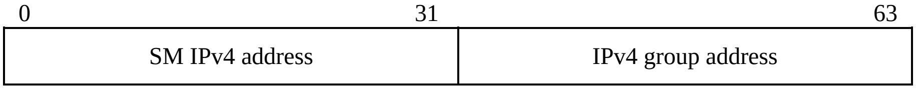


Figure 32 – RMCP-2 SID format

7.1.2 MAID

MAID consists of the local IP address, port number, and serial number as shown in Figure 33. The local IP address is the IP address of the MA. An MA in a RMCP-2 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-2 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.

0	31	63
Local IP address	Port	Serial

Figure 33 – RMCP-2 MAID format

Figure 34 shows the algorithm that the current version of RMCP-2 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 34 – A simple algorithm to generate a unique MAID

7.2 Control data format

Figure 35 shows the RMCP-2 control data format.

0	8	16 n
Control type (8)	Length (8)	Value (variable-size)

Figure 35 – RMCP-2 control format

a) *Control type* – It represents the type of control data.

b) *Length* – It represents the length in byte of control data value as well as type and length field except sub-control data field.

c) *Value* – It contains the value of control data.

Whenever RMCP-2 control data wants to specify its control in detail, it can apply sub-option data. The format of sub-option data takes that of RMCP-2 control data as shown in Figure 36.

0	8	16 n
Sub-control type (8)	Length or number (8)	Value (variable-size)

Figure 36 – RMCP-2 sub-control format

a) *Sub-control type* – It describes the type of sub-control data.

b) *Length or number* – It represents the length in byte or the number of sub-control data value depending on the sub-control data value.

c) *Value* – It represents the value of sub-control data.

A control data can be represented by using only one control data alone as shown in Figure 37.

0 n – 1 n
Control data (Type A)

Figure 37 – Usage of control data alone

Whenever sub-control data is used, an appropriate control data must precede. Figure 38 shows an appropriate control data must precede the sub-control data to be used.

0 n – 1	n n + m – 1
Control data (Type B)	Sub-control data (Type b)

Figure 38 – Usage of control data with sub-control data

One or more control data can be located in RMCP-2 control data field at once. When a RMCP-2 packet wants to include multiple control data, it should align multiple control data as shown in Figure 39.

Control data (Type A)	Control data (Type D)	Sub-control data (Type d)	Control data (Type E)

Figure 39 – Usage of multiple control data

7.3 Messages

This clause defines each message used in RMCP-2. The message types and corresponding values for the messages are listed in Table 23.

7.3.1 SUBSREQ message

7.3.1.1 General

The SUBSREQ message is used to subscribe to a RMCP-2 session.

<table>
<tr><th>0</th><th>4</th><th>8</th><th>16 31</th></tr>
<tr><td>Ver (0x2)</td><td>NT
(SMA|MA)</td><td>Message type
(SUBSREQ)</td><td>Length (variable)</td></tr>
<tr><td colspan="4">Session ID (64)</td></tr>
<tr><td colspan="4">MAID (MAID proposed by the subscriber)</td></tr>
<tr><td colspan="4">Control data (variable length)</td></tr>
</table>

Figure 40 – SUBSREQ message

7.3.1.2 SUBSREQ message format

The format of the SUBSREQ message is shown in Figure 40. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the SUBSREQ message. Its value shall be set to 0x01 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of the SUBSREQ message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID proposed by the subscriber. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the SUBSREQ message, and their status, are shown in Table 2.

Table 2 – Control types for the SUBSREQ message

Control type	Meaning	Status	Reference
SYSINFO	A description of the system information of MA.	Optional	See 7.3.1.3
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.1.4

7.3.1.3 SYSINFO control

The SYSINFO control in the SUBSREQ message is used to convey system information about the subscribing MA in its SYSINFO sub-controls.

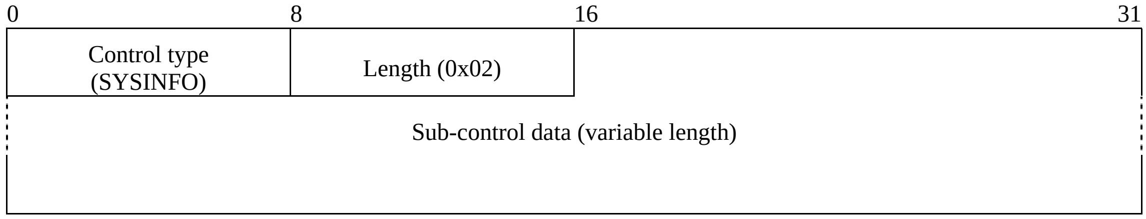


Figure 41 – SYSINFO control

The format of the SYSINFO control is shown in Figure 41. The description of each field is as follows:

a) *Control type* – This field denotes the SYSINFO control. Its value shall be set to 0x08 (see Table 24).

b) *Length* – This field denotes the length (2 bytes) of the SYSINFO control. Its value shall be set to 0x02.

c) *Sub-control data* – The SYSINFO sub-control types that may be used in the SUBSREQ message are listed in Table 3. If more than one SYSINFO sub-control is required, each sub-control shall be preceded by a two-byte SYSINFO control.

Table 3 – SYSINFO sub-control types for the SUBSREQ message

Sub-control type	Meaning	Status	Reference
SI_ROOM_CMA	The number of CMA places that an MA has allocated and the total number that it is able to support.	Optional	See 7.3.11.4.3
SI_POSS_BW	The possible forwarding bandwidth that the MA can afford.	Optional	See 7.3.11.4.5

NOTE – The additional SYSINFO controls defined for other RMCP-2 messages are not relevant for session subscription as they relate to the position once the MA has joined the RMCP-2 tree.

7.3.1.4 DATAPROFILE control

The DATAPROFILE control is used to describe the proposed data profile of the subscribing MA.

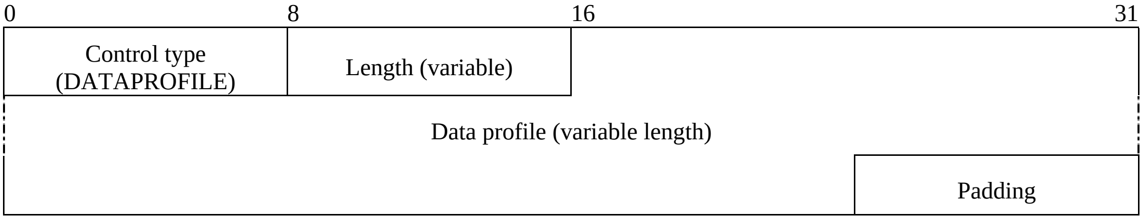


Figure 42 – DATAPROFILE control

The format of the DATAPROFILE control is shown in Figure 42. The description of each field is as follows:

a) *Control type* – This field denotes the DATAPROFILE control. Its value shall be set to 0x03 (see Table 24).

b) *Length* – This field denotes the length in bytes of the DATAPROFILE control. Its value shall be a multiple of four bytes (see item d) in this list) and it shall not exceed 0xFC.

c) *Data profile* – This field shall contain the data profile for the MA formatted in text mode. It follows an SDL-like encoding scheme. An example is shown in Figure 87.

d) *Padding* – If the total length of the control type, length and data profile fields is not a multiple of 4 bytes, the padding field shall be filled with zeros to ensure that the length of the DATAPROFILE control is a multiple of 4 bytes.

7.3.2 SUBSANS message

7.3.2.1 General

The SUBSANS message is used by SM to provide the results of subscription request and bootstrapping information for the session.

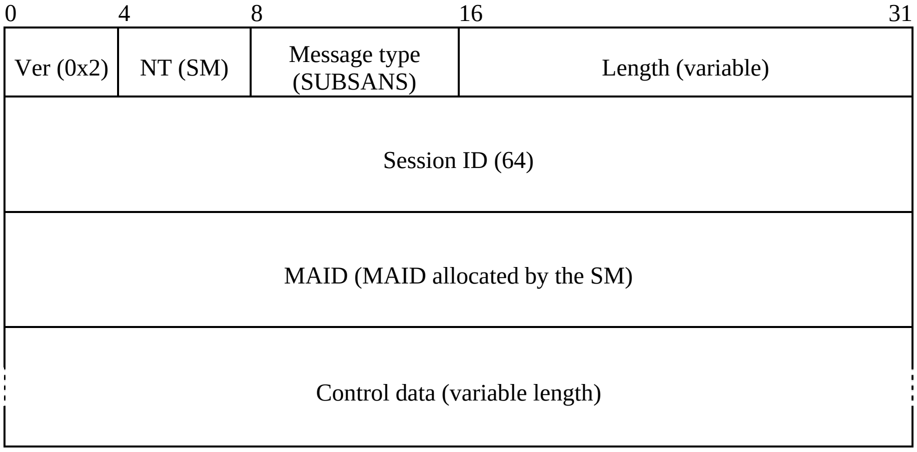


Figure 43 – SUBSANS message

7.3.2.2 SUBSANS message format

The format of the SUBSANS message is shown in Figure 43. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for SM in Table 22.

c) *Message type* – This field denotes the SUBSANS message. Its value shall be set to 0x02 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of the SUBSANS message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the subscriber as allocated by the SM. Its value shall be formatted as defined in 7.1.2.

NOTE – This may not be identical to the MAID proposed by the subscriber (see 6.1.2).

g) *Control data* – The control types that may be used in the SUBSANS message, and their status, are shown in Table 4.

Table 4 – Control types for the SUBSANS message

Control type	Meaning	Status	Reference
RESULT	The result of the subscription request.	Mandatory	See 7.3.2.3
NEIGHBORLIST	A list of MAIDs for performing the map discovery.	See condition 1	See 7.3.2.5
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.2.4
Condition 1: If the RESULT is successful, the NEIGHBORLIST is mandatory; if not, the NEIGHBORLIST shall not be included.			

7.3.2.3 RESULT control

The RESULT control in a SUBSANS message is used to convey whether or not the MA's subscription request is successful. If successful, it returns an OK result code. If not, it returns an appropriate error code.

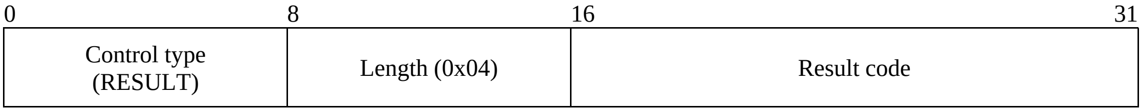


Figure 44 – RESULT control

The format of the RESULT control is shown in Figure 44. The description of each field is as follows:

a) *Control type* – This field denotes the RESULT control. Its value shall be set to 0x06 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the RESULT control. Its value shall be set to 0x04.

c) *Result code* – This field denotes the result of the request. It shall be set to one of the result codes listed in Table 25.

7.3.2.4 DATAPROFILE control

The DATAPROFILE control is used by the SM to confirm the data profile proposed by the subscriber, or to provide extra data forwarding information to the subscriber.

The content and format of the DATAPROFILE control are specified in 7.3.1.4, Figures 42 and 87.

7.3.2.5 NEIGHBORLIST control

The NEIGHBORLIST control in a SUBSANS message to a successful subscriber is used to convey a list of active MAs that may be used for bootstrapping purpose.

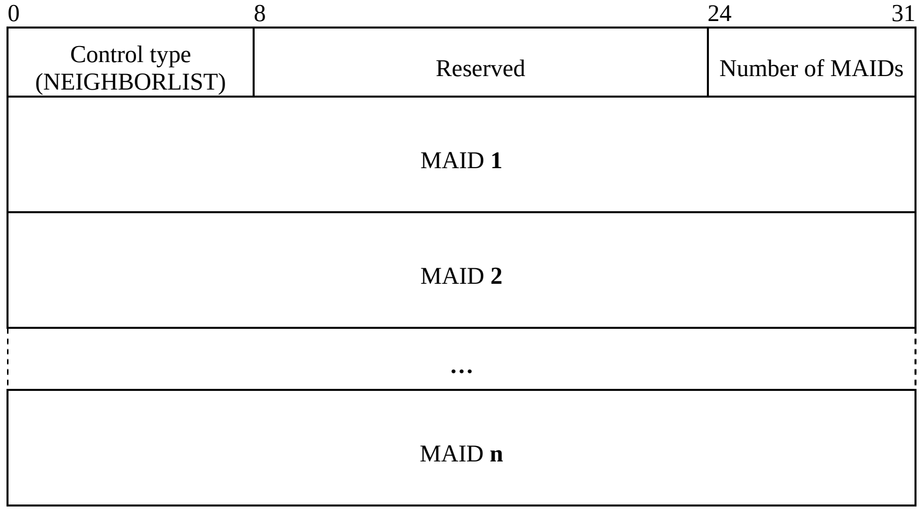


Figure 45 – NEIGHBORLIST control

The format of NEIGHBORLIST control is shown in Figure 45. The description of each field is as follows:

a) *Control type* – This field denotes the NEIGHBORLIST control. Its value shall be set to 0x04 (see Table 24).

b) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

c) *Number of MAIDs* – This field shall be set to the number of MAIDs listed in NEIGHBORLIST control.

d) *MAID(s)* – These fields MAID *1* to MAID *n* shall contain a list of MAIDs up to 255 active neighbours.

7.3.3 PPROBREQ message

7.3.3.1 General

The PPROBREQ message is used in the map discovery procedure to explore network conditions and to identify potential near neighbour. It is also used to check whether the neighbouring MA is still active.

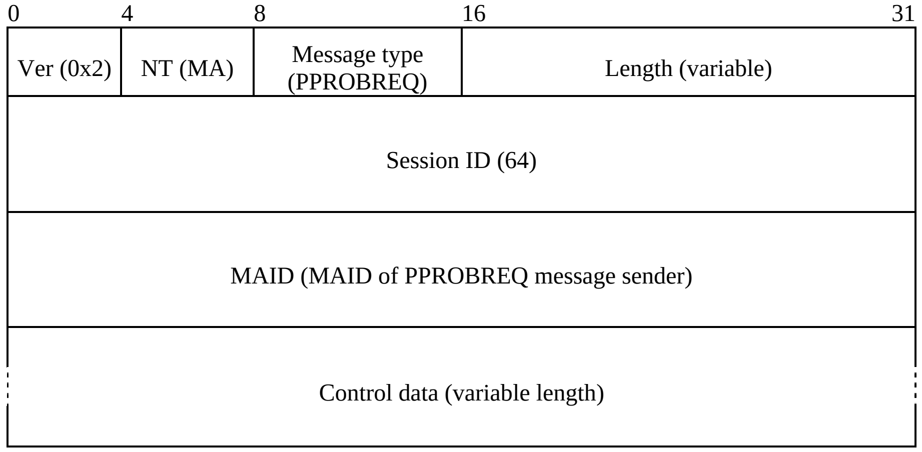


Figure 46 – PPROBREQ message

7.3.3.2 PPROBREQ message format

The format of the PPROBREQ message is shown in Figure 46. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA in Table 22.

c) *Message type* – This field denotes the PPROBREQ message. Its value shall be set to 0x03 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of the PPROBREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the PPROBREQ message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the PPROBREQ message, and their status, are shown in Table 5.

Table 5 – Control types for the PPROBREQ message

Control type	Meaning	Status	Reference
TIMESTAMP	A measure of transmission time between sending and receiving MAs.	Mandatory	See 7.3.3.3
NEIGHBORLIST	A list of MAIDs for performing the map discovery.	Optional	See 7.3.3.4
ROOTPATH	A description of the path from the SMA.	Optional	See 7.3.3.5
SYSINFO	A description of the system information of MA.	Optional	See 7.3.3.6
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.3.7

7.3.3.3 TIMESTAMP control

The TIMESTAMP control is used to measure transmission time between the sending MA and the receiving MA.

<table>
<tr><td>0</td><td>8</td><td>16</td><td>31</td></tr>
<tr><td>Control type
(TIMESTAMP)</td><td>Length (0x10)</td><td colspan="2">Reserved</td></tr>
<tr><td colspan="4">Time 1 (when the sender starts to send)</td></tr>
<tr><td colspan="4">Time 2 (when the packet appears at receiver)</td></tr>
<tr><td colspan="4">Time 3 (when the receiver starts to reply)</td></tr>
</table>

Figure 47 – TIMESTAMP control

The format of the TIMESTAMP control is shown in Figure 47. The description of each field is as follows:

a) *Control type* – This field denotes the TIMESTAMP control. Its value shall be set to 0x09 (see Table 24).

b) *Length* – This field denotes the length (16 bytes) of the TIMESTAMP control. Its value shall be set to 0x10.

c) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

d) *Time 1* – This field shall be set to the time when the request message is started to be sent to its recipient.

e) *Time 2* – This field shall be set to the time when the request message appears at the recipient. When this field is included in a request message, its value shall be set to zero.

f) *Time 3* – This field shall be set to the time when the answer message is started to be sent to the requestor. When this field is included in a request message, its value shall be set to zero.

7.3.3.4 NEIGHBORLIST control

The NEIGHBORLIST control in a PPROBREQ message is used to convey neighbour list information held by the probing MA.

The content and format of the NEIGHBORLIST control are specified in 7.3.2.5 and Figure 45.

7.3.3.5 ROOTPATH control

The ROOTPATH control is used to convey the rootpath from the SMA to the message sender. It may be used for network diagnosis and loop detection.

NOTE – This control cannot be used before an MA has joined the RMCP-2 tree as it will not yet have a rootpath.

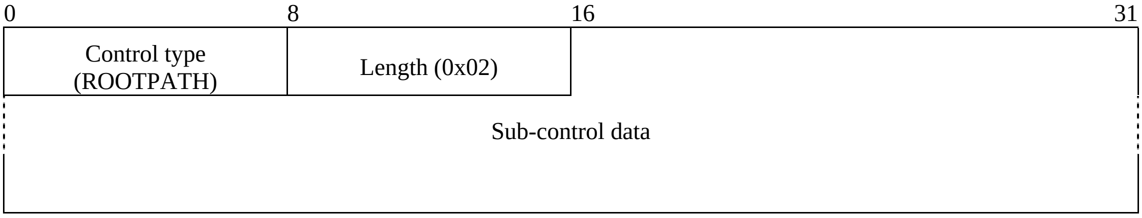


Figure 48 – ROOTPATH control

The format of the ROOTPATH control is shown in Figure 48. The description of each field is as follows:

a) *Control type* – This field denotes the ROOTPATH control. Its value shall be set to 0x07 (see Table 24).

b) *Length* – This field denotes the length (2 bytes) of the ROOTPATH control. Its value shall be set to 0x02.

c) *Sub-control data* – The RP_XXX sub-control that shall be used in the ROOTPATH control is shown in Table 6.

Table 6 – RP_XXX sub-control type for the ROOTPATH control

Sub-control type	Meaning	Status	Reference
RP_XXX	Specification of rootpath elements to be used.	Mandatory	See 7.3.3.5.1

7.3.3.5.1 RP_XXX sub-control

Figure 49 shows the general format of the RP_XXX sub-control preceded by a ROOTPATH control. RP_XXX stands for one of the ROOTPATH types listed in Table 26 (see the Note in the table). This RP_XXX sub-control represents different combinations of fields for MAIDs, bandwidth and delay. If the RP_XXX sub-control indicates that any of the MAIDs, bandwidth or delay fields are not needed, these fields shall not be present in the RP_XXX sub-control. The length of the rootpath element, in bytes, for each of the RP_XXX sub-control is indicated in Table 26.

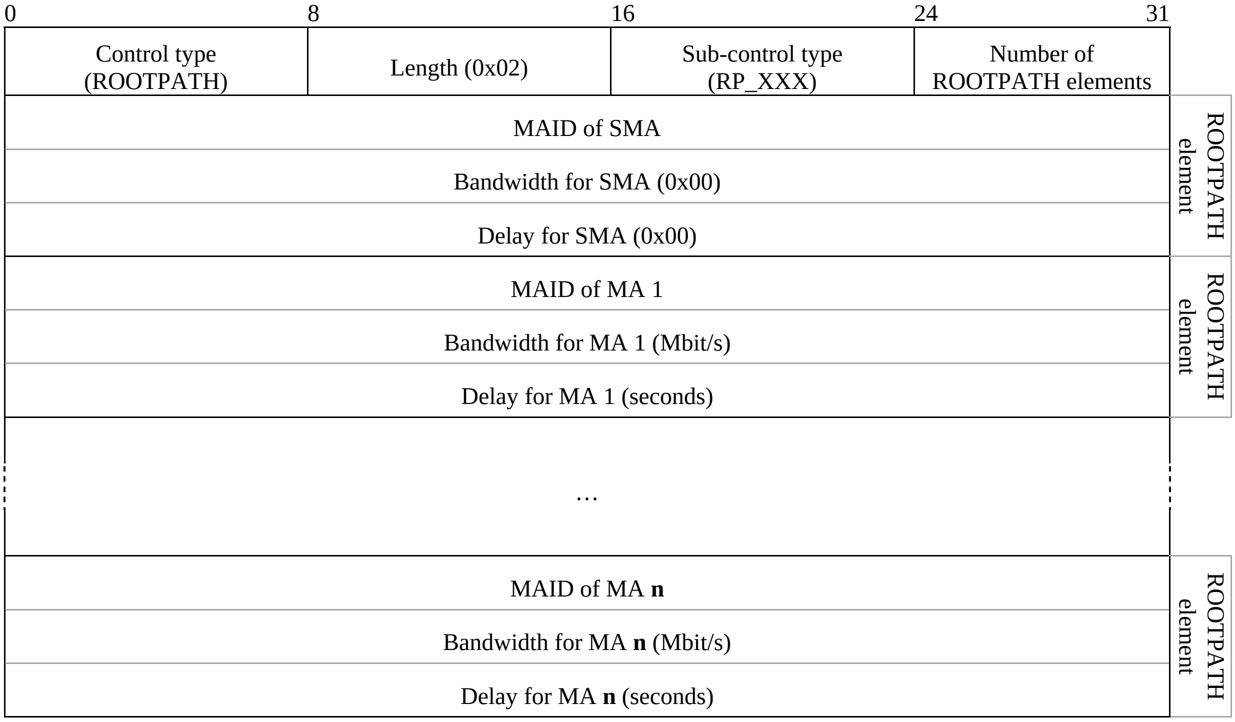


Figure 49 – General format for RP_XXX sub-control

The format of an RP_XXX sub-control preceded by a ROOTPATH control is shown in Figure 49. The description of each field of the RP_XXX sub-control is as follows:

a) *Sub-control type* – This field denotes the RP_XXX sub-control. Its value shall be set to one of the code values in Table 26.

b) *Number of ROOTPATH elements* – This field shall be set to the number of ROOTPATH elements in the RP_XXX sub-control.

c) *MAID* – This field shall be set to that of the MAID corresponding to that element, if present. This field is for each element in the rootpath, listed in order from the SMA.

d) *Bandwidth* – This field shall be set to the bandwidth, in Mbit/s, between the MA and its parent, as perceived by the MA for each element in the rootpath, listed in order from the SMA, if present. In the case of the SMA element, the value for the bandwidth shall be set to zero.

e) *Delay* – This field shall be set to the delay in seconds from the SMA as perceived by the MA for each element in the rootpath, listed in order from the SMA, if present. In the case of the SMA element, the value for the bandwidth shall be set to zero.

NOTE – The values for the perceived bandwidth and delay for the SMA elements are set to zero as the ROOTPATH is assumed to start at the SMA.

7.3.3.6 SYSINFO control

The SYSINFO control in the PPROBREQ message is used to convey system information about the MA in its sub-controls.

The content and format of the SYSINFO control are specified in 7.3.1.3 and Figure 41. The SYSINFO sub-controls that may be used in the PPROBREQ message, together with their status and reference to their content and specification, are listed in Table 7.

Table 7 – SYSINFO sub-control types for the PPROBREQ and PPROBANS messages

Sub-control type	Meaning	Status	Reference
SI_UPTIME	The elapsed time in seconds since the node joined the RMCP-2 session.	Optional	See 7.3.11.4.1
SI_DELAY	The delay in seconds from the SMA, as perceived by the MA.	Optional	See 7.3.11.4.2
SI_ROOM_CMA	The number of CMA places that an MA has allocated and the total number that it is able to support.	Optional	See 7.3.11.4.3
SI_PROV_BW	The maximum incoming and outgoing bandwidths in Mbit/s of the network interface card.	Optional	See 7.3.11.4.4
SI_POSS_BW	The possible forwarding bandwidth that the MA can afford.	Optional	See 7.3.11.4.5
SI_SND_BW	The total bandwidth in Mbit/s consumed by the MA to serve its CMAs.	Optional	See 7.3.11.4.6
SI_SND_PACKET	The total number of packets sent by the MA from start-up.	Optional	See 7.3.11.4.7
SI_SND_BYTES	The total number of bytes sent by the MA from start-up.	Optional	See 7.3.11.4.8
SI_RCV_BW	The bandwidth in Mbit/s perceived by the MA.	Optional	See 7.3.11.4.9
SI_RCV_PACKET	The number of packets received by the MA from start-up.	Optional	See 7.3.11.4.10
SI_RCV_BYTES	The number of bytes received by the MA from start-up.	Optional	See 7.3.11.4.11
SI_TREE_CONN	A list of PMA and CMAs directly attached to the sending MA.	Optional	See 7.3.11.4.12
SI_TREE_MEM	A set of MAs defined by the use of a TREEEXPLOR control.	Optional	See 7.3.11.4.13

7.3.3.7 DATAPROFILE control

The DATAPROFILE control in the PPROBREQ message contains data profile proposed by probing MA.

The content and format of the DATAPROFILE control are specified in 7.3.1.4, Figures 42 and 87.

7.3.4 PPROBANS message

7.3.4.1 General

The PPROBANS message provides a response to the PPROBREQ message in the map discovery procedure and confirms that the probed MA is still active. It contains information about the network condition, and a list of its neighbour information.

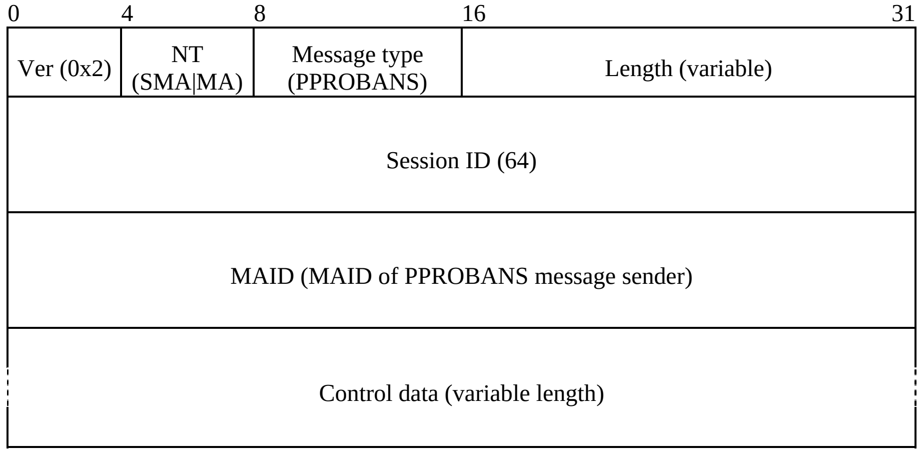


Figure 50 – PPROBANS message

7.3.4.2 PPROBANS message format

The format of the PPROBANS message is shown in Figure 50. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the PPROBANS message. Its value shall be set to 0x04 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of PPROBANS message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the PPROBANS message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the PPROBANS message, and their status, are shown in Table 8.

Table 8 – Control types for the PPROBANS message

Control type	Meaning	Status	Reference
TIMESTAMP	A measure of transmission time between sending and receiving MAs.	Mandatory	See 7.3.4.3
NEIGHBORLIST	A list of MAs for performing the map discovery.	Mandatory	See 7.3.4.4
ROOTPATH	A description of the path from the SMA.	Mandatory	See 7.3.4.5
SYSINFO	A description of the system information of the MA.	Mandatory	See 7.3.4.6
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.4.7

7.3.4.3 TIMESTAMP control

This TIMESTAMP control is used to measure transmission time between the sending MA and the receiving MA.

The content and format of the TIMESTAMP control are specified in 7.3.3.3 and Figure 47.

7.3.4.4 NEIGHBORLIST control

The NEIGHBORLIST control in a PPROBANS message is used to convey neighbour list information held by the probed MA.

The content and format of the NEIGHBORLIST control are specified in 7.3.2.5 and Figure 45.

7.3.4.5 ROOTPATH control

The ROOTPATH control is used to describe the path from the SMA to the message sender.

The content and format of the ROOTPATH control are specified in 7.3.3.5, 7.3.3.5.1 and in Figures 48 and 49.

7.3.4.6 SYSINFO control

The SYSINFO control in the PPROBANS message is used to convey system information about the probed MA for use in the map discovery procedure in its sub-controls.

The content and format of the SYSINFO control are specified in 7.3.1.3 and Figure 41. The SYSINFO sub-controls that may be used in PPROBANS message, together with their status and reference to their content and specification, are listed in Table 7.

7.3.4.7 DATAPROFILE control

The DATAPROFILE control in the PPROBANS message indicates whether the probed MA can afford the data profile proposed by the probing MA.

The content and format of the DATAPROFILE control are specified in 7.3.1.4, Figures 42 and 87.

7.3.5 HSOLICIT message

7.3.5.1 General

The HSOLICIT message is used to find the HMA inside its local network.

<table>
<tr><th>0</th><th>4</th><th>8</th><th>16 … 31</th></tr>
<tr><td>Ver
(0x2)</td><td>NT
(MA)</td><td>Message type
(HSOLICIT)</td><td>Length (0x14)</td></tr>
<tr><td colspan="4">Session ID (64)</td></tr>
<tr><td colspan="4">MAID (MAID of HSOLICIT message sender)</td></tr>
</table>

Figure 51 – HSOLICIT message

7.3.5.2 HSOLICIT message format

The format of the HSOLICIT message is shown in Figure 51. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA in Table 22.

c) *Message type* – This field denotes the HSOLICIT message. Its value shall be set to 0x05 (see Table 23).

d) *Length* – This field shall be set to the total length (20 bytes) of the HSOLICIT message. Its value shall be set to 0x14.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the HSOLICIT message sender. Its value shall be formatted as defined in 7.1.2.

NOTE – There is no control data associated with the HSOLICIT message.

7.3.6 HANNOUNCE message

7.3.6.1 General

The HANNOUNCE message is sent by the HMA as a reply to an HSOLICIT message, in order to announce the HMA's existence in a local network.

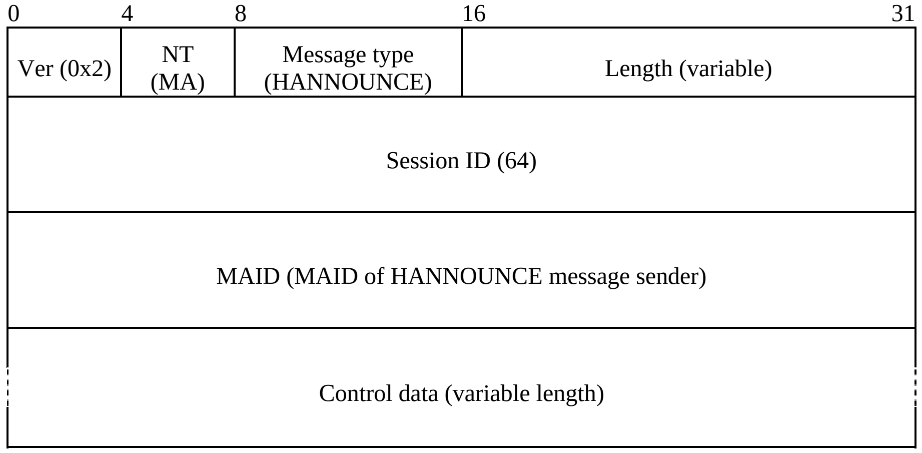


Figure 52 – HANNOUNCE message

7.3.6.2 HANNOUNCE message format

The format of the HANNOUNCE message is shown in Figure 52. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA in Table 22.

c) *Message type* – This field denotes the HANNOUNCE message. Its value shall be set to 0x06 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of HANNOUNCE message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the HANNOUNCE message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the HANNOUNCE message, and their status, are shown in Table 9.

Table 9 – Control types for the HANNOUNCE message

Control type	Meaning	Status	Reference
SYSINFO	A description of the system information of MA.	Optional	See 7.3.6.3
NEIGHBORLIST	A list of MAs for performing the map discovery.	Optional	See 7.3.6.4

7.3.6.3 SYSINFO control

The SYSINFO control in the HANNOUNCE message is used to convey system information about the HMA to the non-HMAs in the same multicast area in its SYSINFO sub-controls.

The content and format of the SYSINFO control are specified in 7.3.1.3 and Figure 41. The SYSINFO sub-controls that may be used in the HANNOUNCE message, together with their status and reference to their content and specification, are listed in Table 10.

Table 10 – SYSINFO sub-control types for the HANNOUNCE message

Sub-control type	Meaning	Status	Reference
SI_UPTIME	The elapsed time in seconds since the node joined the RMCP-2 session.	Optional	See 7.3.11.4.1
SI_DELAY	The delay in seconds from the SMA, as perceived by the MA.	Optional	See 7.3.11.4.2
SI_ROOM_CMA	The number of CMA places that an MA has allocated and the total number that it is able to support.	Optional	See 7.3.11.4.3
SI_PROV_BW	The maximum incoming and outgoing bandwidths in Mbit/s of the network interface card.	Optional	See 7.3.11.4.4
SI_POSS_BW	The possible forwarding bandwidth that the MA can afford.	Optional	See 7.3.11.4.5
SI_SND_BW	The total bandwidth in Mbit/s consumed by the MA to serve its CMAs.	Optional	See 7.3.11.4.6
SI_SND_PACKET	The total number of packets sent by the MA from start-up.	Optional	See 7.3.11.4.7
SI_SND_BYTES	The total number of bytes sent by the MA from start-up.	Optional	See 7.3.11.4.8
SI_RCV_BW	The bandwidth in Mbit/s perceived by the MA.	Optional	See 7.3.11.4.9
SI_RCV_PACKET	The number of packets received by the MA from start-up.	Optional	See 7.3.11.4.10
SI_RCV_BYTES	The number of bytes received by the MA from start-up.	Optional	See 7.3.11.4.11
SI_TREE_CONN	A list of PMA and CMAs directly attached to the sending MA.	Optional	See 7.3.11.4.12
SI_TREE_MEM	A set of MAs defined by the use of a TREEEXPLOR control.	Optional	See 7.3.11.4.13

7.3.6.4 NEIGHBORLIST control

The NEIGHBORLIST control in a HANNOUNCE message is used by an HMA to convey neighbour list information to non-HMAs in the same multicast area.

The content and format of the NEIGHBORLIST control are specified in 7.3.2.5 and Figure 45.

7.3.7 HLEAVE message

7.3.7.1 General

The HLEAVE message is sent by HMA to announce it is leaving from the RMCP-2 session to its local network.

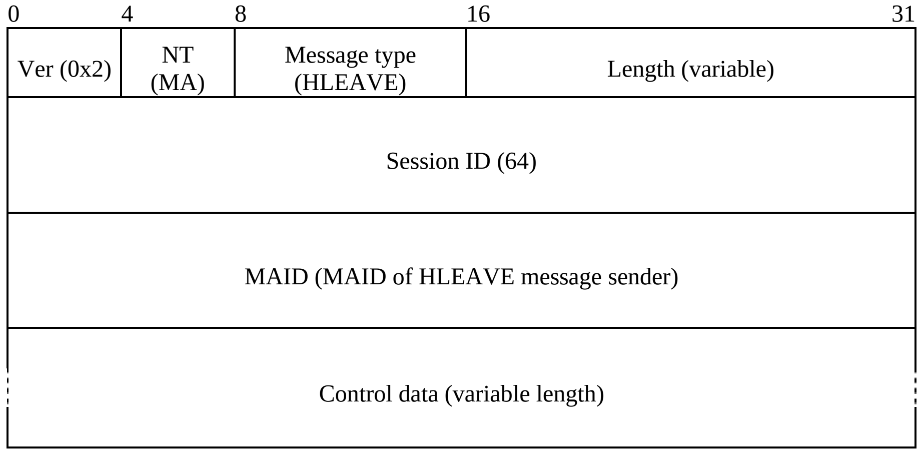


Figure 53 – HLEAVE message

7.3.7.2 HLEAVE message format

The format of the HLEAVE message is shown in Figure 53. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA in Table 22.

c) *Message type* – This field denotes the HLEAVE message. Its value shall be set to 0x07 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of the HLEAVE message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the HLEAVE message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the HLEAVE message, and their status, are shown in Table 11.

Table 11 – Control types for the HLEAVE message

Control type	Meaning	Status	Reference
CANDIDATEHMA	A set of candidate HMAs provided by the leaving HMA.	See conditions 1 and 2	See 7.3.7.3
NEIGHBORLIST	A list of MAs for performing the map discovery.	Optional	See 7.3.7.4
ROOTPATH	A description of the path from the SMA.	Mandatory	See 7.3.7.5
REASON	The reason for leaving the RMCP-2 session.	Mandatory	See 7.3.7.6
Condition 1: If the CANDIDATEHMA control is present, the competition to become the replacement HMA shall be restricted to the MAs in the list of MAIDs (see 7.3.7.3). Condition 2: If the CANDIDATEHMA control is absent, the competition to become the replacement HMA shall be open to all of the MAs in the same multicast enabled area.			

7.3.7.3 CANDIDATEHMA control

When an HMA leaves a session, every non-HMA in the multicast-enabled area may compete to become an HMA. This can cause the multicast-enabled area to be flooded with the HANNOUNCE message. To prevent HMA selection collision, CANDIDATEHMA control in an HLEAVE message is used to convey a restricted list of candidate HMAs that are invited, selected by the leaving HMA, to compete to become the replacement HMA.

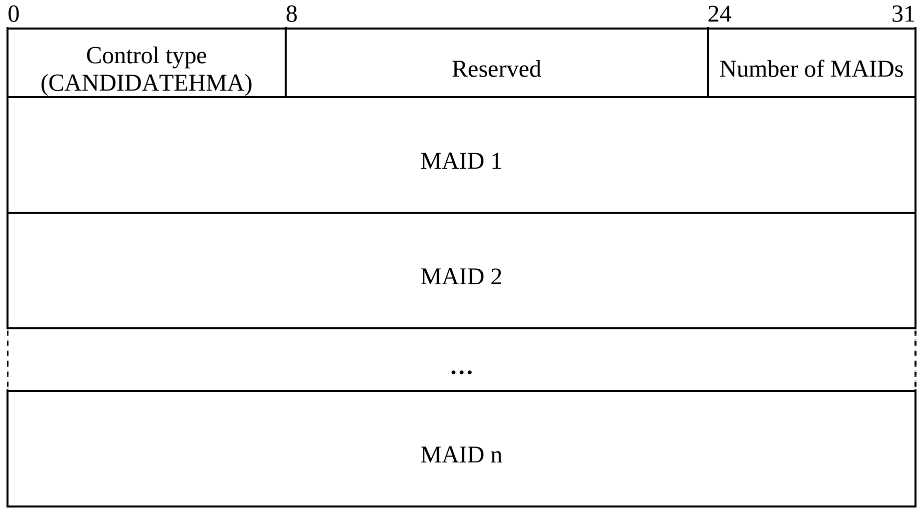


Figure 54 – CANDIDATEHMA control

The format of the CANDIDATEHMA control is shown in Figure 54. The description of each field is as follows:

a) *Control type* – This field denotes the CANDIDATEHMA control. Its value shall be set to 0x0A (see Table 24).

b) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

c) *Number of MAIDs* – This field shall be set to the number of MAIDs listed in CANDIDATEHMA control.

d) *MAID(s)* – These fields shall be set to the MAIDs of candidate HMAs selected by the leaving HMA.

7.3.7.4 NEIGHBORLIST control

The NEIGHBORLIST control in a HLEAVE message is used by an HMA to convey neighbour list information to non-HMAs in the same multicast area.

The content and format of the NEIGHBORLIST control are specified in 7.3.2.5 and Figure 45.

7.3.7.5 ROOTPATH control

The ROOTPATH control is used to describe the path from the SMA to the leaving HMA so that the newly selected HMA can follow the same root path.

The content and format of the ROOTPATH control are specified in 7.3.3.5, 7.3.3.5.1 and in Figures 48 and 49.

7.3.7.6 REASON control

The REASON control in an HLEAVE message is used to convey the HMA's reason for leaving the session.

0	8	16 … 31
Control type (REASON)	Length (0x04)	Reason code

Figure 55 – REASON control

The format of the REASON control is shown in Figure 55. The description of each field is as follows:

a) *Control type* – This field denotes the REASON control. Its value shall be set to 0x05 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the REASON control. Its value shall be set to 0x04.

c) *Reason code* – This field denotes the reason for leaving. Its value shall be set to 0x10 00, leave initiated by MA (see Table 29).

7.3.8 RELREQ message

7.3.8.1 General

The RELREQ message is used by a CMA to request its PMA to forward data. It usually includes a data profile which may be negotiated through the message exchanges of RELREQ and RELANS messages. It is also used to periodically notify the PMA of its presence.

0	4	8	16 … 31
Ver (0x2)	NT (MA)	Message type (RELREQ)	Length (variable)
Session ID (64)			
MAID (MAID of RELREQ message sender)			
Control data (variable length)			

Figure 56 – RELREQ message

7.3.8.2 RELREQ message format

The format of the RELREQ message is shown in Figure 56. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA in Table 22.

c) *Message type* – This field denotes the RELREQ message. Its value shall be set to 0x08 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of the RELREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the RELREQ message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the RELREQ message, and their status, are shown in Table 12.

Table 12 – Control types for the RELREQ message

Control type	Meaning	Status	Reference
RP_COMMAND	A request for rootpath information.	Optional	See 7.3.8.3
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.8.4
TIMESTAMP	A measure of transmission time between sending and receiving MAs.	Mandatory	See 7.3.8.5

7.3.8.3 RP_COMMAND control

The RP_COMMAND control in the RELREQ message is used by a CMA to request rootpath information from its PMA. For example, whenever a MA connects to PMA during joining or parent switching procedure, it requires the rootpath information including MAID of its new PMA for network diagnosis and loop detection.

0	8	16	31
Control type (RP_COMMAND)	Length (0x04)	RP_Command code	

Figure 57 – RP_COMMAND control

The format of the RP_COMMAND control is shown in Figure 57. The description of each field is as follows:

a) *Control type* – This field denotes the RP_COMMAND control. Its value shall be set to 0x01 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the RP_COMMAND control. Its value shall be set to 0x04.

c) *RP_Command code* – This field denotes the components to be returned in the ROOTPATH control of the RELANS message. Its value shall be set to one of the code values in Table 26.

7.3.8.4 DATAPROFILE control

The DATAPROFILE control is used to describe the proposed data profile of the sender of the RELREQ message.

The content and format of the DATAPROFILE control are specified in 7.3.1.4, Figures 42 and 87.

7.3.8.5 TIMESTAMP control

The TIMESTAMP control is used to measure transmission time between the sending MA and the receiving MA.

The content and format of the TIMESTAMP control are specified in 7.3.3.3 and Figure 47.

7.3.9 RELANS message

7.3.9.1 General

The RELANS message is issued by a PMA to notify whether a relay request in a RELREQ message from its CMA has been allowed. It may also contain additional information which is necessary to negotiate the data channel between the CMA and itself. It is also used to confirm that the answering MA is still active.

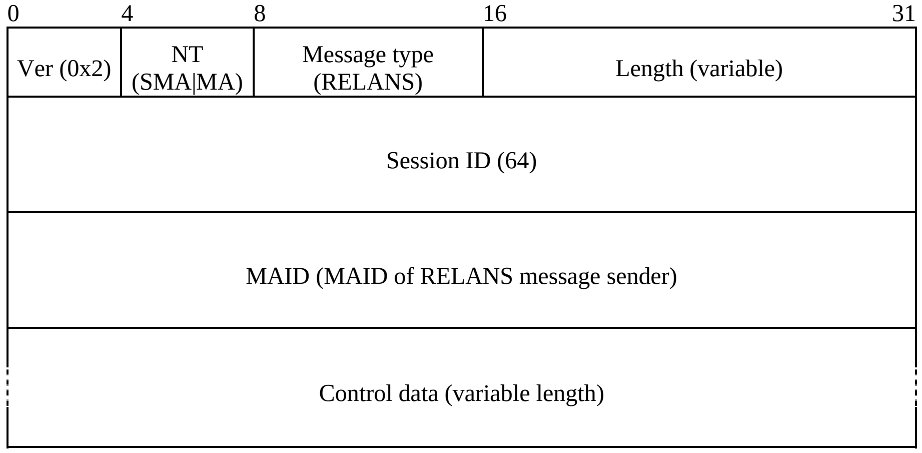


Figure 58 – RELANS message

7.3.9.2 RELANS message format

The format of the RELANS message is shown in Figure 58. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the RELANS message. Its value shall be set to 0x09 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of RELANS message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the RELANS message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the RELANS message, and their status, are shown in Table 13.

Table 13 – Control types for the RELANS message

Control type	Meaning	Status	Reference
RESULT	A result of the relay request.	Mandatory	See 7.3.9.3
DATAPROFILE	A description of the requirements for forwarding data.	Optional	See 7.3.9.4
TIMESTAMP	A measure of transmission time sending and receiving MAs.	Mandatory	See 7.3.9.5
ROOTPATH	A description of the path from the SMA.	See condition 1	See 7.3.9.6
Condition 1: The ROOTPATH control shall be included, if requested, in a RELREQ message.			

7.3.9.3 RESULT control

The RESULT control in a RELANS message is used by a PMA to convey whether or not its CMA's relay request is successful. If successful, it returns OK result code. If not, it returns an appropriate error code.

The format of RESULT control is shown in Figure 44. The description of each field is as follows:

a) *Control type* – This field denotes the RESULT control. Its value shall be set to 0x06 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the RESULT control. Its value shall be set to 0x04.

c) *Result code* – This field denotes the result of the request. Its value shall be set to one of the result codes listed in Table 25.

7.3.9.4 DATAPROFILE control

The DATAPROFILE control is used to describe the data profile confirmed by the sender of the RELANS message.

The content and format of the DATAPROFILE control are specified in 7.3.1.4, Figures 42 and 87.

7.3.9.5 TIMESTAMP control

The TIMESTAMP control is used to measure transmission time between the sending MA and the receiving MA.

The content and format of the TIMESTAMP control are specified in 7.3.3.3 and Figure 47.

7.3.9.6 ROOTPATH control

The ROOTPATH control in the RELANS message is used to describe the path from the SMA to the message sender in response to the RP_COMMAND in the RELREQ message from its CMA.

The content and format of the ROOTPATH control are specified in 7.3.3.5, 7.3.3.5.1 and in Figures 48 and 49.

7.3.10 STREQ message

7.3.10.1 General

The STREQ message is used by SM to request for system information from a single MA.

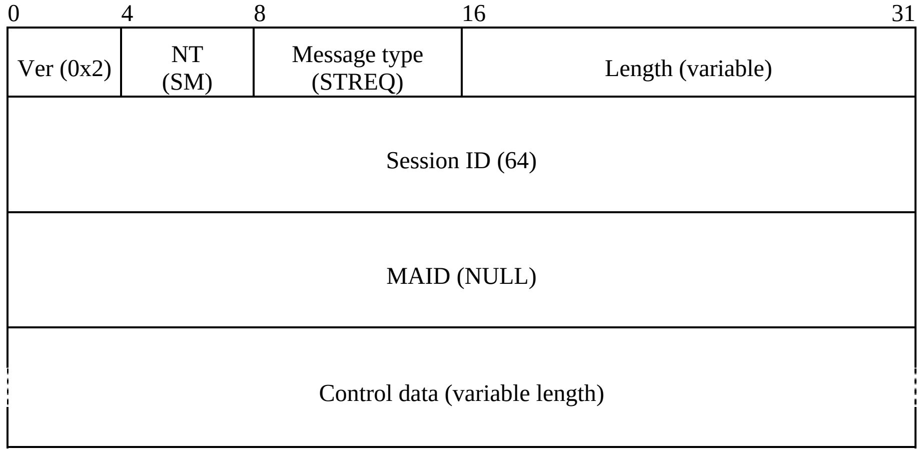


Figure 59 – STREQ message

7.3.10.2 STREQ message format

The format of the STREQ message is shown in Figure 59. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for SM in Table 22.

c) *Message type* – This field denotes the STREQ message. Its value shall be set to 0x0A (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of STREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to zero because the SM does not have a MAID.

g) *Control data* – The control types that may be used in the STREQ message, and their status, are shown in Table 14.

Table 14 – Control types for the STREQ message

Control type	Meaning	Status	Reference
SI_COMMAND	A request for specific information from an MA.	Mandatory	See 7.3.10.3
TREEEXPLOR	Specification limiting the scope of the tree.	See conditions 1 and 2	See 7.3.10.4
Condition 1: If the TREEEXPLOR control is absent, the STREQ message requests system information related only to the recipient of the STREQ message. Condition 2: If the TREEEXPLOR control is present, the STREQ message requests system information related to the set of MAs specified in 7.3.10.4.d).			

7.3.10.3 SI_COMMAND control

The SI_COMMAND control in a STREQ message is used by the SM to specify the specific information that is required from the recipient MA.

0	8	16	31
Control type (SI_COMMAND)	Length (0x04)	SI_Command code	

Figure 60 – SI_COMMAND control

The format of the SI_COMMAND control is shown in Figure 60. The description of each field is as follows:

a) *Control type* – This field denotes the SI_COMMAND control. Its value shall be set to 0x02 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the SI_COMMAND control. Its value shall be set to 0x04.

c) *SI_Command code* –This field shall be set to the arithmetic total of the command codes in Table 28 corresponding to the combination of SYSINFO sub-controls for which an answer is required (see 8.3.6).

7.3.10.4 TREEEXPLOR control

The TREEEXPLOR control is used to limit inspection size of a tree.

0	8	16	24	31
Control type (TREEEXPLOR)	Length (0x04)	Reserved	Tree depth	

Figure 61 – TREEEXPLOR control

The format of the TREEEXPLOR control is shown in Figure 61. The description of each field is as follows:

a) *Control type* – This field denotes the TREEEXPLOR control. Its value shall be set to 0x0B (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the TREEEXPLOR control. Its value shall be set to 0x04.

c) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

d) *Tree depth* – This field shall be set to the value to specify the scope of tree inspection. A tree depth of *n* defines the set of MAs consisting of the selected MA (or the SMA) that receives the STREQ message, all of its CMAs and all of its descendents on the RMCP-2 tree within *n* hops of the selected MA (or the SMA).

7.3.11 STANS message

7.3.11.1 General

The STANS message provides a response to the STREQ message.

0	4	8	16	31
Ver (0x2)	NT (SMA|MA)	Message type (STANS)	Length (variable)	
Session ID (64)				
MAID (MAID of STANS message sender)				
Control data (variable length)				

Figure 62 – STANS message

7.3.11.2 STANS message format

The format of the STANS message is shown in Figure 62. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the STANS message. Its value shall be set to 0x0B (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of STANS message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the STANS message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the STANS message, and their status, are shown in Table 15.

Table 15 – Control types for the STANS message

Control type	Meaning	Status	Reference
COLLECT	A header that identifies and delimits information related to individual MAs.	See condition 1	See 7.3.11.3
SYSINFO	A description of the system information of an MA, or a set of MAs.	Mandatory	See 7.3.11.4
Condition 1: The COLLECT control shall be included in the STANS message that forwards data collected in the STCOLANS message (see 7.3.13.1).			

7.3.11.3 COLLECT control

The COLLECT control is used in a STANS message only for reporting information collected in a STCOLANS message (see 7.3.13.1) in order to identify and delimit the SYSINFO control that pertains to a single MA. It shall be sequenced with SYSINFO control as defined in 7.3.13.3.

The content and format of the COLLECT control are specified in 7.3.13.4 and Figure 80.

7.3.11.4 SYSINFO control

The SYSINFO control in the STANS message is used by a MA to convey specific system information about itself in its SYSINFO sub-controls in response to an SI_COMMAND in a STREQ message.

The content and format of the SYSINFO control are specified in 7.3.1.3 and Figure 41.

The SYSINFO sub-controls that may be used in the STANS message, together with their status and reference to their content and specification, are listed in Table 16.

Table 16 – SYSINFO sub-control types for STANS and STCOLANS message

Sub-control type	Meaning	Status	Reference
SI_UPTIME	The elapsed time in seconds since the node joined the RMCP-2 session.	See condition 1	See 7.3.11.4.1
SI_DELAY	The delay in seconds from the SMA, as perceived by the MA.	See condition 1	See 7.3.11.4.2
SI_ROOM_CMA	The number of CMA places that an MA has allocated and the total number that it is able to support.	See condition 1	See 7.3.11.4.3
SI_PROV_BW	The maximum incoming and outgoing bandwidths in Mbit/s of the network interface card.	See condition 1	See 7.3.11.4.4
SI_POSS_BW	The possible forwarding bandwidth that the MA can afford.	See condition 1	See 7.3.11.4.5
SI_SND_BW	The total bandwidth in Mbit/s consumed by the MA to serve its CMAs.	See condition 1	See 7.3.11.4.6
SI_SND_PACKET	The total number of packets sent by the MA from start-up.	See condition 1	See 7.3.11.4.7
SI_SND_BYTES	The total number of bytes sent by the MA from start-up.	See condition 1	See 7.3.11.4.8
SI_RCV_BW	The bandwidth in Mbit/s perceived by the MA.	See condition 1	See 7.3.11.4.9

Table 16 – SYSINFO sub-control types for STANS and STCOLANS message

Sub-control type	Meaning	Status	Reference
SI_RCV_PACKET	The number of packets received by the MA from start-up.	See condition 1	See 7.3.11.4.10
SI_RCV_BYTES	The number of bytes received by the MA from start-up.	See condition 1	See 7.3.11.4.11
SI_TREE_CONN	A list of PMA and CMAs directly attached to the sending MA.	See condition 1	See 7.3.11.4.12
SI_TREE_MEM	A set of MAs defined by the use of a TREEEXPLOR control.	See condition 1	See 7.3.11.4.13
Condition 1: These sub-controls shall be included in the STANS message, if requested in a STREQ message.			

7.3.11.4.1 SI_UPTIME sub-control

The format of the SI_UPTIME sub-control preceded by a SYSINFO control is shown in Figure 63. The description of each field of the SI_UPTIME sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_UPTIME sub-control. Its value shall be set to 0x11 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_UPTIME sub-control. Its value shall be set to 0x06.

c) *Uptime* – This field shall be set to the elapsed time in seconds since the MA joined the RMCP-2 session.

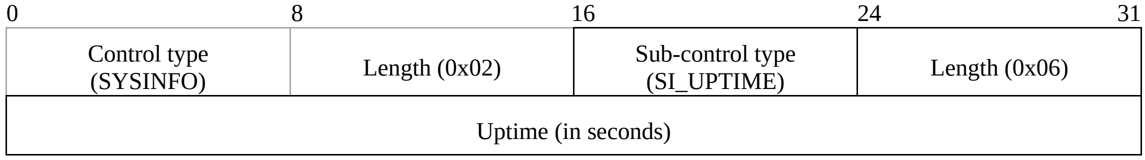


Figure 63 – SI_UPTIME sub-control

7.3.11.4.2 SI_DELAY sub-control

The format of the SI_DELAY sub-control preceded by a SYSINFO control is shown in Figure 64. The description of each field of the SI_DELAY sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_DELAY sub-control. Its value shall be set to 0x12 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_DELAY sub-control. Its value shall be set to 0x06.

c) *Delay* – This field shall be set to the delay in seconds from the SMA, as perceived by the MA.

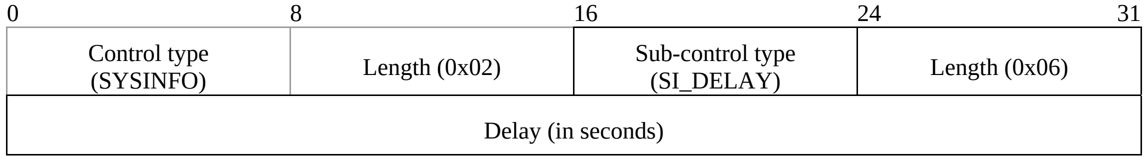


Figure 64 – SI_DELAY sub-control

7.3.11.4.3 SI_ROOM_CMA sub-control

The format of the SI_ROOM_CMA sub-control preceded by a SYSINFO control is shown in Figure 65. The description of each field of the SI_ROOM_CMA is as follows:

a) *Sub-control type* – This field denotes the SI_ROOM_CMA sub-control. Its value shall be set to 0x13 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_ROOM_CMA sub-control. Its value shall be set to 0x06.

c) *Number of CMAs allocated* – This field shall be set to number of CMA places that have been allocated by the MA. When the SI_ROOM_CMA sub-control is used in a SUBSREQ message this field shall be set to 0x0000.

d) *Total CMA capacity* – This field shall be set to the total number of CMA capacity that the MA is able to support.

NOTE – The available number of CMAs will be the difference between the total number of CMA capacity and the number of CMAs allocated.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_ROOM_CMA)	Length (0x06)
Number of CMAs allocated		Total CMA capacity	

Figure 65 – SI_ROOM_CMA sub-control

7.3.11.4.4 SI_PROV_BW sub-control

The format of the SI_PROV_BW sub-control preceded by a SYSINFO control is shown in Figure 66. The description of each field of the SI_PROV_BW is as follows:

a) *Sub-control type* – This field denotes the SI_PROV_BW sub-control. Its value shall be set to 0x15 (see Table 27).

b) *Length* – This field denotes the length of the SI_PROV_BW sub-control. Its value shall be set to 0x06.

c) *Incoming BW of NIC* – This field shall be set to the maximum incoming bandwidth in Mbit/s of the network interface card.

d) *Outgoing BW of NIC* – This field shall be set to the maximum outgoing bandwidth in Mbit/s of the network interface card.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_PROV_BW)	Length (0x06)
Incoming BW of NIC (in Mbit/s)		Outgoing BW of NIC (in Mbit/s)	

Figure 66 – SI_PROV_BW sub-control

7.3.11.4.5 SI_POSS_BW sub-control

The format of the SI_POSS_BW sub-control preceded by a SYSINFO control is shown in Figure 67. The description of each field of the SI_POSS_BW is as follows:

a) *Sub-control type* – This field denotes the SI_POSS_BW sub-control. Its value shall be set to 0x25 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_POSS_BW sub-control. Its value shall be set to 0x06.

c) *Forwarding bandwidth* – This field shall be set to the possible forwarding bandwidth in Mbit/s that the MA can support.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_POSS_BW)	Length (0x06)
Forwarding bandwidth (in Mbit/s)			

Figure 67 – SI_POSS_BW sub-control

7.3.11.4.6 SI_SND_BW sub-control

The format of the SI_SND_BW sub-control preceded by a SYSINFO control is shown in Figure 68. The description of each field of the SI_SND_BW sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_SND_BW sub-control. Its value shall be set to 0x35 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_SND_BW sub-control. Its value shall be set to 0x06.

c) *Bandwidth* – This field shall be set to the total bandwidth in Mbit/s consumed by the MA to serve its CMAs.

<table>
<tr><th>0</th><th>8</th><th>16</th><th>24 31</th></tr>
<tr><td>Control type
(SYSINFO)</td><td>Length (0x02)</td><td>Sub-control type
(SI_SND_BW)</td><td>Length (0x06)</td></tr>
<tr><td colspan="4">Bandwidth (in Mbit/s)</td></tr>
</table>

Figure 68 – SI_SND_BW sub-control

7.3.11.4.7 SI_SND_PACKET sub-control

The format of the SI_SND_PACKET sub-control preceded by a SYSINFO control is shown in Figure 69. The description of each field of the SI_SND_PACKET sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_SND_PACKET sub-control. Its value shall be set to 0x36 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_SND_PACKET sub-control. Its value shall be set to 0x06.

c) *Number of packets* – This field shall be set to the total number of packets sent by the MA from start-up.

<table>
<tr><th>0</th><th>8</th><th>16</th><th>24 31</th></tr>
<tr><td>Control type
(SYSINFO)</td><td>Length (0x02)</td><td>Sub-control type
(SI_SND_PACKET)</td><td>Length (0x06)</td></tr>
<tr><td colspan="4">Number of packets</td></tr>
</table>

Figure 69 – SI_SND_PACKET sub-control

7.3.11.4.8 SI_SND_BYTES sub-control

The format of the SI_SND_BYTES sub-control preceded by a SYSINFO control is shown in Figure 70. The description of each field of the SI_SND_BYTES sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_SND_BYTES sub-control. Its value shall be set to 0x37 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_SND_BYTES sub-control. Its value shall be set to 0x06.

c) *Number of bytes* – This field shall be set to the total number of bytes sent by the MA from start-up.

<table>
<tr><th>0</th><th>8</th><th>16</th><th>24 31</th></tr>
<tr><td>Control type
(SYSINFO)</td><td>Length (0x02)</td><td>Sub-control type
(SI_SND_BYTES)</td><td>Length (0x06)</td></tr>
<tr><td colspan="4">Number of bytes</td></tr>
</table>

Figure 70 – SI_SND_BYTES sub-control

7.3.11.4.9 SI_RCV_BW sub-control

The format of the SI_RCV_BW sub-control preceded by a SYSINFO control is shown in Figure 71. The description of each field of the SI_RCV_BW sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_RCV_BW sub-control. Its value shall be set to 0x45 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_RCV_BW sub-control. Its value shall be set to 0x06.

c) *Bandwidth* – This field shall be set to the bandwidth in Mbit/s perceived by the MA between itself and its PMA.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_RCV_BW)	Length (0x06)
Bandwidth (in Mbit/s)			

Figure 71 – SI_RCV_BW sub-control

7.3.11.4.10 SI_RCV_PACKET sub-control

The format of the SI_RCV_PACKET sub-control preceded by a SYSINFO control is shown in Figure 72. The description of each field of the SI_RCV_PACKET sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_RCV_PACKET sub-control. Its value shall be set to 0x46 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_RCV_PACKET sub-control. Its value shall be set to 0x06.

c) *Number of packets* – This field shall be set to the number of packets received by the MA from start-up.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_RCV_PACKET)	Length (0x06)
Number of packets			

Figure 72 – SI_RCV_PACKET sub-control

7.3.11.4.11 SI_RCV_BYTES sub-control

The format of the SI_RCV_BYTES sub-control preceded by a SYSINFO control is shown in Figure 73. The description of each field of the SI_RCV_BYTES sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_RCV_BYTES sub-control. Its value shall be set to 0x47 (see Table 27).

b) *Length* – This field denotes the length (6 bytes) of the SI_RCV_BYTES sub-control. Its value shall be set to 0x06.

c) *Number of bytes* – This field shall be set to the number of bytes received by the MA from start-up.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_RCV_BYTES)	Length (0x06)
Number of bytes			

Figure 73 – SI_RCV_BYTES sub-control

7.3.11.4.12 SI_TREE_CONN sub-control

The format of the SI_TREE_CONN sub-control preceded by a SYSINFO control is shown in Figure 74. The description of each field of the SI_TREE_CONN sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_TREE_CONN sub-control. Its value shall be set to 0x68 (see Table 27).

b) *Number of MAIDs* – This field shall be set to the number of MAIDs in the list including that of the PMA.

c) *MAID of PMA* – This field shall be set to the MAID of the PMA of the reporting MA.

d) *MAIDs of CMAs* – These fields shall be set to the MAIDs of the CMAs of the reporting MA.

NOTE – There is no significance in the ordering of MAIDs.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_TREE_CONN)	Number of MAIDs (**n** + 1)
MAID of PMA			
MAID of CMA **1**			
…			
MAID of CMA **n**			

Figure 74 – SI_TREE_CONN sub-control

7.3.11.4.13 SI_TREE_MEM sub-control

The format of the SI_TREE_MEM sub-control preceded by a SYSINFO control is shown in Figure 75. The description of each field of the SI_TREE_MEM sub-control is as follows:

a) *Sub-control type* – This field denotes the SI_TREE_MEM sub-control. Its value shall be set to 0x69 (see Table 27).

b) *Number of MAIDs* – This field shall be set to the number of MAIDs listed in the SI_TREE_MEM sub-control.

c) *MAIDs of member* – These fields shall be set to the MAIDs of the members of the sub-tree defined by a TREEEXPLOR control.

NOTE – There is no significance in the ordering of MAIDs.

0	8	16	24 31
Control type (SYSINFO)	Length (0x02)	Sub-control type (SI_TREE_MEM)	Number of MAIDs (**n**)
MAID of member **1**			
…			
MAID of member **n**			

Figure 75 – SI_TREE_MEM sub-control

7.3.12 STCOLREQ message

7.3.12.1 General

STCOLREQ and STCOLANS messages are used to collect system information from a set of MAs as requested in a STREQ message. The STREQ message is issued by the SM and is sent to a selected MA (or the SMA). If the STREQ message contains a TREEEXPLOR control, it defines the set of MAs as the selected MA (or the SMA) that receives the STREQ message, all of its CMAs and all of its descendents on the RMCP-2 tree within *n* hops of the selected MA. This information is collected from members of the set through STCOLREQ and STCOLANS messages via the RMCP-2 tree.

The STCOLREQ messages are relayed to all members of the set. The TREEEXPLOR control defines the members of the set and the SI_COMMAND control indicates the type of information to be returned. Figure 76 shows an example of such a set where PMA is the selected MA and *n* = 3.

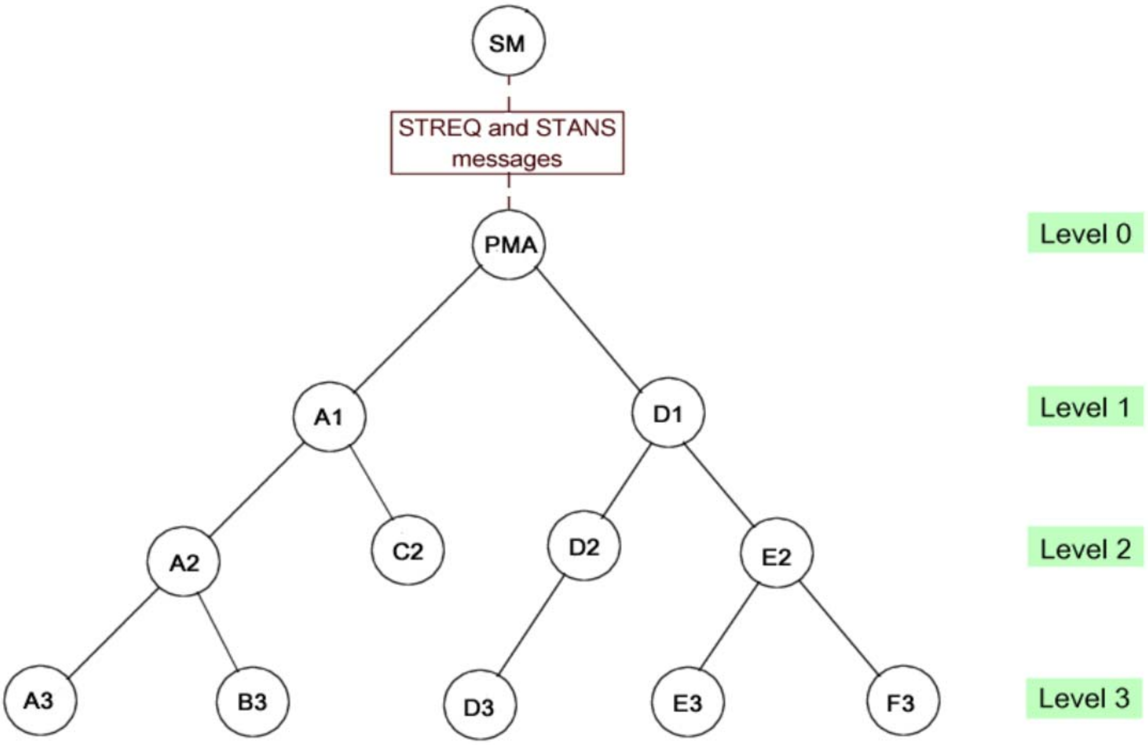


Figure 76 – Example of a delivery tree for STCOLREQ and STCOLANS messages

0	4	8	16 31
Ver (0x2)	NT (SMA|MA)	Message type (STCOLREQ)	Length (variable)
Session ID (64)			
MAID (MAID of STCOLREQ message sender)			
Control data (variable length)			

Figure 77 – STCOLREQ message

7.3.12.2 STCOLREQ message format

The format of the STCOLREQ message is shown in Figure 77. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the STCOLREQ message. Its value shall be set to 0x1A (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of STCOLREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the STCOLREQ message sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The control types that may be used in the STCOLREQ message, and their status, are shown in Table 17.

Table 17 – Control types for the STCOLREQ message

Control type	Meaning	Status	Reference
SI_COMMAND	A request for specific information from an MA.	Mandatory	See 7.3.12.3
TREEEXPLOR	Specification limiting the scope of the tree.	Optional	See 7.3.12.4

7.3.12.3 SI_COMMAND control

The SI_COMMAND control in a STCOLREQ message is used by a PMA to request specific system information from its CMAs.

The content and format of the SI_COMMAND control are specified in 7.3.10.3 and Figure 60. The SI_Command code value shall be identical to that in the STREQ message from the SM that initiates the collection of the information through the STCOLREQ and STCOLANS messages.

7.3.12.4 TREEEXPLOR control

The TREEEXPLOR control is used to limit the scope of the tree to be inspected.

The content and format of the TREEEXPLOR control are specified in 7.3.10.4 and in Figure 61. The tree depth value set by the MA that is the recipient of the STREQ message that initiates the STCOLREQ message shall be identical to that in the STREQ message. The value of the tree depth shall be decreased by one every time that the message is relayed to a lower level.

NOTE – When an MA receives an STCOLREQ message with tree depth = 0, it will know that it is at the end of the relaying chain and that it should start to return its information in a STCOLANS message to its PMA.

7.3.13 STCOLANS message

7.3.13.1 General

The STCOLANS message is used to relay system information from a defined subset of the RMCP-2 tree. This information is collected in a controlled manner. Firstly, the MAs in the lowest layer, each sends information related to their own node to their PMA. The PMA amalgamates this information and adds its own information before relaying the combined information to its parent in the next layer. The relaying continues until the head of the sub-tree has collected all the information and then forwards it in a STANS message to the SM.

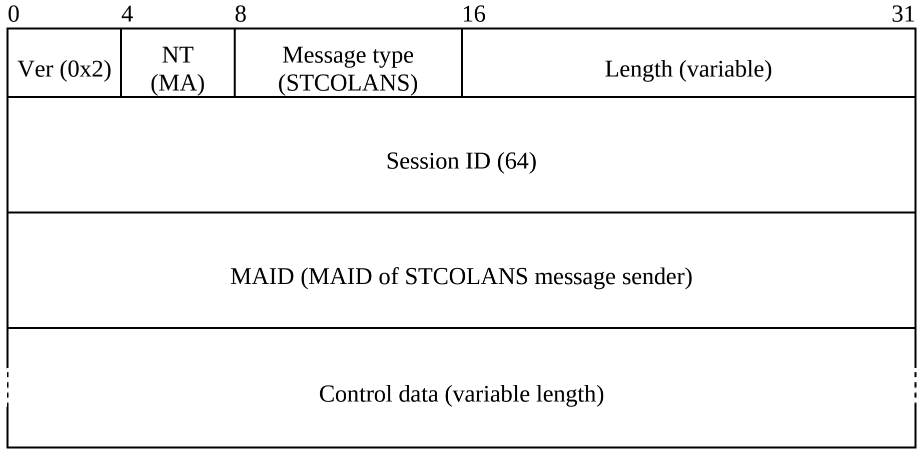


Figure 78 – STCOLANS message

7.3.13.2 STCOLANS message format

The format of the STCOLANS message is shown in Figure 78. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for MA as in Table 22.

c) *Message type* – This field denotes the STCOLANS message. Its value shall be set to 0x1B (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of STCOLANS message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the STCOLANS message sender. Its value shall be formatted as defined in 7.1.2

g) *Control data* – The control types that may be used in the STCOLANS message, and their status, are shown in Table 18.

Table 18 – Control types for the STCOLANS message

Control type	Meaning	Status	Reference
COLLECT	A header that identifies and delimits information related to individual MAs.	Optional	See 7.3.13.4
SYSINFO	A description of the system information of MA.	Mandatory	See 7.3.13.5

7.3.13.3 Sequence of controls in STCOLANS message

A STCOLANS message contains system information for a set of MAs at a given level in the RMCP-2 tree. The set is defined as a PMA and its CMAs and descendents to a depth defined by the TREEEXPLOR control of the original STREQ message from the SM requesting this information.

Figure 79 shows the sequence of controls in a STCOLANS message. Each COLLECT control identifies a particular MA and the subsequent SYSINFO sub-controls contain the type of system information defined by the SI_COMMAND control of the original STREQ message from the SM requesting this information.

COLLECT control identifying MA1
SYSINFO control and sub-control A containing information relating to MA 1
SYSINFO control and sub-control B containing information relating to MA 1
SYSINFO control and sub-control C containing information relating to MA 1
COLLECT control identifying MA2
SYSINFO control and sub-control A containing information relating to MA 2
SYSINFO control and sub-control B containing information relating to MA 2
SYSINFO control and sub-control C containing information relating to MA 2
. . Fields relating to other MAs . .

Figure 79 – Sequence of COLLECT controls and SYSINFO controls in a STCOLANS message

7.3.13.4 COLLECT control

0	8	16	24 31
Control type (COLLECT)	Length (0x0C)	Reserved	Number of sub-controls
MAID (MAID of MA to which the following status reports apply)			

Figure 80 – COLLECT control

The format of the COLLECT control is shown in Figure 80. The description of each field of the COLLECT control is as follows:

a) *Control type* – This field denotes the COLLECT control. Its value shall be set to 0x0C (see Table 24).

b) *Length* – This field denotes the length (12 bytes) of the COLLECT control. Its value shall be set to 0x0C.

c) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

d) *Number of sub-controls* – This field shall be set to the number of sub-controls associated with the MA indentified in the MAID in e).

e) *MAID* – This field shall be set to the MAID of the MA to which the status reports in the following SYSINFO sub-controls apply (see Figure 79).

7.3.13.5 SYSINFO control

The content and format of the SYSINFO control are specified in 7.3.1.3 and Figure 41. The SYSINFO sub-controls in an STCOLANS message shall correspond to those indicated by the SI_COMMAND in the STCOLREQ message that triggered the message. A list of valid SYSINFO sub-controls that may be used in the STCOLANS message, together with their status and reference to their content and specification, are listed in Table 16.

7.3.14 LEAVREQ message

7.3.14.1 General

The LEAVREQ message is used

a) by an MA to inform its PMA and CMAs of its leaving;

b) by the SM or a PMA to expel an MA from the session; or

c) by an MA when it changes its PMA.

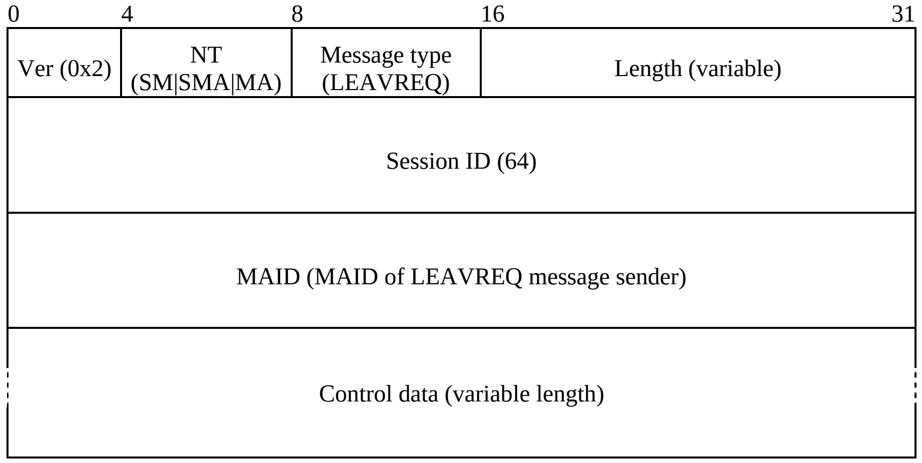


Figure 81 – LEAVREQ message

7.3.14.2 LEAVREQ message format

The format of the LEAVREQ message is shown in Figure 81. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SM, SMA, or MA coded as in Table 22.

c) *Message type* – This field denotes the LEAVREQ message. Its value shall be set to 0x0C (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of LEAVREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the LEAVREQ message sender and its value shall be formatted as defined in 7.1.2. If the message is sent by SM, this field shall be set to zero.

g) *Control data* – The control type that shall be used in the LEAVREQ message, and its status, is shown in Table 19.

Table 19 – Control type for the LEAVREQ message

Control type	Meaning	Status	Reference
REASON	The reason for leaving of MA.	Mandatory	See 7.3.14.3

7.3.14.3 REASON control

The REASON control in an LEAVREQ message is used to convey the MA's reason for leaving the session. The REASON control format is shown in Figure 55. The description of each field is as follows:

a) *Control type* – This field denotes the REASON control. Its value shall be set to 0x05 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the REASON control. Its value shall be set to 0x04.

c) *Reason code* – This field denotes the reason for leaving. Its value shall be set to one of the code values in Table 29.

7.3.15 LEAVANS message

7.3.15.1 General

The LEAVANS message is used to acknowledge the receipt of a LEAVREQ message.

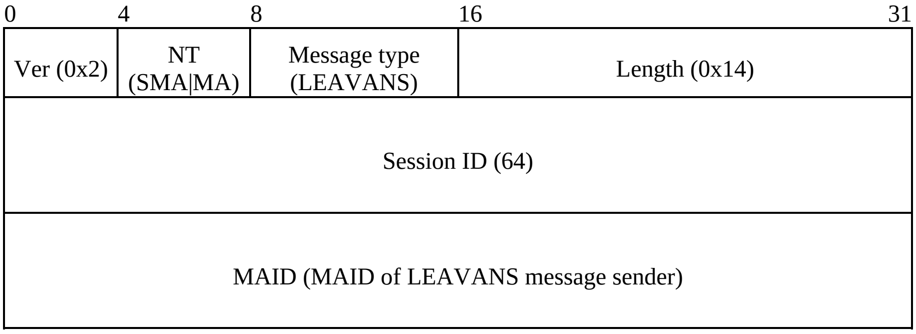


Figure 82 – LEAVANS message

7.3.15.2 LEAVANS message format

The format of the LEAVANS message is shown in Figure 82. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the LEAVANS message. Its value shall be set to 0x0D (see Table 23).

d) *Length* – This field shall be set to the total length (20 bytes) of LEAVANS message. Its value shall be set to 0x14.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the LEAVANS message sender. Its value shall be formatted as defined in 7.1.2.

NOTE – There is no control data associated with the LEAVANS message.

7.3.16 HB message

7.3.16.1 General

The HB message is issued periodically by the SMA to examine the condition of the RMCP-2 tree and to create the rootpath information for receiving MAs. This information enables each MA to diagnose the network condition.

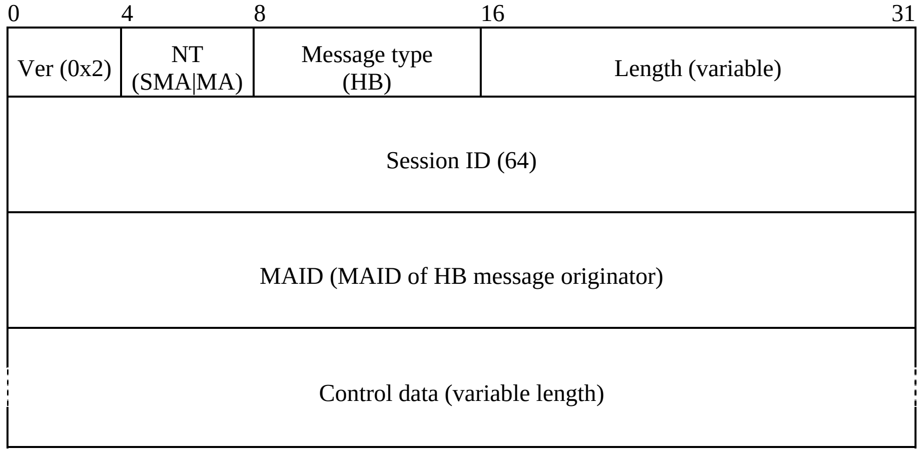


Figure 83 – HB message

7.3.16.2 HB message format

The format of the HB message is shown in Figure 83. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message originator's node type. For a regular HB message, its value shall be set to SMA coded as in Table 22. For a pseudo-HB message, its value shall be set to that of MA coded as in Table 22. This field shall not be changed as the HB message is relayed down the RMCP-2 tree.

c) *Message type* – This field denotes the HB message. Its value shall be set to 0x10 (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of HB message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the HB message originator. Its value shall be formatted as defined in 7.1.2. This field shall not be changed as the HB message is relayed down the RMCP-2 tree.

g) *Control data* – The control types that may be used in the HB message, and their status, are shown in Table 20.

Table 20 – Control types for the HB message

Control type	Meaning	Status	Reference
ROOTPATH	A description of the path from the SMA.	See condition 1	See 7.3.16.3
PSEUDO_HB	An indication that the message is a pseudo-HB message for network partitioning detection and recovery.	See condition 2	See 7.3.16.4
Condition 1: The ROOTPATH control shall always be included in a regular HB message. Condition 2: The PSEUDO_HB control shall always be included in a pseudo-HB message.			

7.3.16.3 ROOTPATH control

The ROOTPATH control in the HB message is used to describe the path from the SMA to the message sender.

The content and format of the ROOTPATH control are specified in 7.3.3.5, 7.3.3.5.1 and in Figures 48 and 49.

7.3.16.4 PSEUDO_HB control

When a PMA tries to recover from network partition, its descendants may start network fault recovery procedure due to HB expectation timeout. A single point of partitioning may cause a fault recovery chain effect.

To avoid this, the MA generates PSEUDO_HB control in order to notify its descendants of a network fault and to delay its descendants' fault recovery procedure.

0	8	16 31
Control type (PSEUDO_HB)	Length (0x04)	Reserved

Figure 84 – PSEUDO_HB control

The format of the PSEUDO_HB control for a pseudo-HB message is shown in Figure 84. The description of each field is as follows:

a) *Control type* – This field denotes the PSEUDO_HB control. Its value shall be set to 0x0D (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the PSEUDO_HB control for the HB message. Its value shall be set to 0x04.

c) *Reserved* – This field is reserved for future use. Its value shall be set to zero. It is ignored by the receiver.

7.3.17 TERMREQ message

7.3.17.1 General

TERMREQ message is used to terminate the existing RMCP-2 session. The SM sends the TERMREQ message to the SMA and the SMA relays the message to members of the RMCP-2 tree.

0	4	8	16 31
Ver (0x2)	NT (SM|SMA|MA)	Message type (TERMREQ)	Length (variable)
Session ID (64)			
MAID (MAID of TERMREQ message sender)			
Control data (variable length)			

Figure 85 – TERMREQ message

7.3.17.2 TERMREQ message format

The format of the TERMREQ message is shown in Figure 85. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SM, SMA, or MA coded as in Table 22.

c) *Message type* – This field denotes the TERMREQ message. Its value shall be set to 0x0E (see Table 23).

d) *Length* – This field shall be set to the total length in bytes of TERMREQ message including control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the TERMREQ message sender and its value shall be formatted as defined in 7.1.2. If the message is sent by the SM, this field shall be set to zero.

g) *Control data* – The control type that shall be used in the TERMREQ message, and its status, is shown in Table 21.

Table 21 – Control type for the TERMREQ message

Control type	Meaning	Status	Reference
REASON	The reason for termination of the session.	Mandatory	See 7.3.17.3

7.3.17.3 REASON control

The REASON control in the TERMREQ message is used to convey the reason for termination of the session.

The REASON control format is shown in Figure 55. The description of each field is as follows:

a) *Control type* – This field denotes the REASON control. Its value shall be set to 0x05 (see Table 24).

b) *Length* – This field denotes the length (4 bytes) of the REASON control. Its value shall be set to 0x04.

c) *Reason code* – This field denotes the reason for termination of the session. Its value shall be set by the SM to one of the code values in Table 30. When the TERMREQ message is forwarded subsequently by an MA, the value set by the SM shall remain unchanged.

7.3.18 TERMANS message

7.3.18.1 General

The TERMANS message is used to acknowledge the receipt of a TERMREQ message.

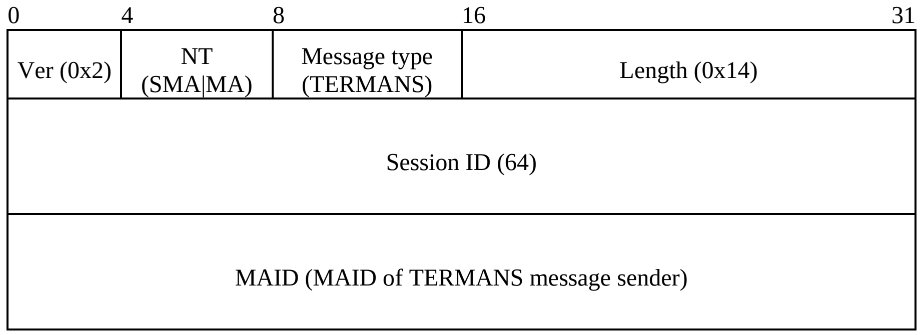


Figure 86 – TERMANS message

7.3.18.2 TERMANS message format

The format of the TERMANS message is shown in Figure 86. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x2.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA or MA coded as in Table 22.

c) *Message type* – This field denotes the TERMANS message. Its value shall be set to 0x0F (see Table 23).

d) *Length* – This field shall be set to the total length (20 bytes) of the TERMANS message. Its value shall be set to 0x14.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field shall be set to the MAID of the TERMANS message sender. Its value shall be formatted as defined in 7.1.2.

NOTE – There is no control data associated with the TERMANS message.

8 Parameters

This clause explains the parameter values of RMCP-2 tree management. RMCP-2 defines the data forwarding profile as a means of specifying the data channel information in terms of data types. In addition, some of the control parameters are used for efficient and optimized management of the control tree.

8.1 Data forwarding profile

RMCP-2 defines the data forwarding profile as a profile that describes the requirements for forwarding data between a PMA and the PMA's direct CMA. The data forwarding 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 described for the negotiation.

In Figure 87, a data forwarding profile is illustrated as SDP-style text format.

```
Stream1: Protocol = UDP, Listen address = a.b.c.d:9898, Encapsulation = IP-IP
Stream2: Protocol = UDP, Listen address = a.b.c.d:9899, Encapsulation = UDP
Stream3: Protocol = TCP, Llisten address = a.b.c.d:9899, Encapsulation = TCP, CurrentSeq=xxxx,
BufferedSeq = yyyy, CurrentRcvdSeq = xxxx-1
```

Figure 87 – An example of the data forwarding profile

8.2 Parameters used in RMCP-2

RMCP-2 defines some parameters to manage the control tree. These parameters control the time information of the RMCP-2 session or define the number of messages or provide other information.

8.2.1 Parameters for session initialization

Each MA that wants to join an RMCP-2 session should contact the SM to fetch the bootstrapping information for the session. The SM gives an NL as the bootstrapping information. Because of resource limitations, the NL cannot keep all the MAs of the session. Hence, the following parameter is used to limit the size of the NL:

a) *N_StartNL*: It defines the number of MAs listed in the Neighbour List. It can be changed by the SM before the session starts as well as after the session started. The default value for *N_StartNL* is 100.

8.2.2 Parameters for map discovery

Every MA in the RMCP-2 performs a map discovery procedure by periodically exchanging PPROBREQ messages and PPROBANS messages with neighbouring MAs. The following parameters are related to the map discovery procedure:

a) *PPROB.time*: It defines the period of issuing PPROBREQ message. Each MA sends PPROBREQ messages at every *PPROB.time*. The default value for *PPROB.time* is 45 seconds but it can be changed arbitrarily.

b) *N_MAX_PROBE*: This parameter limits the maximum number of PPROBREQ messages that can be sent by each MA simultaneously to prevent PPROBREQ implosion. The default value for *N_MAX_PROBE* is 1, but it can be changed arbitrarily.

8.2.3 Parameters for session maintenance

This subclause includes the mechanism for the session heartbeat as well as the session maintenance.

RMCP-2 uses the HB for tree maintenance. The HB synchronizes the whole session along the data delivery path. Within the synchronized session, each MA can switch its parent to improve the RMCP-2 session. The HB also detects any network faults, such as loops and partitions. RMCP-2 defines the parameters for heartbeat as follows:

a) *HB.time*: It is the period of HB message. The SMA of a session sends a HB message at every HB.time. The default value for *HB.time* is 15 seconds.

b) *MAX_PARTITION_CNT*: It is used to examine the tree is partitioned. If a MA does not receive HB message for MAX_PARTITION_CNT * HB.time, then it can detect tree has partitioned. The default value for *MAX_PARTITION_CNT* is 3.

8.2.4 Parameters for HMA selection

RMCP-2 enables an IP multicast data transmission in a multicast-enabled area. The following parameters support this functionality:

a) *H_SOLICIT.time*: It defines the time period of HSOLICIT message. An MA in the multicast-enabled area sends H_SOLICIT message at every H_SOLICIT.time. The default value for *H_SOLICIT.time* is 2 seconds.

b) *N_SOLICIT*: It is the maximum trial number of HSOLICIT message generation as non-HMA. After N_SOLICIT times of HSOLICIT message issue, the MA tries to become new HMA in the multicast-enabled area. The default value for *N_SOLICIT* is 3.

c) *H_ANNOUNCE.time*: It defines the period of HANNOUNCE message. HMA sends an H_ANNOUNCE message at every H_ANNOUNCE.time. The default value for *H_ ANNOUNCE.time* is 6 seconds.

d) *N_ANNOUNCE*: It is the maximum trial number of HANNOUNCE message generation as HMA. If no HSOLICIT message appears, HMA stops forwarding data into the multicast-enabled area. The default value for *N_ ANNOUNCE* is set to 3.

8.2.5 Parameters used during data delivery

To connect and continue the data relay, each CMA periodically sends a RELREQ message to its PMA. The following parameters are used to support data relay procedure:

a) *RELREQ.time*: It is the maximal period to generate the RELREQ message. Both PMA and CMA have the same size of RELREQ.time. The initial value of *RELREQ.time* is 6 seconds.

b) *N_ RELREQ*: It is used to examine whether the CMA is still alive or not. If a PMA does not receive RELREQ message for RELREQ.time * N_RELREQ, then the PMA considers that its CMA has left the session abruptly. The default value for *N_ RELREQ* is 3.

8.2.6 Parameters for session leave

RMCP-2 allows MA's early session leave. When an MA in the middle of a tree is due to leave a session, the MA should wait for a certain period for soft tree reconfiguration. The following parameters are used to support session leave procedure:

a) *LEAVE.time*: It is the duration managed by leaving PMA to make its CMA enable to find new PMAs and attach to them. The default value for *LEAVE.time* is 10 seconds.

8.3 Code values used in RMCP-2

8.3.1 Codes values for basic RMCP-2 node types

Table 22 lists the node types (NT) for the basic RMCP-2 protocol and their corresponding 4-bit code values.

NOTE – The code value for the MA node type applies only to the basic RMCP-2 protocol defined in clauses 5-7. The secure RMCP-2 protocol does not use the code value for MAs: it has its own code values for DMA and RMA node types.

Table 22 – Node type code values for basic RMCP-2

Node type	Code value (4 bits)
SM	0x1
SMA	0x2
MA	0x4

8.3.2 Code values for RMCP-2 message types

Table 23 lists the RMCP-2 message types and their corresponding code values.

Table 23 – Code values for RMCP-2 message types

Message type	Code value (Hexadecimal)
SUBSREQ	0x01
SUBSANS	0x02
PPROBREQ	0x03
PPROBANS	0x04
HSOLICIT	0x05
HANNOUNCE	0x06
HLEAVE	0x07
RELREQ	0x08
RELANS	0x09
STREQ	0x0A
STANS	0x0B
STCOLREQ	0x1A
STCOLANS	0x1B
LEAVREQ	0x0C

Table 23 – Code values for RMCP-2 message types

Message type	Code value (Hexadecimal)
LEAVANS	0x0D
HB	0x10
TERMREQ	0x0E
TERMANS	0x0F

8.3.3 Code values for RMCP-2 control types

Table 24 lists the RMCP-2 control types and their corresponding code values

Table 24 – Code values for RMCP-2 control types

Control type	Value (Hexadecimal)
RP_COMMAND	0x01
SI_COMMAND	0x02
DATAPROFILE	0x03
NEIGHBORLIST	0x04
REASON	0x05
RESULT	0x06
ROOTPATH	0x07
SYSINFO	0x08
TIMESTAMP	0x09
CANDIDATEHMA	0x0A
TREEEXPLOR	0x0B
COLLECT	0x0C
PSEUDO_HB	0x0D

8.3.4 RMCP-2 return value

Table 25 lists the encoded values and meaning of the RESULT codes.

Table 25 – Result code

Result Code	Meaning
0x10 00	OK
0x20 00	System Problem
0x30 00	Administrative Problem

8.3.5 Code values related to the ROOTPATH control

Table 26 lists the code values for the sub-controls of the ROOTPATH control. The length in bytes of each rootpath element is indicated for each ROOTPATH type.

Table 26 – Sub-control type codes for ROOTPATH control and the RP_COMMAND control

Sub-control type	Code (8 bits)	Command code (16 bits)	Meaning	Length of rootpath element (in bytes)
RP_ID	0x11	0x00 01	The ROOTPATH control contains only the MAID for each node.	8
RP_BW	0x12	0x00 02	The ROOTPATH control contains only the bandwidth in Mbit/s as perceived by the MA for each node.	4
RP_DL	0x14	0x00 04	The ROOTPATH control contains only the delay in seconds from the SMA as perceived by the MA for each node.	4
RP_ID_BW	0x13	0x00 03	The ROOTPATH control contains the MAID and bandwidth in Mbit/s as perceived by the MA for each node.	12
RP_ID_DL	0x15	0x00 05	The ROOTPATH control contains the MAID and the delay in seconds from the SMA as perceived by the MA for each node.	12
RP_ID_BW_DL	0x17	0x00 07	The ROOTPATH control contains the MAID, bandwidth in Mbit/s and the delay in seconds as perceived by the MA for each node.	16
NOTE – The code values for RP_ID_BW, RP_ID_DL and RP_ID_BW_DL sub-controls are calculated by 0x10 plus the arithmetic sums of last four bits of the individual codes of the RP_ID, RP_BW and RP_DL components.				

8.3.6 Code values related to SYSINFO control

Table 27 lists the sub-control types, their code values, and meanings.

Table 27 – Sub-control types for SYSINFO control

Type	Code (8 bits)	Meaning
SI_UPTIME	0x11	The elapsed time in seconds since the node joined the RMCP-2 session.
SI_DELAY	0x12	The delay in seconds from the SMA, as perceived by the MA.
SI_ROOM_CMA	0x13	The number of CMA places that an MA has allocated and the total number that it is able to support.
SI_PROV_BW	0x15	The maximum incoming and outgoing bandwidths in Mbit/s of the network interface card.
SI_POSS_BW	0x25	The possible forwarding bandwidth that the MA can afford.
SI_SND_BW	0x35	The total bandwidth in Mbit/s consumed by the MA to serve its CMAs.
SI_SND_PACKET	0x36	The total number of packets sent by the MA from start-up.
SI_SND_BYTES	0x37	The total number of bytes sent by the MA from start-up.
SI_RCV_BW	0x45	The bandwidth in Mbit/s perceived by MA.
SI_RCV_PACKET	0x46	The number of packets received by the MA from start-up.
SI_RCV_BYTES	0x47	The number of bytes received by the MA from start-up.
SI_TREE_CONN	0x68	A list of PMA and CMAs directly attached to the sending MA.
SI_TREE_MEM	0x69	A set of MAs defined by the use of a TREEEXPLOR control.

Table 28 lists the command codes corresponding to the sub-controls for the SYSINFO control. Combinations of different sub-control may be indicated by adding together the corresponding individual SI_Command codes.

NOTE – The 16-bit format column in Table 28 demonstrates how the SI_Command code values may be added together to give unique combinations. The bit positions can be considered as representing individual sub-control types and the 1 or 0 values can be interpreted as presence or absence of these sub-control types. For example, 0000 0010 0100 0010 represents the combination of SI_DELAY, SI_SND_PACKET, and SI_RCV_PACKET sub-controls.

Table 28 – SI_Command codes for sub-control types for SYSINFO control

Sub-control type	Sub-control code	Command code	16-bit format
SI_UPTIME	0x11	0x00 01	0000 0000 0000 000**1**
SI_DELAY	0x12	0x00 02	0000 0000 0000 00**1**0
SI_ROOM_CMA	0x13	0x00 04	0000 0000 0000 0**1**00
SI_PROV_BW	0x15	0x00 08	0000 0000 0000 **1**000
SI_POSS_BW	0x25	0x00 10	0000 0000 000**1** 0000
SI_SND_BW	0x35	0x00 20	0000 0000 00**1**0 0000
SI_SND_PACKET	0x36	0x00 40	0000 0000 0**1**00 0000
SI_SND_BYTES	0x37	0x00 80	0000 000 **1**000 0000
SI_RCV_BW	0x45	0x01 00	0000 000**1** 0000 0000
SI_RCV_PACKET	0x46	0x02 00	0000 00**1**0 0000 0000
SI_RCV_BYTES	0x47	0x04 00	0000 0**1**00 0000 0000
SI_TREE_CONN	0x68	0x10 00	000**1** 0000 0000 0000
SI_TREE_MEM	0x69	0x20 00	00**1**0 0000 0000 0000

8.3.7 Code values related to the leave of the SMA or an MA

Table 29 lists the reason codes for leaving. The first four bits of the code specify the main cause of leaving, and the second four bits specify further details for leaving.

Table 29 – Leave reason code

Category	Code	Meaning
Leave	0x10 00	Leave initiated by MA
	0x11 00	Leave of SMA
Kick out	0x20 00	Expulsion by SM
	0x21 00	Expulsion by PMA
Parent switching	0x40 00	Parent switching by MA

8.3.8 Code values related to session termination

Table 30 lists the reason codes for the session termination. The first four bits of the code specify the main reason for the session termination, and the second four bits specify the detailed reason for session termination.

Table 30 – Termination reason code

Category	Code	Meaning
Normal session termination	0xE0 00	Session is terminated normally
Abnormal session termination	0xF0 00	Session is terminated abnormally without reason
	0xF1 00	Session is terminated abnormally for administrative reasons

9 Overview of secure RMCP-2

9.1 Conventions

9.1.1 Use of basic RMCP-2 protocol

The term basic RMCP-2 protocol, when used in clauses 9-12, refers to the protocol defined in clauses 5-8.

9.1.2 Hexadecimal notation

Code values for message parameters in clause 11 (Format of secure RMCP-2 messages) and clause 12 (Parameters) are expressed in hexadecimal notation, e.g., 0x14 for 20 in decimal notation.

9.2 Secure RMCP-2 entities

9.2.1 Introduction

The secure RMCP-2 protocol supports security functions of the RMCP-2 used for relayed multicast data transport through unicast communication over the Internet.

The secure RMCP-2 protocol components correspond to those described in the basic RMCP-2 protocol except that a new type of MA, a dedicated multicast agent (DMA), has been introduced. A dedicated multicast agent is an intermediate MA pre-deployed as a trust server by the SM. For secure communication, each session consists of an SM, an SMA, DMAs, RMAs, together with a single sending application and multiple receiving applications. Their topology, as shown in Figure 88, corresponds with that in the basic RMCP-2 protocol (see 5.1).

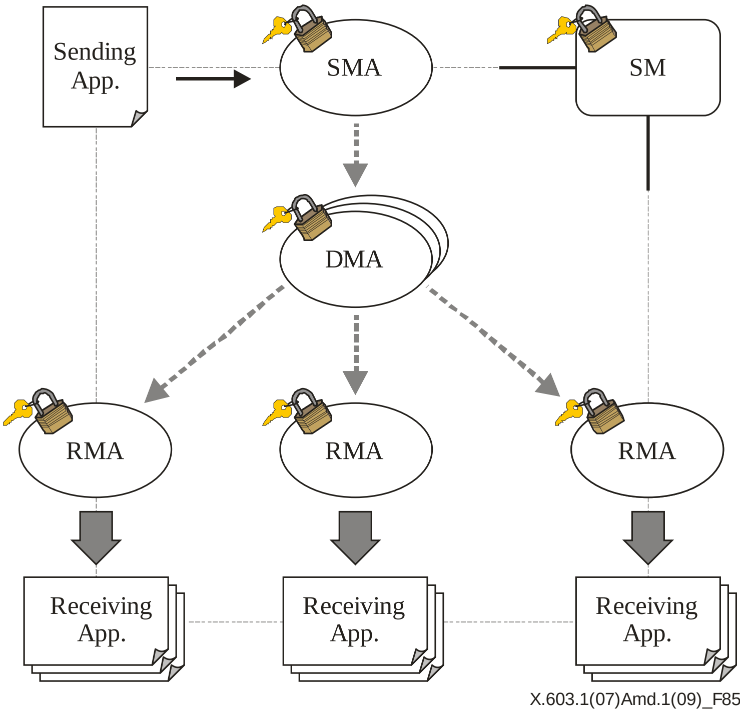


Figure 88 – RMCP-2 service topology with security

9.2.2 Session manager

The SM is responsible for maintaining session security, which includes the management of service membership, the management of key and ACL for DMA and RMA, and message encryption/decryption together with the SM functions of basic RMCP-2. Figure 89 shows an abstract protocol stack for the operation of SM functions. The SM has TLS and multicast session security modules for the provision of security. TLS is used for the initial authentication of DMAs and RMAs when they join the session. The Multicast session security module performs the following security functions after the completion of TLS authentication:

a) Security policy;

b) Session admission management;

c) Session key management;

d) Access Control list management;

e) Secure group and membership management;

f) Message encryption/decryption.

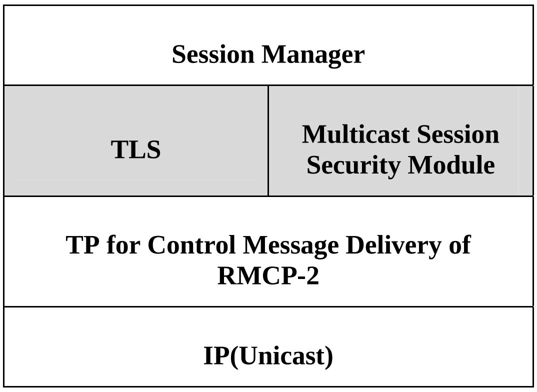

Session Manager
TLS
Multicast Session Security Module
TP for Control Message Delivery of RMCP-2
IP(Unicast)

Figure 89 – Internal structure of the SM

9.2.3 Dedicated multicast agents

DMAs are in charge of the secure establishment and maintenance of the RMCP-2 tree, support of membership authentication and data confidentiality. Figure 90 shows the internal structure of the DMAs with modules for Key/Message Security Management and Group/Member Security Management. These modules support the following security functions:

Key/Message Security Management Module

a) Group key management;

b) Message encryption/decryption;

c) Contents encryption key management.

Group/Member Security Management Module

a) Secure tree configuration;

b) Session key management;

c) Secure group and membership management.

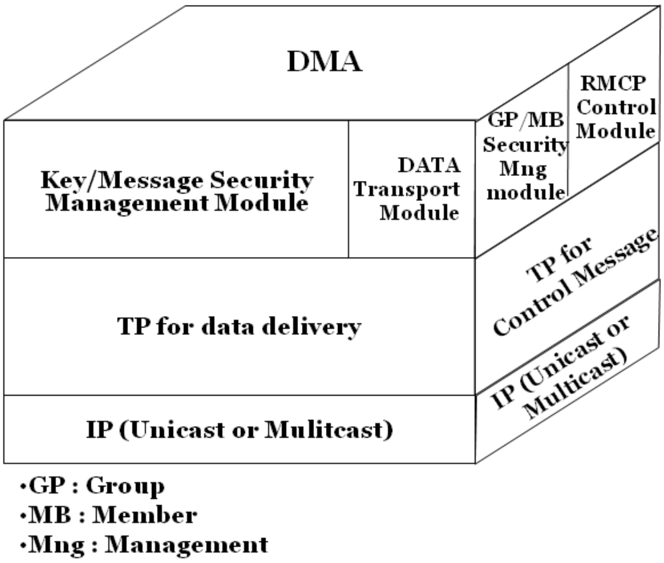


Figure 90 – Internal structure of DMAs

9.2.4 Sender and receiver multicast agents

The internal structure of the SMA and the RMAs is shown in Figure 91. The structure is the same as for DMAs except that the Group Security Management Module is not included.

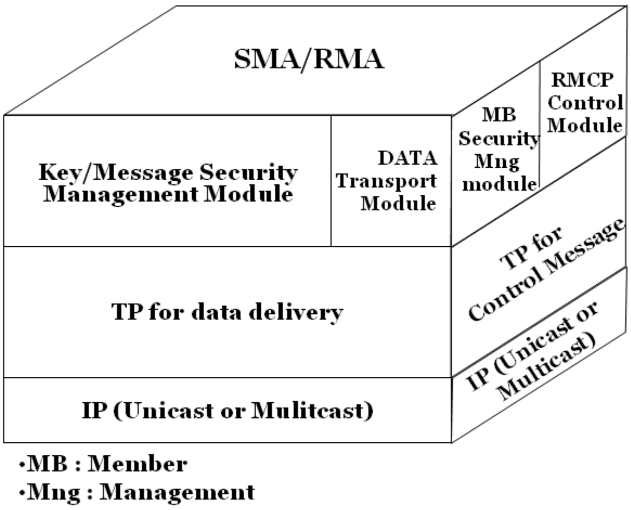


Figure 91 – Internal structure of the SMA and RMAs

9.3 Protocol blocks

The protocol blocks for the SM, Group/Member Security Management of MAs and Key/Message Security Management of MAs are shown in Figures 92, 93 and 94. They correspond to the protocol stacks in the basic RMCP-2 protocol in 5.2 (see Figures 2, 3 and 4) but also include the TLS protocol and the Multicast Session Security Module.

The secure RMCP-2 protocol supports general encryption/decryption algorithms of TLS for a variety of common applications. The SM and MAs (SMA, DMAs and RMAs) share the security information described in the security policy. The Multicast Session Security Module contains common symmetric encryption/decryption algorithms, authentication mechanisms, and multicast security modules related to RMCP-2 security functions.

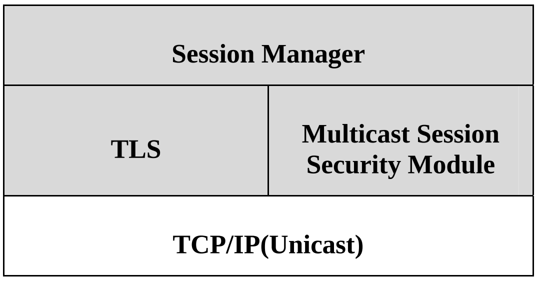


Figure 92 – Protocol block of the SM

The SM messages and the Group/Member Security Management messages of MAs are transmitted reliably through the TCP protocol.

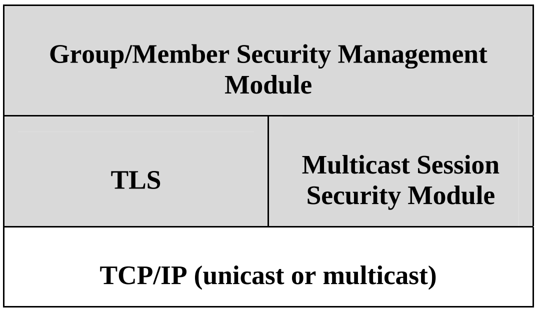

Group/Member Security Management Module
TLS
Multicast Session Security Module
TCP/IP (unicast or multicast)

Figure 93 – Protocol block for the group/member security management of MAs

Key/Message Security Management messages may be transferred using any transport protocol. The transport protocol may be selected according to the nature of the transferred data types. TLS provides secure communication for TCP over unicast communication. The Multicast Security Encryption/Decryption and Authentication Modules protect the multicast packets. These modules contain common symmetric encryption algorithms, hash algorithms, and multicast security modules defined in this Recommendation | International Standard to protect the multicast packets.

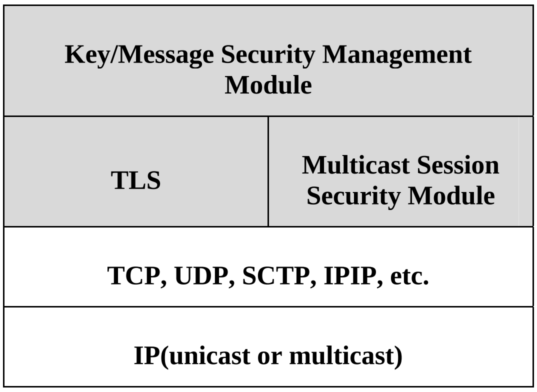

Key/Message Security Management Module
TLS
Multicast Session Security Module
TCP, UDP, SCTP, IPIP, etc.
IP(unicast or multicast)

Figure 94 – Protocol block for the key/message security management of MAs

9.4 Types of secure RMCP-2 protocol messages

Control messages are exchanged between secure RMCP-2 protocol nodes in a request-and-answer manner.

Table 31 shows the messages that are specific to the secure RMCP-2 protocol. They complement the messages listed in Table 1 (see 5.4).

Table 31 – Secure RMCP-2 messages

Messages	Meaning	Operations
SUBSREQ (control type= SERV_USER_IDENT)	Additional control type= SERV_USER_IDENT in SUBSREQ (Subscription request)	Session Initialization
RELREQ (control type=AUTH)	Additional control type=AUTH in RELREQ (Relay request)	Membership Authentication
RELANS (control type=AUTH_ANS)	Additional control type=AUTH_ANS in RELANS (Relay answer)	
SECAGREQ	Security Agreement request	Establishment of Multicast Security Policy
SECLIST	Security List	
SECALGREQ	Security Algorithms request	
SECAGANS	Security Agreement answer	
KEYDELIVER	Key Delivery	Key Distribution
HRSREQ	Head Required Security request	Group Member Authentication Group Key Distribution ACL Management
HRSANS	Head Required Security answer	

9.5 Structure of regional security management

For scalable security management, the secure RMCP-2 protocol supports security functions in two independent regions: a RM (Relayed Multicast) region and a MM (Member Multicast) region.

The RM region is a management zone of the session key (Ks). It consists of the SM, the SMA and DMAs in a multicast disabled area.

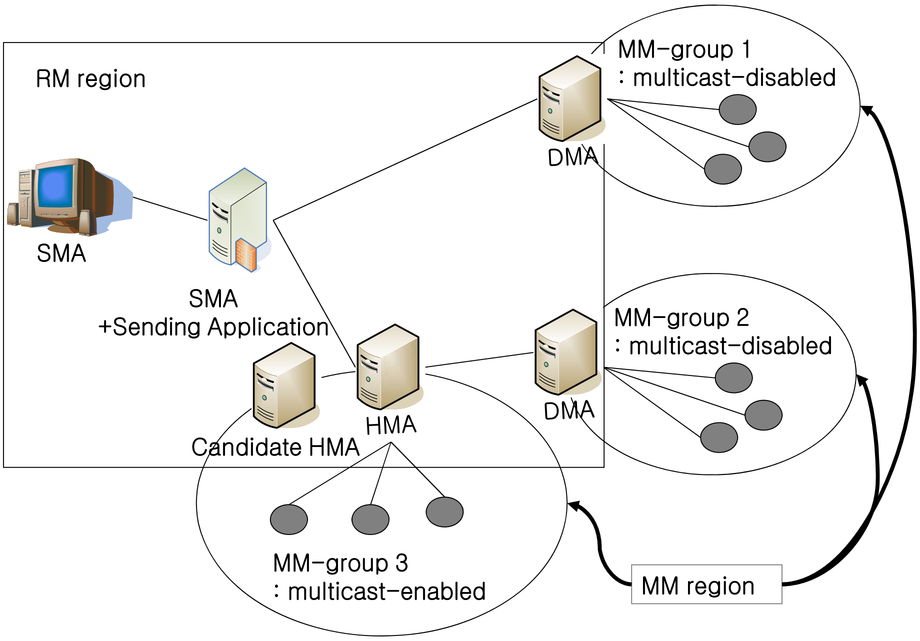


Figure 95 – Security management regions

The MM region is a management zone defined by the use of group keys (Kg). The MM region consists of DMAs and RMAs. They can be connected over a multicast-enabled or a multicast-disabled network. The MM region consists of one or more MM groups each using its own Kg group key.

Multicast-enabled MM groups consist of an HMA, one or more candidate HMAs and multiple RMAs that receive the same multicast messages. Candidate HMAs are DMAs that are not connected to the data delivery tree, but have the capability to assume the role of HMA if required. Multicast-disabled MM groups consist of one DMA and multiple RMAs. In both cases, the RMAs are logically connected direct to their parent DMA on the data delivery tree.

Any change in an MM group is localized within the scope of its own MM group.

10 Protocol operation

10.1 SM operation

The SM supports the establishment of security policies applied to each secure RMCP-2 session, and is responsible for user and MA security management such as user and MA authentication. It manages the session key for each RMCP-2 session through the creation, update, and distribution of key information. The SM also has message encryption and decryption abilities through the use of TLS and owned cryptography suites.

10.1.1 Admission control

10.1.1.1 TLS authentication

TLS authentication is performed in advance of the subscription requests of MAs (SMA, DMAs or RMAs). An MA establishes a TLS session with the SM according to IETF RFC 3546. The SM, as part of the IETF 3546 procedure, decides which TLS mode, TLS_CERT or TLS_PSK, is applied for the verification of the parties concerned. The SM responds to the MA and, if the mutual authentication is successful, shares a secret key K_{TLS} with the MA.

The SM also delivers the session key Ks, encrypted using K_{TLS}, to the SMA and the DMAs, but not to the RMAs.

The TLS session with the SMA and DMAs is closed after the session key is delivered, since the SM, SMA and DMAs exchange control messages that have been encrypted with the session key. The TLS session with RMAs is retained and not closed until membership authentication with their parent DMA in the secure tree join procedure (see 10.2.4) and the individual key K_{MAS} has been established.

10.1.1.2 Admission of the SMA

A secure RMCP-2 session is initiated through the subscription of the SMA. The SMA first obtains authorization for providing the contents from the SM. The SMA is authenticated by the SM through the TLS session (see 10.1.1.1) and then joins the session by exchanging SUBSREQ and SUBSANS messages with the SM. As a result of this, the SMA receives the session key Ks and is enabled to act as an administrative node of the secure RMCP-2 tree.

10.1.1.3 Admission of DMAs

The DMAs, as prospective trust parties, are invited by the SM to join the session and to establish the DMA network before the subscription of RMAs. The means of this invitation are outside the scope of this Recommendation | International Standard.

The DMAs are authenticated by the SM through the TLS session and they join the session through the exchange of SUBSREQ and SUBSANS messages with the SM. They receive the session key Ks from the SM and join the RMCP-2 tree through the secure tree join procedure (see 10.2.4).

The SM consults with the DMAs joining the session before the announcement of the opening of the secure RMCP-2 session, giving a date and time when the subscription of RMAs begins. The means of this announcement are outside the scope of this Recommendation | International Standard.

10.1.1.4 Admission of RMAs to open groups

A potential RMA will know from the announcement of the session whether or not the session supports open groups. The RMAs are authenticated by the SM through the TLS session and join the session through the exchange of SUBSREQ and SUBSANS messages with the SM. They do not receive the session key Ks. They join the RMCP-2 tree through the secure tree join procedure (see 10.2.4).

10.1.1.5 Admission of RMAs to closed groups

A potential RMA will know from the announcement of the session whether or not the session supports closed groups. Access to membership of closed groups is controlled by the content provider (CP). A potential RMA requests a service user identifier from the CP. The CP provides a service user identifier to the potential RMA and also sends the service user identifier, without revealing the identity of the potential RMA, to the SM. The CP is responsible for the format of this identifier and this is not defined in this Recommendation | International Standard.

When the session is opened to RMAs, the RMAs are authenticated by the SM through the TLS session and they join the session through the exchange of SUBSREQ and SUBSANS messages with the SM. The SUBSREQ message shall contain the service user identifier. The SM shall send a rejection in the RESULT control of the SUBSANS message if the SM does not hold an identical service user identifier.

The RMAs do not receive the session key Ks. They join the RMCP-2 tree through the secure tree join procedure (see 10.2.4).

10.1.2 Key management for which the SM is responsible

10.1.2.1 Session key

The session key (Ks) is shared between the SM, the SMA and DMAs and is used to encrypt/decrypt control messages in the RM region. It is initially created by the SM in the bootstrapping of the RMCP-2 session. Ks is encrypted by the individual key K_{TLS} (see 10.1.2.2) for delivery to the SMA and to each DMA through the data protection procedure of TLS following successful TLS authentication.

Ks is updated at regular intervals through the hash function. When a DMA is truncated or an abnormal situation occurs, the SM does not use the hash function, but instead creates a totally new session key Ks, without hashing. The new key is delivered to the SMA and all DMAs in the RMCP-2 session (see Figure 96).

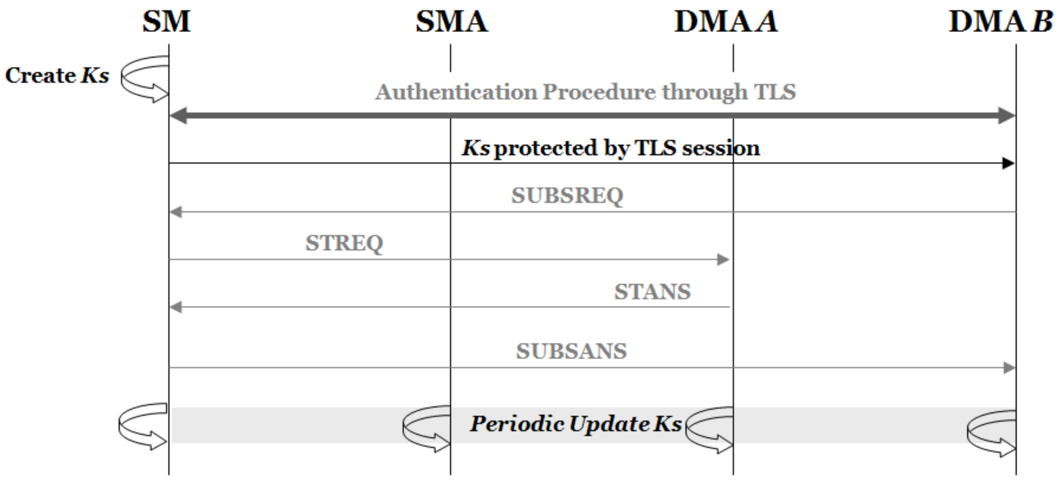


Figure 96 – Session key management

10.1.2.2 TLS key

The TLS key K_{TLS} is a private key generated through successful TLS authentication during admission control. Each MA (SMA, DMA and RMA) shares a different K_{TLS} with the SM, which is not shared with the other MAs. K_{TLS} is not updated during the lifetime of the RMCP-2 session.

10.1.3 Establishment of security policy

When a new RMCP-2 session is created, the SM, together with the SMA and the DMAs, establish the security policy for the session. The policy is established through the exchange of SECAGREQ, SECLIST and SECAGANS messages that enable the selection of the parameters in Table 32 to define the level of security that is to be provided, as well as the choice of algorithms to be used. The security policy is the set of selected attributes of policy items after the agreement on security mechanisms.

Table 32 – Multicast security policy

Item	Attributes	Definition	Further details
CON_EN_DEC_ID	– AES CBC Mode 128-bit key – AES CTR Mode 128-bit key – PKCS #1 – SEED	Notifies which encryption/decryption algorithm is used for content data	See Table 38
GK_EN_DEC_ID	– AES CBC Mode 128-bit key – AES CTR Mode 128-bit key – PKCS #1 – SEED	Notifies which encryption/decryption algorithm is used for content data for group keys	See Table 38
AUTH_ID	– HMAC-SHA – HMAC-MD5 – MD5	Notifies which hash/MAC algorithm is applied	See Table 39
GP_ATTRIBUTE	– closed – open (default)	Notifies the nature of the group	See Table 40
GK_MECHA	– static – periodic – backward – forward – periodic+backward – periodic+forward – periodic+backward+forward	Notifies updating properties of the group key	See Table 41
GK_NAME	– KDC – GKMP – MIKEY – GSAKMP – LKH	Notifies which group key mechanism is used.	See Table 42
AUTH_ATTRIBUTE	– membership	Notifies the type of authentication used	See Table 43
AUTH_NAME	– MEM_AUTH	Notifies the authentication mechanism used	See Table 44

10.1.4 Agreement of security mechanisms

10.1.4.1 SMA and DMAs

The security procedure is initiated after the admission control. The messages are protected by the session key between the SM, SMAs and DMAs, and by the K_{TLS} between the SM and the RMAs. The SMA and the DMAs perform the procedure prior to RMA subscription because the server-oriented systems (SMA and DMAs) need to set up the security policy in order to provide a stable service. The SMA and DMAs (see Figure 97) each request a security agreement (SECAGREQ) containing their own security mechanisms and algorithms. After a Security Agree.time, the SM examines the SECAGREQ messages, determines the security policy for the session and sends the security policy (SECLIST) to the SMA and DMAs. If any of these MAs do not have the algorithms of the security policy, they request copies from the SM (SECALGREQ) and the SM sends the corresponding security modules to them. The method for the delivery of these modules is outside the scope of this Recommendation | International Standard. The SMA and each DMA configure the agreed security mechanisms. After configuration, the MAs send an acknowledgement (SECAGANS) to the SM.

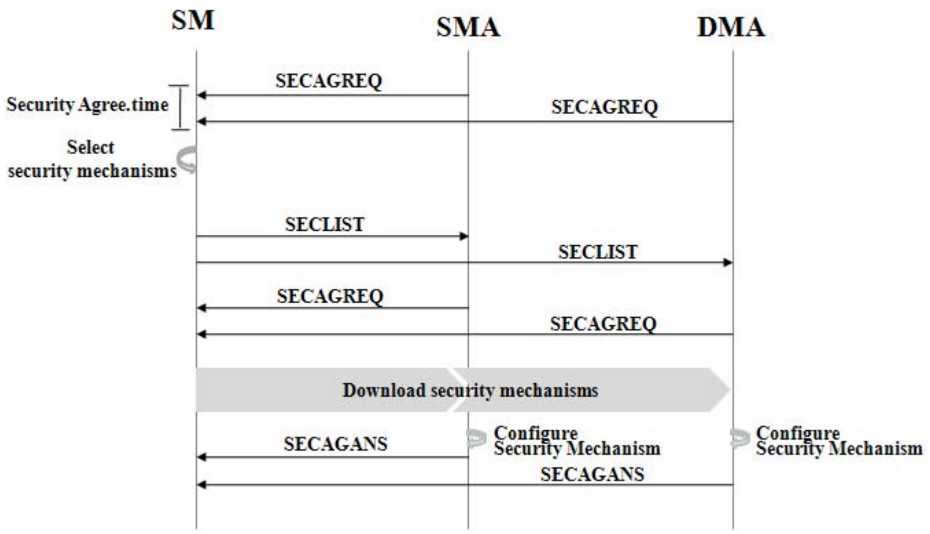


Figure 97 – Security agreement of DMA and SMA

10.1.4.2 RMAs

When the session is opened for RMA subscription, each RMA requests a security agreement (SECAGREQ) (see Figure 98). The SM sends the security policy (SECLIST) to the RMA. If the RMA does not have any of the algorithms of the security policy, it requests copies from the SM (SECALGREQ) and the SM sends the corresponding security modules to the RMA. The method for the delivery of these modules is outside the scope of this Recommendation | International Standard. The RMA configures the agreed security mechanisms and sends an acknowledgement (SECAGANS) to the SM.

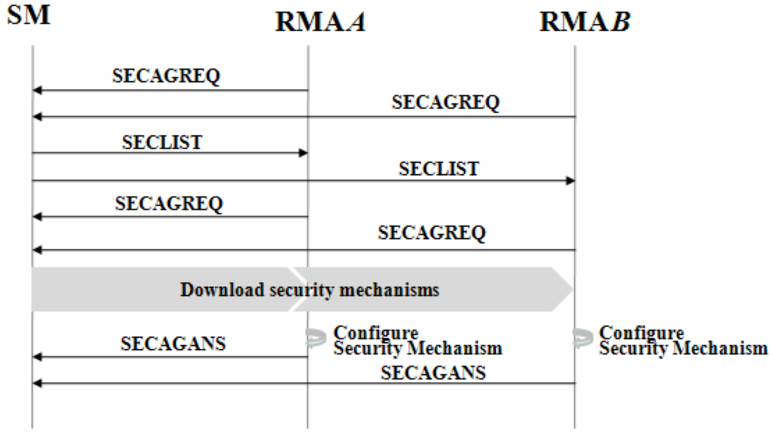


Figure 98 – Security agreement of RMAs

10.1.5 Access control for RMAs

The SM creates an access control list (ACL) containing hashed MAID and HASHED_AUTH for each authenticated RMA in the current session. Figure 99 illustrates the ACL procedure. After the session has been opened to RMAs, a DMA may request an ACL from the SM using an HRSREQ message encrypted by Ks. The SM responds with an HRSANS message encypted by Ks which contains the ACL. A DMA may update its ACL information through the periodic exchange of HRSREQ and HRSANS messages with the SM.

A DMA shall reject a request from an RMA to join the group if the ACL list does not contain the information for that RMA.

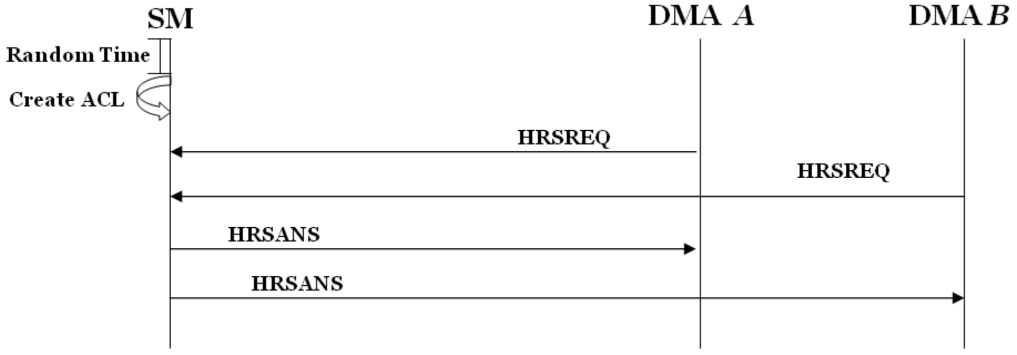


Figure 99 – ACL management

10.2 MA operation

As main components of the secure RMCP-2 protocol, the SMA and the DMAs are responsible for secure tree configuration and key management, as well as for group and member management and message encryption/decryption.

10.2.1 Key management for which the SMA and DMAs are responsible

10.2.1.1 Group key management

A group key (Kg) is shared between a DMA and its child RMAs, and it is used in an MM-group for data delivery. The Kg is initially created by the DMA and is encrypted by K_{MAS} (see 10.2.1.3) for delivery to its RMAs in the RELANS message confirming successful membership authentication (see 11.2.4).

Kg is updated by the DMA or RMA according to the update conditions selected for the security policy (see the GK_MECHA control in Table 32).

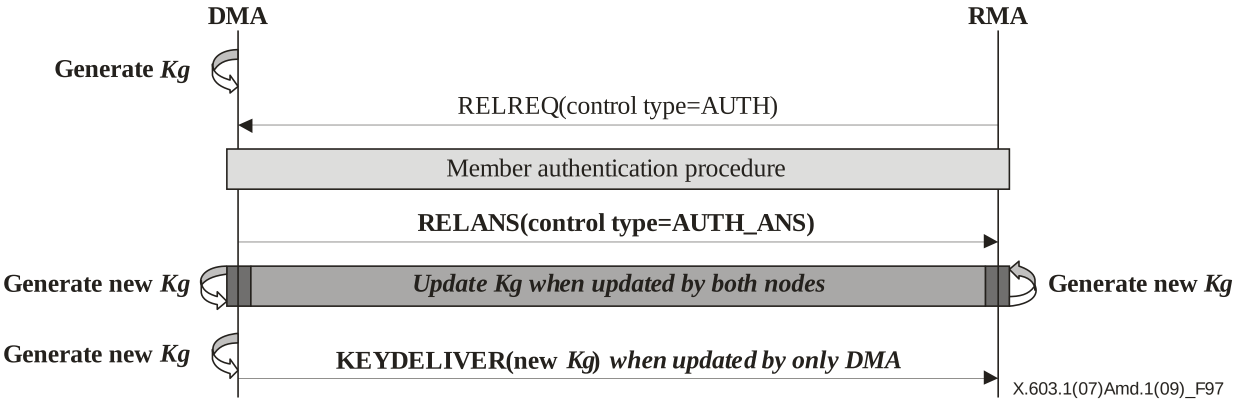

DMA
RMA
Generate *Kg*
RELREQ(control type=AUTH)
Member authentication procedure
RELANS(control type=AUTH_ANS)
Generate new *Kg*
Update Kg when updated by both nodes
Generate new *Kg*
Generate new *Kg*
KEYDELIVER(new *Kg*) *when updated by only DMA*
X.603.1(07)Amd.1(09)_F97

Figure 100 – Group key management

10.2.1.2 Contents encryption key management

The contents encryption key (Kc) is shared between the SMA and RMAs in the RMCP-2 session and is used to encrypt/decrypt contents data. Kc is generated by the SMA and is delivered to RMAs through the intermediate DMAs on the delivery path. Kc is encrypted by Ks for transmission between the SMA and DMAs and is encrypted by Kg for transmission between the DMAs and the RMAs. Kc key information need not be known by the SM or intermediate DMAs.

Kc is randomly updated by the SMA at periodic intervals. The delivery of Kc is synchronized with the delivery of the contents data (see 10.2.7).

10.2.1.3 Membership authentication Key

The membership authentication key K_{MAS} is a private key generated as a result of successful membership authentication between the RMAs and their parent DMA, as specified in Annex E. Each RMA shares a different K_{MAS} with the DMA and this is not shared with the other RMAs in the same group. K_{MAS} is not updated while the RMA remains a member of the relevant group.

10.2.2 Secure session subscription

The procedure for secure session subscription for the SMA, DMAs and RMAs is described in 10.1.1.2, 10.1.1.3, 10.1.1.4 and 10.1.1.5. This procedure is illustrated in Figure 101.

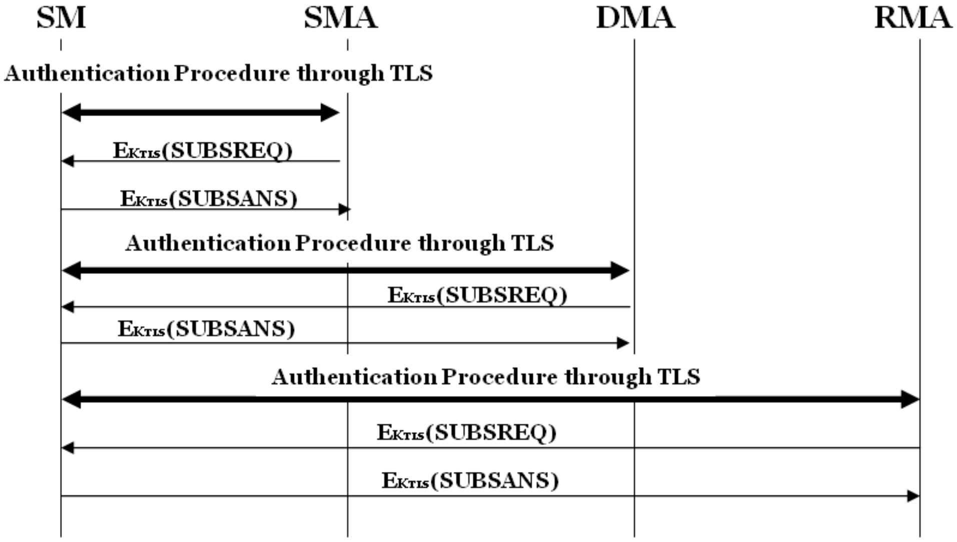

SM
SMA
DMA
RMA
Authentication Procedure through TLS
EKTIS(SUBSREQ)
EKTIS(SUBSANS)
Authentication Procedure through TLS
EKTIS(SUBSREQ)
EKTIS(SUBSANS)
Authentication Procedure through TLS
EKTIS(SUBSREQ)
EKTIS(SUBSANS)

Figure 101 – Secure MA subscription

10.2.3 Membership authentication for joining RMCP tree

Although DMAs are authenticated by the SM through TLS authentication, there is also a need for the DMAs and RMAs to verify their membership authority upon joining the RMCP tree and for construction of the pathway from the SMA to the RMAs. This procedure is important for the integrity of the RMCP-2 tree.

The membership authentication procedure defined in Annex E is used for mutual authentication.

The procedure is illustrated in Figure 102. The RMA|DMA sends a RELREQ message confirming the use of the membership authentication mechanism defined in Annex E. The SMA|DMA responds with a RELANS message containing the authentication result in the AUTH_ANS control. If the recipient is an RMA, the message to the RMA shall include the KEY_MATERIAL sub-control.

On receipt of confirmation by the RMA, the TLS session between the SM and the RMA need not be maintained.

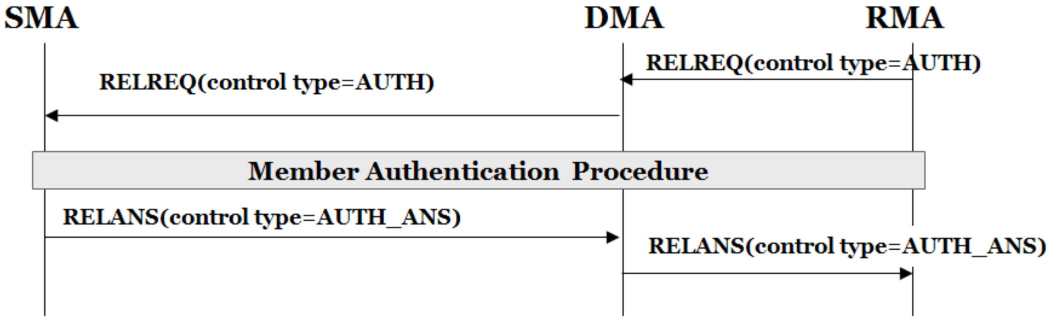

SMA
DMA
RMA
RELREQ(control type=AUTH)
RELREQ(control type=AUTH)
Member Authentication Procedure
RELANS(control type=AUTH_ANS)
RELANS(control type=AUTH_ANS)

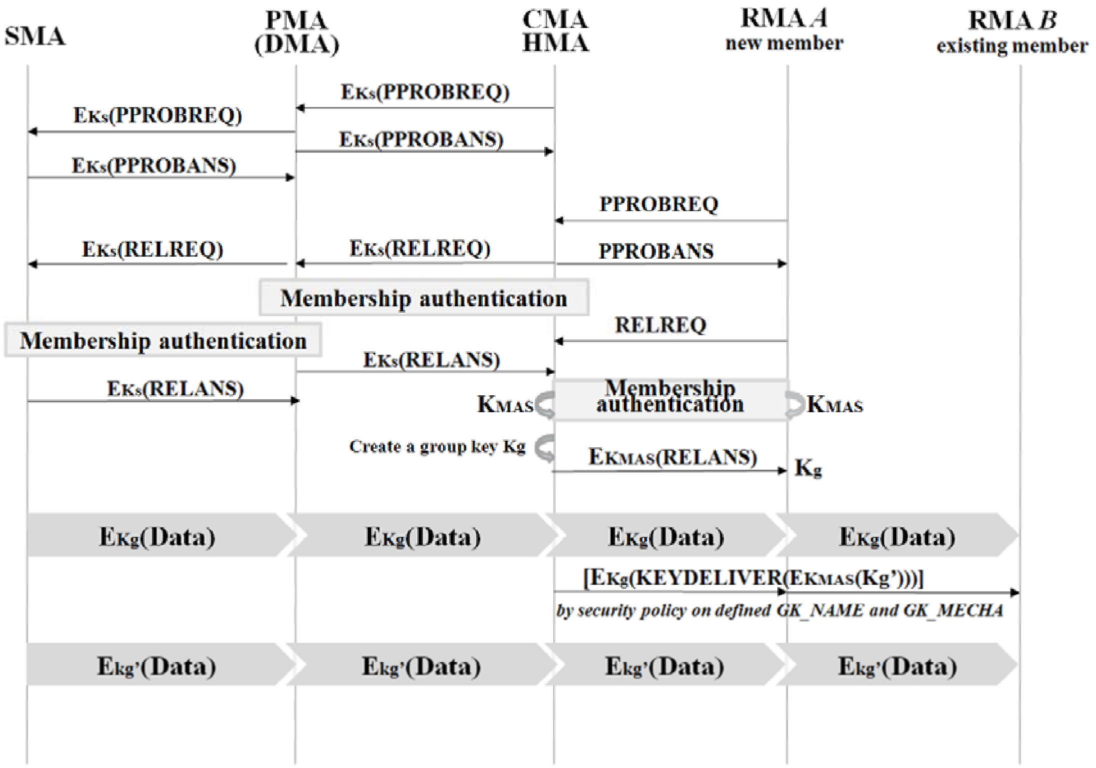
Figure 102 – Authentication between MAs

10.2.4 Secure tree join

Map discovery (see 6.2.2) occurs before the tree join procedure. Map discovery messages (PPROBREQ and PPROBANS) between DMAs are securely transmitted using Ks. Map discovery messages between RMAs and DMAs are delivered with hashed AUTH in plain text.

The tree join procedure is illustrated in Figure 103. Membership authentication (see 10.2.3) and group key distribution are processed. When the group key update is required (as indicated by the defined GK_MECHA code in the SECLIST, see Table 41), the parent DMA (see Note) of the RMA joining the tree re-creates and distributes the group key to its RMAs using the GK_NAME mechanism selected for the security policy (see Table 42). When this procedure is completed, the TLS session between the SM and the RMA is closed.

NOTE – In the case of a multicast-enabled group, the parent DMA will be the HMA.



NOTE – The PPROBREQ, PPROBANS and RELREQ messages between RMA A and the HMA are not encrypted, as RMA A has not yet received the KMAS or Kg keys.

Figure 103 – Secure tree join

10.2.5 Secure tree leave

Whenever an HMA, DMA or RMA leaves the group, the group key or the session key may be updated on the defined GK_MECHA code of multicast security policy (see Table 41).

10.2.5.1 Leave of RMA from multicast-enabled and multicast-disabled areas

When an RMA leaves, it notifies its parent DMA (its HMA in the case of multicast-enabled areas) and it is truncated from the tree. The DMA acknowledges the result, and updates and distributes the updated group key to the remaining members (see Figure 104). No further notification is required.

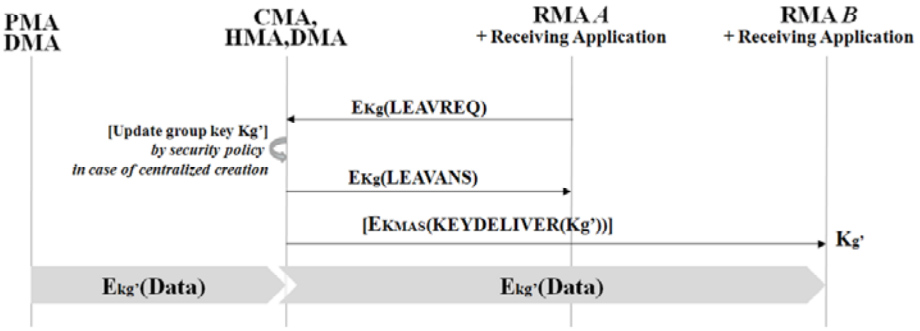


Figure 104 – Secure leave of RMA

10.2.5.2 Leave of HMA from a multicast-enabled area

Figure 105 illustrates the HMA leave procedure. The HMA issues a leave request to its members, and announces the leave to its candidate HMAs. The successful candidate HMA joins the RMCP-2 tree and announces its existence to the RMAs in its MM group. The RMAs request to re-join tree and perform membership authentication with the new HMA. The RMAs are then able to receive multicast data normally from the new HMA, and the old HMA leaves the RMCP-2 tree or see Figure 105.

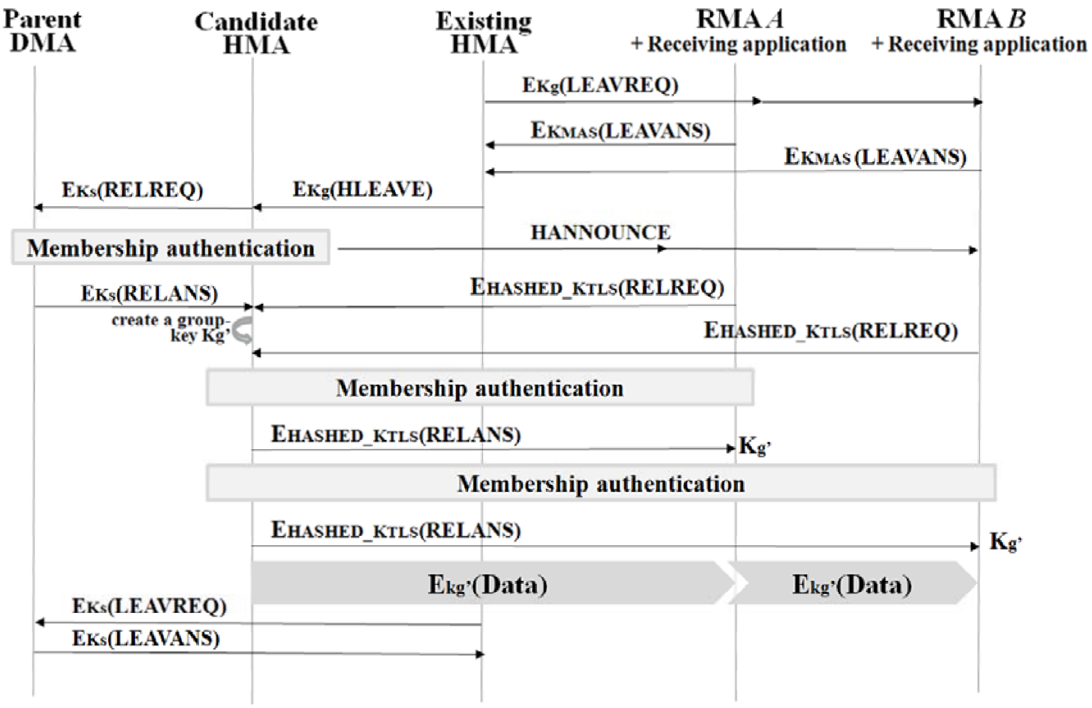


Figure 105 – HMA leave in multicast-enabled area

10.2.5.3 Leave of DMA from a multicast-disabled area

Figure 106 illustrates the leave of a DMA from a multicast-disabled area. The DMA (PMA A of B, C) announces its departure from the RMCP tree to its children B, C. CMAs B and C search for their candidate PMA and perform the join procedure as shown in Figure 106. CMAs B and C request to join the RMCP tree at the node of the candidate PMA. The PMA verifies authenticity of CMAs B and C, and if the authentication check is successful, it sends RELANS to confirm the graft to the RMCP tree. The PMA of B, C then initiates the leaving procedure with its PMA.

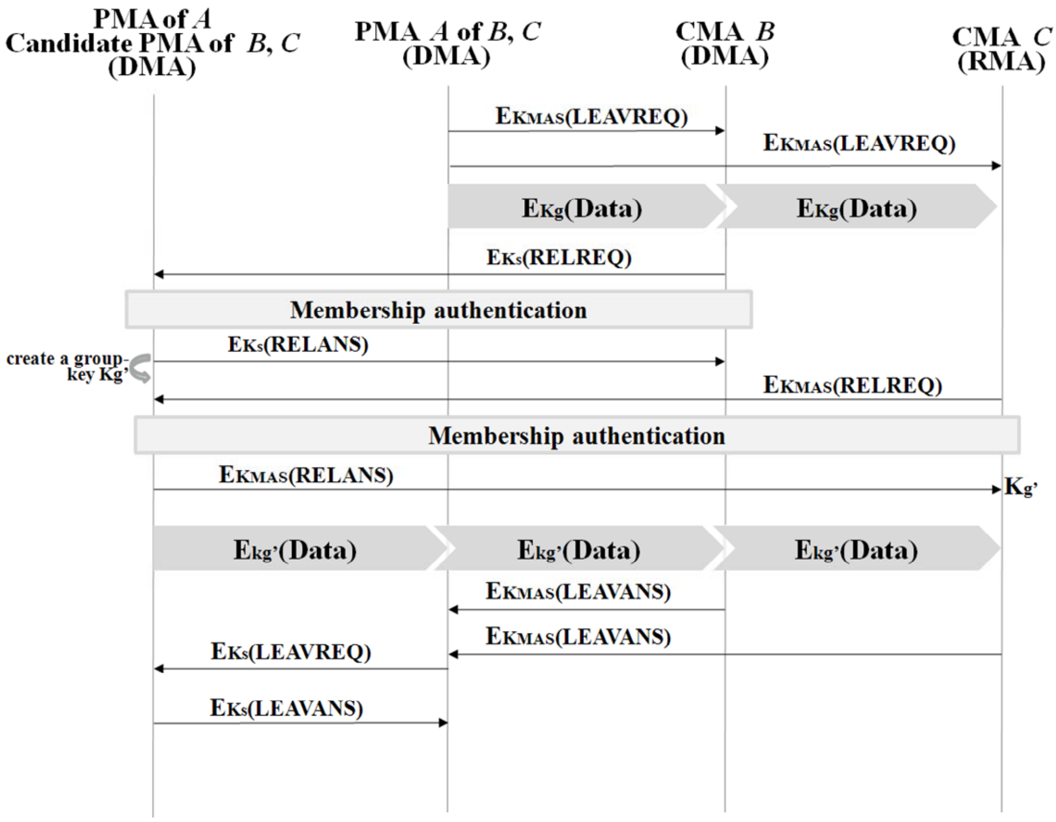


Figure 106 – DMA leave in multicast-disabled area

Membership authentication is performed between the RELREQ and RELANS messages in cases when a CMA is expelled by the PMA. If the SM expels an MA, the LEAVREQ and LEAVANS messages are en/decrypted.

10.2.6 Control message encryption/decryption

All secure RMCP-2 messages between the SM, SMA and DMAs are encrypted using agreed encryption algorithms in the SECLIST. Messages between RMAs and their parent DMA are encrypted by K_{MAS}, as shown in Table 33.

Table 33 – Encryption of basic and secure RMCP-2 protocol messages

Messages	Meaning	Key	
		DMA	RMA
SUBSREQ	Subscription request	*Ks*	K_{TLS}
SUBSANS	Subscription answer		K_{TLS}
PPROBREQ	Parent probe request		N/A
PPROBANS	Parent probe answer		N/A
HSOLICIT	HMA solicit		N/A
HANNOUNCE	HMA announce		N/A
HLEAVE	HMA leave		N/A
RELREQ	Relay request		K_{MAS}
RELANS	Relay answer		K_{MAS}
STREQ	Status report request		K_{TLS}
STANS	Status report answer		K_{TLS}
STCOLREQ	Status collect request		*N/A*
STCOLANS	Status collect answer		*N/A*
LEAVREQ	Leave request		K_{MAS}
LEAVANS	Leave answer		K_{MAS}

Table 33 – Encryption of basic and secure RMCP-2 protocol messages

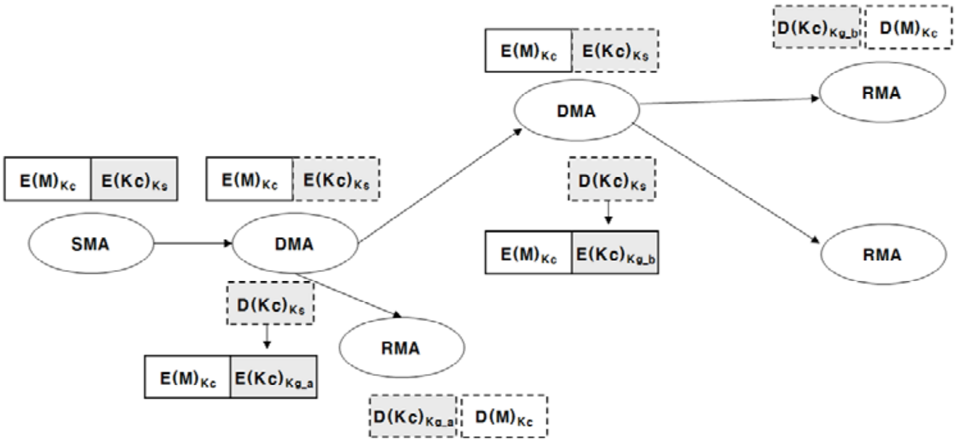
Messages	Meaning	Key	
		DMA	RMA
HB	Heartbeat		*N/A*
TERMREQ	Termination request		*HASHED* K_{TLS}
TERMANS	Termination answer		*HASHED* K_{TLS}
SECAGREQ	Security agreement request		K_{TLS}
SECLIST	Security list		K_{TLS}
SECALGREQ	Security algorithm request		K_{TLS}
SECAGANS	Security agreement answer		K_{TLS}
KEYDELIVER	Key delivery		K_{MAS}, *Kg*
HRSREQ	Head Required Security request		*N/A*
HRSANS	Head Required Security answer		*N/A*

10.2.7 Encryption/decryption and delivery of contents data

The contents are securely forwarded from the SMA to RMAs through the RMCP tree. Streaming or reliable data encrypted by *Kc* is delivered to individual RMAs without a decryption process at the intermediate nodes. In contrast, the key information is encrypted at intermediate nodes. The SMA encrypts *Kc* using *Ks* and delivers it to DMAs. The DMAs then decrypt the key information and encrypt it using Kg for delivery to RMAs in their own MM groups. Figure 107 illustrates how the encryption and decryption may be implemented.

The data and key information may be delivered separately. If separately transmitted, they should be synchronized.

NOTE – The encrypted data is efficiently transmitted to the RMAs without change in order to reduce the time of encryption/decryption by the intermediate nodes. Faster transmission is enabled due to the considerably reduced computation time.



NOTE – E(M) and D(M) refer to encrypted and decrypted data. E(Kc) and D(Kc) refer to encrypted and decrypted contents key information. Subscripts refer to keys used to encrypt (M) and (Kc). The suffixes $_{Kg\_a}$ and $_{Kg\_b}$ are used to distinguish different group keys used in separate MM groups.

Figure 107 – Example of data encryption/decryption

11 Format of secure RMCP-2 messages

11.1 Common format for secure RMCP-2 messages

The common format for secure RMCP-2 messages is the same as for RMCP-2 messages (see 7.1 and Figure 31) except that:

a) all secure RMCP-2 messages, including those that are defined for RMCP-2 in 7.3 and used in the secure RMCP-2 protocol, shall be defined as version 0x4; and

b) the range of valid Node Types for secure RMCP-2 messages is SM|SMA|DMA|RMA.

11.2 Secure RMCP-2 messages

This subclause defines those messages that are specific to RMCP-2 security. They are used in addition to the messages already defined in 7.3. Specific reference is made to the values for individual parameters that are defined in tables associated with clause 12.

11.2.1 SUBSREQ message

The SUBSREQ message for RMCP-2 is defined in 7.3.1 and its common format fields are shown in Figure 40. For use in secure RMCP-2, the following common format fields in the SUBSREQ message shall be set as indicated below:

a) *Version* – This field denotes the current version of RMCP-2. Its value shall be set to 0x4.

b) *Node type* – This field denotes the message issuer's node type. Its value shall be set to one of the SMA, DMA or RMA coded as in Table 34.

The remaining common format fields for SUBSREQ messages shall be as specified in 7.3.1.

0	8	16	31
Control Type (SERV_USER_IDENT)	Length (variable)		
SERV_USER_ID (variable length)			

Figure 108 – SERV_USER_IDENT control data

This subclause defines an additional SERV_USER_IDENT control type for use in secure RMCP-2, in order to confirm that the RMA issuing the SUBSREQ message has been registered by the Content Provider for participation in closed groups (see 10.1.1.5). The SERV_USER_IDENT control type shall be used only when the RMA joins a secure RMCP-2 session in which the MM groups are defined as closed.

The format of the SERV_USER_IDENT control type is shown in Figure 108. The description of each field is as follows:

- SERV_USER_IDENT

  a) *Control type* – This field denotes SERV_USER_IDENT control. Its value shall be set to 0x22 (see Table 36).

  b) *Length* – This field shall be set to the length in bytes of the SERV_USER_IDENT control in bytes.

  c) *SERV_USER_ID* – This field denotes the service user identifier allocated to the RMA by the Content Provider (see 10.1.1.5). Its value shall be identical to that provided to the RMA by the Content Provider.

NOTE – The length of the SERV_USER_ID field and the SERV_USER_IDENT control will be dependent on the length of the identifier provided by the Content Provider.

11.2.2 SUBSANS message

Two additional result codes, specific to the secure RMCP-2 protocol, are defined in Table 45 in order to record reasons for rejecting the subscription of an RMA due to a missing or unrecognized SERV_USER_ID in the SUBSREQ message, in cases where the session supports closed groups. These values extend the range of valid codes but do not affect the formatting of the RESULT control of the SUBSANS message specified in 7.3.2.

11.2.3 RELREQ message

11.2.3.1 The RELREQ message for RMCP-2 is defined in 7.3.8, and its common format fields are shown in Figure 56. For use in secure RMCP-2, the following common format fields in the RELREQ message shall be set as indicated below:

a) *Version* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *Node type* – This field denotes the message issuer's node type. Its value shall be set to one of the DMA or RMA coded, as in Table 34.

The remaining common format fields for RELREQ messages shall be as specified in 7.3.8.

0	8	16	31
Control Type (AUTH)	Length (= 4)	AUTH_NAME	Reserved (0x00)

Figure 109 – AUTH control data

11.2.3.2 This subclause defines an additional AUTH control for use in secure RMCP-2 in order to initiate membership authentication. This control is a mandatory part of the secure RMCP_2 RELREQ message.

The format of the AUTH control type is shown in Figure 109. The description of each field is as follows:

- AUTH
  a) *Control type* – This field denotes AUTH control. Its value shall be set to 0x23 (see Table 36).
  b) *Length* – This field denotes the length in bytes of the AUTH control. Its value shall be set to 0x04.
  c) *AUTH_NAME* – This field denotes the authentication mechanism. Its value shall be set to 0x01 denoting MEM_AUTH (see Table 44).
  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

11.2.4 RELANS message

11.2.4.1 The RELANS message for RMCP-2 is defined in 7.3.9, and its common format fields are shown in Figure 58. For use in secure RMCP-2, the following common format fields in the RELANS message shall be set as indicated below:

a) *Version* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *Node type* – This field denotes the message issuer's node type. Its value shall be set to one of the SMA or DMA coded, as in Table 34.

The remaining common format fields for RELANS messages shall be as specified in 7.3.9.

<table>
<tr><th>0</th><th>8</th><th>16</th><th>24</th><th>31</th></tr>
<tr><td>Control Type
(AUTH_ANS)</td><td>Length (= 4)</td><td>Auth_result</td><td colspan="2">Key_Flag</td></tr>
<tr><td>Sub-control type
(KEY_MATERIAL)</td><td colspan="2">Length (= variable up to 0x804)</td><td colspan="2">Key_Type</td></tr>
<tr><td colspan="5">Key_DATA</td></tr>
</table>

Figure 110 – AUTH_ANS control, including KEY_MATERIAL sub-control

11.2.4.2 This subclause defines an additional AUTH_ANS control for use in secure RMCP-2 in order to notify the result of membership authentication. This control is a mandatory part of the secure RMCP_2 RELANS message.

Figure 110 shows the format of the AUTH_ANS control type and its KEY_MATERIAL sub-control type. The description of each field of the AUTH_ANS control is as follows:

- AUTH_ANS

  a) *Control type* – This field denotes the AUTH_ANS control. Its value shall be set to 0x24 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the AUTH_ANS control. Its value shall be set to 0x04.

  c) *Auth_result* – This field denotes the result of authentication. Its value shall be set to 0x01 for successful authentication; in the case of unsuccessful authentication, the value shall be set to one of the other codes in Table 46.

  d) *Key_Flag* – This field denotes the presence or absence of key information in the KEY_MATERIAL sub-control of the AUTH_ANS control. Its value shall be set to 0x01 if key information is provided in the message; its value shall be set to 0x00 if this information is not provided.

11.2.4.3 The KEY_MATERIAL sub-control shall not be included in the RELANS message if the key flag is set to 0x00. The description of each field of the KEY_MATERIAL sub-control is as follows:

- KEY_MATERIAL

  a) *Sub-control type* – This field denotes the KEY_MATERIAL sub-control. Its value shall be set to 0x01 (see Table 37).

  b) *Length* – This field shall be set to the total length of the KEY_MATERIAL sub-control in bytes. Its value shall not exceed 0x804.

  c) *Key_Type* – This field denotes the type of the key information. Its value shall be set to one of the code values in Table 47.

  d) *Key_DATA* – This field shall contain key information resulting from 10.2.3, and it shall be included if the receiver is an RMA.

11.2.5 SECAGREQ message

11.2.5.1 The format of the SECAGREQ message is shown in Figure 111. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of the SMA, DMA or RMA coded as in Table 34.

c) *Message type* – This field denotes the type of SECAGREQ message. Its value shall be set to 0x21 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the SECAGREQ message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the proposed MAID of the SECAGREQ sender. Its value shall contain the local IP address and port number.

g) *Control data* – The controls associated with the SECAGREQ message are specified in 11.2.5.2-11.2.5.5. The following conditions apply to the use of these controls:

   The SMA_PROPOSE control in 11.2.5.2 is used by the SMA to propose values to the SM for GR_ATTRIBUTE, GK_MECHA and CON_EN_DEC_ID and shall be included in a SECAGREQ message sent by the SMA. This control shall not be included in a SECAGREQ message sent by a DMA or an RMA.

   The controls in 11.2.5.3-11.2.5.5 are used to indicate the capabilities of the SMA and DMAs during the establishment of the security policy (see 10.1.3 and 10.1.4). These controls shall not be included in a SECAGREQ message sent by an RMA or by a DMA that joins the session after the security policy has been established.

0	4	8	16 24 31
Ver (0x4)	NT (SMA|DMA|RMA)	Message type (SECAGREQ)	Length (variable)
Session ID (64)			
MAID (of the SECAGREQ sender)			
Control data (variable length)			

Figure 111 – SECAGREQ message

11.2.5.2 The format of the SMA_PROPOSE control is shown in Figure 112. The description of each field is as follows:

- SMA_PROPOSE
  a) *Control type* – This field denotes the SMA_PROPOSE control. Its value shall be set to 0x11 (see Table 36).
  b) *Length* – This field denotes the length in bytes of the SMA_PROPOSE control. Its value shall be set to 0x08.
  c) *GP_ATTRIBUTE* – This field denotes the group property proposed by the SMA. Its value shall be set to one of the code values in Table 40.
  d) *GK_MECHA* – This field denotes the update property of the group key proposed by the SMA. Its value shall be set to one of the code values in Table 41.
  e) *CON_EN_DEC_ID* – This field denotes the contents encryption algorithm proposed by the SMA. Its value shall be set to one of the code values less than 1x00 in Table 38.
  f) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (SMA_PROPOSE)	Length (= 8)	GP_ATTRIBUTE	GK_MECHA
CON_EN_DEC_ID	Reserved (0x00)		

Figure 112 – SMA_PROPOSE control

11.2.5.3 The format of the GK_MECH_CAPAB control is shown in Figure 113. This control may be repeated in order to indicate several mechanisms, each with their own order of preference. The description of each field is as follows:

- GK_MECH_CAPAB
  a) *Control type* – This field denotes the GK_MECH_CAPAB control. Its value shall be set to 0x12 (see Table 36).
  b) *Length* – This field denotes the length in bytes of the GK_MECH_CAPAB control. Its value shall be set to 0x04.
  c) *GK_NAME* – This field denotes a security mechanism held by the SMA or DMA for possible use in the secure RMCP-2 session. Its value shall be set to one of the code values in Table 42.
  d) *PREFER* – This field denotes the priority of the proposed security mechanism in the preceding field. Its value shall be set to an integer in the range 1 to 6. The integer '1' shall indicate the highest priority.

0	8	16	24 31
Control Type (GK_MECH_CAPAB)	Length (= 4)	GK_NAME	PREFER
Control Type (GK_MECH_CAPAB)	Length (= 4)	GK_NAME	PREFER
Control Type (GK_MECH_CAPAB)	Length (= 4)	GK_NAME	PREFER

Figure 113 – GK_MECH_CAPAB control

11.2.5.4 The format of the EN_DEC_CAPAB control is shown in Figure 114. This control may be repeated in order to indicate several mechanisms, each with their own order of preference. The description of each field is as follows:

- EN_DEC_CAPAB

  a) *Control type* – This field denotes the EN_DEC_CAPAB control. Its value shall be set to 0x13 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the EN_DEC_CAPAB control. Its value shall be set to 0x04.

  c) *EN_DEC_ID* – This field denotes a proposed encryption algorithm held by the SMA or DMA for possible use in the secure RMCP-2 session. Its value shall be set to one of the code values in Table 38.

  d) *PREFER* – This field denotes the priority of the proposed security mechanism in the preceding field. Its value shall be set to an integer in the range 1 to 5. The integer '1' shall indicate the highest priority.

0	8	16	24 31
Control Type (EN_DEC_CAPAB)	Length (= 4)	EN_DEC_ID	PREFER
Control Type (EN_DEC_CAPAB)	Length (= 4)	EN_DEC_ID	PREFER
Control Type (EN_DEC_CAPAB)	Length (= 4)	EN_DEC_ID	PREFER

Figure 114 – EN_DEC_CAPAB control

11.2.5.5 The format of the AUTH_ALG_CAPAB control is shown in Figure 115. This control type may be repeated in order to indicate several mechanisms, each with their own order of preference. The description of each field is as follows:

- AUTH_ALG_CAPAB

  a) *Control type* – This field denotes the AUTH_ALG_CAPAB control. Its value shall be set to 0x14 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the AUTH_ALG_CAPAB control. Its value shall be set to 0x04.

  c) *AUTH_ID* – This filed denotes a hash/MAC algorithm held by the SMA or DMA for possible use in the secure RMCP-2 session. Its value shall be set to one of the code values in Table 39.

  d) *PREFER* – This field denotes the priority of the proposed security mechanism in the preceding field. Its value shall be set to an integer in the range 1 to 3. The integer '1' shall indicate the highest priority.

0	8	16	24 31
Control Type (AUTH_ALG_CAPAB)	Length (= 4)	AUTH_ID	PREFER
Control Type (AUTH_ALG_CAPAB)	Length (= 4)	AUTH_ID	PREFER
Control Type (AUTH_ALG_CAPAB)	Length (= 4)	AUTH_ID	PREFER

Figure 115 – AUTH_ALG_CAPAB control

11.2.6 SECLIST message

11.2.6.1 The format of the SECLIST message is shown in Figure 116. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the code value for the SM in Table 34.

c) *Message type* – This field denotes the SECLIST message. Its value shall be set to 0x22 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the SECLIST message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID of the SECLIST recipient. Its value shall be as defined in 7.1.2.

g) *Control data* – The controls associated with the SECLIST message are specified in 11.2.6.2-11.2.6.6. All of these controls are a mandatory part of the message.

<table>
<tr><th colspan="2">0</th><th>8</th><th>16 24 31</th></tr>
<tr><td>Ver (0x4)</td><td>NT
(SM)</td><td>Message type
(SECLIST)</td><td>Length (variable)</td></tr>
<tr><td colspan="4">Session ID</td></tr>
<tr><td colspan="4">MAID (of the SECLIST recipient)</td></tr>
<tr><td colspan="4">Control data (variable length)</td></tr>
</table>

Figure 116 – SECLIST message

11.2.6.2 The format of the GK_MECH control is shown in Figure 117. The description of each field is as follows:

- GK_MECH

  a) *Control type* – This field denotes the GK_MECH control. Its value shall be set to 0x15 (see Table 36).

  b) *Length* – This field denotes the length in bytes of GK_MECH control. Its value shall be set to 0x08.

  c) *GP_ATTRIBUTE* – This field denotes the group property for the security policy. Its value shall be set to one of the code values in Table 40.

  d) *GK_NAME* – This field defines the group key mechanism for the security policy. Its value shall be set to one of the code values in Table 42.

  e) *GK_MECHA* – This field denotes the update property of group key for the security policy. Its value shall be set to one of the code values in Table 41.

0	8	16	24 31
Control Type (GK_MECH)	Length (= 8)	GP_ATTRIBUTE	GK_NAME
GK_MECHA	Reserved (0x00)		

Figure 117 – GK_MECH control

11.2.6.3 The format of the AUTH_MECH control is shown in Figure 118. The description of each field is as follows:

- AUTH_MECH

  a) *Control type* – This field denotes the AUTH_MECH control. Its value shall be set to 0x16 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the AUTH_MECH control. Its value shall be set to 0x04.

  c) *AUTH_ATTRIBUTE* – This field denotes the authentication type for the security policy. Its value shall be set to 0x01 denoting MEMBERSHIP (see Table 43).

  d) *AUTH_NAME* – This denotes the authentication mechanism for the security policy. Its value shall be set to 0x01 denoting MEM_AUTH (see Table 44).

0	8	16	24 31
Control Type (AUTH_MECH)	Length (= 4)	AUTH_ATTRIBUTE	AUTH_NAME

Figure 118 – AUTH_MECH control

11.2.6.4 The format of the CON_EN_DEC_ALG control is shown in Figure 119. The description of each field is as follows:

- CON_EN_DEC_ALG

  a) *Control type* – This field denotes the CON_EN_DEC_ALG control. Its value shall be set to 0x17 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the CON_EN_DEC_ALG control. Its value shall be set to 0x04.

  c) *CON_EN_DEC_ID* – This field denotes the contents encryption algorithm for the security policy. Its value shall be set to one of the code values in Table 38.

  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (CON_EN_DEC_ALG)	Length (= 4)	CON_EN_DEC_ID	Reserved (0x00)

Figure 119 – CON_EN_DEC_ALG control

11.2.6.5 The format of the GK_EN_DEC_ALG control is shown in Figure 120. The description of each field is as follows:

- GK_EN_DEC_ALG

  a) *Control type* – This field denotes the GK_EN_DEC_ALG control. Its value shall be set to 0x18 (see Table 36).

  b) *Length* – This field denotes the length of the GK_EN_DEC_ALG control in bytes. Its value shall be set to 0x04.

  c) *GK_EN_DEC_ID* – This field denotes the group key encryption algorithm for the security policy. Its value shall be set to one of the code values in Table 38.

  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (GK_EN_DEC_ALG)	Length (= 4)	GK_EN_DEC_ID	Reserved (0x00)

Figure 120 – GK_EN_DEC_ALG control

11.2.6.6 The format of the AUTH_ALG control is shown in Figure 121. The description of each field is as follows:

- AUTH_ALG

  a) *Control type* – This field denotes the AUTH_ALG control. Its value shall be set to 0x19 (see Table 36).

  b) *Length* – This field denotes the length in bytes of the AUTH_ALG control. Its value shall be set to 0x04.

  c) *AUTH_ID* – This field denotes the hash/MAC algorithm for the security policy. Its value shall be set to one of the code values in Table 39.

  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (AUTH_ALG)	Length (= 4)	AUTH_ID	Reserved (0x00)

Figure 121 – AUTH_ALG control

11.2.7 SECALGREQ message

11.2.7.1 The format of the SECALGREQ message is shown in Figure 122. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of the SMA, DMA or RMA coded, as in Table 34.

c) *Message type* – This field denotes the SECALGREQ message. Its value shall be set to 0x27 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the SECALGREQ message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of the Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID of the SECALGREQ sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The controls associated with the SECALGREQ message, together with the conditions applying to their use, are specified in 11.2.7.2-11.2.7.6.

<table>
<tr><th>0</th><th>4</th><th>8</th><th>16 24 31</th></tr>
<tr><td>Ver (0x4)</td><td>NT
(SMA|DMA|RMA)</td><td>Message type
(SECALGREQ)</td><td>Length (variable)</td></tr>
<tr><td colspan="4">Session ID (64)</td></tr>
<tr><td colspan="4">MAID (of the SECALGREQ sender)</td></tr>
<tr><td colspan="4">Control data (variable length)</td></tr>
</table>

Figure 122 – SECALGREQ message

11.2.7.2 The format of the GK_MECH_DELIVER control is shown in Figure 123. This control shall only be used by the MA sending the SECALGREQ message when it does not hold the GK_NAME security algorithm, or when the configuration of this algorithm has failed (see the agreement of security mechanisms procedure in 10.1.4). The description of each field is as follows:

- GK_MECH_DELIVER
  a) *Control type* – This field denotes the GK_MECH_DELIVER control. Its value shall be set to 0x1A (see Table 36).
  b) *Length* – This field denotes the length in bytes of GK_MECH_DELIVER control. Its value shall be set to 0x04.
  c) *GK_NAME* – This field denotes the group key mechanism for the security policy. Its value shall be identical to that in the GK_NAME field in the GK_MECH control of the SECLIST message (see 11.2.6.2 d).
  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (GK_MECH_DELIVER)	Length (= 4)	GK_NAME	Reserved (0x00)

Figure 123 – GK_MECH_DELIVER control

11.2.7.3 The format of the AUTH_MECH_DELIVER control is shown in Figure 124. This control shall only be used by the MA sending the SECALGREQ message when it does not hold the AUTH_NAME security algorithm, or when the configuration of this algorithm has failed (see the agreement of security mechanisms procedure in 10.1.4). The description of each field is as follows:

- AUTH_MECH_DELIVER
  a) *Control type* – This field denotes the AUTH_MECH_DELIVER control. Its value shall be set to 0x1B (see Table 36).
  b) *Length* – This field denotes the length in bytes of the AUTH_MECH_DELIVER control. Its value shall be set to 0x04.
  c) *AUTH_NAME* – This field denotes the authentication mechanism for the security policy. Its value shall be set to 0x01 denoting MEM_AUTH (see Table 44).
  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (AUTH_MECH_DELIVER)	Length (= 4)	AUTH_NAME	Reserved (0x00)

Figure 124 – AUTH_MECH_DELIVER control

11.2.7.4 The format of the CON_EN_DEC_DELIVER control is shown in Figure 125. This control shall only be used by the MA sending the SECALGREQ message when it does not hold the CON_EN_DEC_ALG security algorithm, or when the configuration of this algorithm has failed (see the agreement of security mechanisms procedure in 10.1.4). The description of each field is as follows:

- CON_EN_DEC_DELIVER
  a) *Control type* – This field denotes the CON_EN_DEC_DELIVER control. Its value shall be set to 0x1C (see Table 36).
  b) *Length* – This field denotes the length of the CON_EN_DEC_DELIVER control in bytes. Its value shall be set to 0x04.
  c) *CON_EN_DEC_ID* – This field denotes the contents encryption algorithm for the security policy. Its value shall be identical to that in the CON_EN_DEC_ID field of the CON_EN_DEC_ALG control in the SECLIST message (see 11.2.6.4 c).
  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (CON_EN_DEC_DELIVER)	Length (= 4)	CON_EN_DEC_ID	Reserved (0x00)

Figure 125 – CON_EN_DEC_DELIVER control

11.2.7.5 The format of the GK_EN_DEC_DELIVER control is shown in Figure 126. This control shall only be used by the MA sending the SECALGREQ message when it does not hold the GK_EN_DEC_ALG security algorithm, or when the configuration of this algorithm has failed (see the agreement of security mechanisms procedure in 10.1.4). The description of each field is as follows:

- GK_EN_DEC_DELIVER

  a) *Control type* – This field denotes the GK_EN_DEC_DELIVER control. Its value shall be set to 0x1D (see Table 36).

  b) *Length* – This field denotes the length in bytes of the GK_EN_DEC_DELIVER control. Its value shall be set to 0x04.

  c) *GK_EN_DEC_ID* – This field denotes the group key encryption algorithm for the security policy. Its value shall be identical to that in the GK_EN_DEC_ID field of the GK_EN_DEC_ALG control in the SECLIST message (see 11.2.6.5 c).

  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (GK_EN_DEC_DELIVER)	Length (= 4)	GK_EN_DEC_ID	Reserved (0x00)

Figure 126 – GK_EN_DEC_DELIVER control

11.2.7.6 The format of the AUTH_ALG_DELIVER control is shown in Figure 127. This control shall only be used by the MA sending the SECALGREQ message when it does not hold the AUTH_ALG security algorithm, or when the configuration of this algorithm has failed (see the agreement of security mechanisms procedure in 10.1.4). The description of each field is as follows:

- AUTH_ALG_DELIVER

  a) *Control type* – This field denotes the AUTH_ALG_DELIVER control. Its value shall be set to 0x1E (see Table 36).

  b) *Length* – This field denotes the length in bytes of the AUTH_ALG_DELIVER control. Its value shall be set to 0x04.

  c) *AUTH_ID* – This field denotes the hash/MAC algorithm for the security policy. Its value shall be identical to that in the AUTH_ID field of the AUTH_ALG control in the SECLIST message (see 11.2.6.6 c).

  d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (AUTH_ALG_DELIVER)	Length (= 4)	AUTH_ID	Reserved (0x00)

Figure 127 – AUTH_ALG_DELIVER control

11.2.8 SECAGANS message

11.2.8.1 The format of the SECAGANS message is shown in Figure 128. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to one of SMA, DMA or RMA coded, as in Table 34.

c) *Message type* – This field denotes the SECAGANS message. Its value shall be set to 0x23 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the SECAGANS message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of the Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID of the SECAGANS sender. Its value shall be formatted as defined in 7.1.2.

g) *Control data* – The SEC_RETURN control specified in 11.2.8.2 is a mandatory part of the SECAGANS message.

0	4	8	16 24 31
Ver (0x4)	NT (SMA|DMA|RMA)	Message type (SECAGANS)	Length (variable)
Session ID (64)			
MAID (of the SECAGANS sender)			
Control data (variable length)			

Figure 128 – SECAGANS message

11.2.8.2 The format of the SEC_RETURN control is shown in Figure 129. The description of each field is as follows:

- SEC_RETURN

a) *Control type* – This field denotes the SEC_RETURN control. Its value shall be set to 0x1F (see Table 36).

b) *Length* – This field denotes the length in bytes of the SEC_RETURN control. Its value shall be set to 0x04.

c) *SEC_RETURN* – This field denotes the result of SECAGREQ request. Its value shall be set to 0x01 for a successful return; the value for other results shall be indicated by one of the other remaining codes in Table 46.

d) *Reserved* – This field is reserved for future use. Its value shall be set to 0x00.

0	8	16	24 31
Control Type (SEC_RETURN)	Length (= 4)	SEC_RETURN	Reserved (0x00)

Figure 129 – SEC_RETURN control

11.2.9 KEYDELIVER message

11.2.9.1 Figure 130 shows the format of the KEYDELIVER message. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to:

– 0x01, the coded value for SM in Table 34, for the delivery of the Ks key information; or

– 0x05, the coded value for DMA in Table 34, for the delivery of the Kg key information; or

– 0x02, the coded value for SMA in Table 34, for the delivery of the Kc key information.

c) *Message type* – This field denotes the KEYDELIVER message. The value shall be set to 0x24 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the KEYDELIVER message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID of the KEYDELIVER recipient. Its value shall be as defined in 7.1.2.

g) *Control data* – The KEY_INFO control and its KEY_MATERIAL sub-control, specified in 11.2.9.2 and 11.2.9.3, are a mandatory part of the KEYDELIVER message.

0	4	8	16 24 31
Ver (0x4)	NT (SM|SMA|DMA)	Message type (KEYDELIVER)	Length (variable)
Session ID (64)			
MAID (of the KEYDELIVER sender)			
Control data (variable length)			

Figure 130 – KEYDELIVER message

11.2.9.2 The format of the KEY_INFO control and its KEY_MATERIAL sub-control is shown in Figure 131. The description of each field of the KEY_INFO control is as follows:

- KEY_INFO

  a) *Control type* – This field denotes the KEY_INFO control. Its value shall be set to 0x20 (see Table 36).

  b) *Length* – This field denotes the length of the KEY_INFO control in bytes. Its value shall be set to 0x04.

  c) *Key_Type* – This field denotes the type of the proposed key information. Its value shall be set to one of the code values in Table 47.

0	8	16	24 31
Control Type (KEY_INFO)	Length (= 4)	Key_Type	Reserved (0x00)
Sub-control type (KEY_MATERIAL)	Length (= variable up to 0x804)		Key_Type
KEY_DATA			

Figure 131 – KEY_INFO control, including KEY_MATERIAL sub-control

11.2.9.3 The description of each field of the KEY_MATERIAL sub-control is as follows:

- KEY_MATERIAL

  a) *Sub-control type* – This field denotes the KEY_MATERIAL sub-control. Its value shall be set to 0x01 (see Table 37).

  b) *Length* – This field shall be set to the total length in bytes of the KEY_MATERIAL sub-control. Its value shall not exceed 0x804.

  c) *Key_Type* – This field denotes the type of the key information. Its value shall be set to one of the code values in Table 47.

  d) *KEY_DATA* – This field shall contain the time information and seed value needed to generate the key identified by Key_Type.

11.2.10 HRSREQ message

The format of the HRSREQ message is shown in Figure 132. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. The value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to the coded value for DMA in Table 34.

c) *Message type* – This field denotes the HRSREQ message. The value shall be set to 0x25 (see Table 35).

d) *Length* – This field denotes the length in bytes of the HRSREQ message. Its value shall be set to 0x14.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the proposed MAID of the HRSREQ sender. Its value shall be formatted as defined in 7.1.2.

NOTE – There is no control data associated with the HRSREQ message.

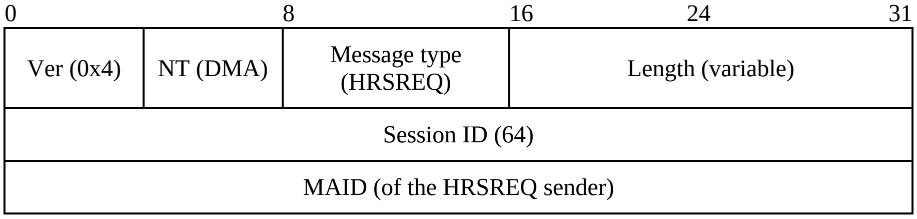


Figure 132 – HRSREQ message

11.2.11 HRSANS message

11.2.11.1 The format of the HRSANS message is shown in Figure 133. The description of each field is as follows:

a) *Ver* – This field denotes the current version of RMCP. Its value shall be set to 0x4.

b) *NT* – This field denotes the message issuer's node type. Its value shall be set to 0x01, the code value for the SM in Table 34.

c) *Message type* – This field denotes the HRSANS message. Its value shall be set to 0x26 (see Table 35).

d) *Length* – This field shall be set to the total length in bytes of the HRSANS message including the control data.

e) *Session ID* – This field shall be set to the 64-bit value of Session ID as defined in 7.1.1.

f) *MAID* – This field denotes the MAID of the HRSANS recipient. Its value shall be as defined in 7.1.2.

g) *Control data* – The ACL_LIST control its ACL_DATA sub-control, specified in 11.2.11.2 and 11.2.11.3, are a mandatory part of the HRSANS message.

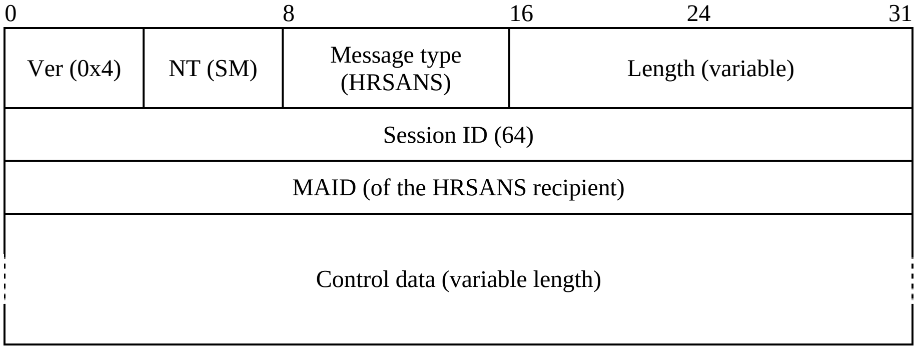


Figure 133 – HRSANS message

11.2.11.2 The format of the ACL_LIST control and its ACL_DATA sub-control is shown in Figure 134. The description of each field of the ACL_LIST control type is as follows:

- ACL_LIST
  a) *Control type* – This field denotes the ACL_LIST control. Its value shall be set to 0x21 (see Table 36).
  b) *Length* – This field denotes the length in bytes of the ACL_LST control. Its value shall be set to 0x02.

<table>
<tr><td>0</td><td>8</td><td>16</td><td>24 31</td></tr>
<tr><td>Control Type
(ACL_LIST)</td><td>Length (= 2)</td><td>Sub-control type
(ACL_DATA)</td><td>Reserved (0x00)</td></tr>
<tr><td colspan="2">Length (variable)</td><td colspan="2">N_ACL</td></tr>
<tr><td colspan="4">DATA(HASHED MAID|| HASHED K_{TLS})</td></tr>
<tr><td colspan="4">DATA(HASHED MAID|| HASHED K_{TLS})</td></tr>
<tr><td colspan="4">:</td></tr>
</table>

Figure 134 – ACL_LIST control, including ACL_DATA sub-control

11.2.11.3 The description of each field of the ACL_DATA sub-control is as follows:

- ACL_DATA
  a) *Sub-control type* – This field denotes the ACL_DATA sub-control. Its value shall be set to 0x02 (see Table 37).
  b) *Length* – This field shall be set to the length in bytes of the ACL_DATA sub-control.
  c) *N_ACL* – This field shall be set to the number of the entries in the ACL_list.
  d) *ACL_DATA* – This field shall contain the HASHED MAID, HASHED K_{TLS} for each authenticated RMA in the current session.

12 Parameters

12.1 Secure RMCP-2 node types and code values

Table 34 lists the node types and their corresponding code values.

Table 34 – Node type codes for secure RMCP-2

Code	Node type	Definition
0x01	SM	Session Manager
0x02	SMA	Sender Multicast Agent
0x03	RMA	Receiver Multicast Agent
0x05	DMA	Dedicated Multicast Agent

12.2 Secure RMCP-2 message types and code values

Table 35 lists the message types and their corresponding code values.

Table 35 – RMCP-2 message types and code values

Message type	Meaning	Code value (hexadecimal)	Cross reference to message format
SUBSREQ	Subscription request (Control type = SERV_USER_IDENT)	0x02	See 11.2.1
RELREQ	Relay request (Control type=AUTH)	0x09	See 11.2.3
RELANS	Relay answer (Control type =AUTH_ANS)	0x0C	See 11.2.4
SECAGREQ	Security Agreement Request	0x21	See 11.2.5
SECLIST	Selected Security List	0x22	See 11.2.6
SECALGREQ	Security Algorithms Request	0x27	See 11.2.7
SECAGANS	Security Agreement Answer	0x23	See 11.2.8
KEYDELIVER	Key Delivery	0x24	See 11.2.9
HRSREQ	Head Required Security Request	0x25	See 11.2.10
HRSANS	Head Required Security Answer	0x26	See 11.2.11

NOTE – The code values for the SUBSREQ, RELREQ and RELANS messages are as specified in Table 23 for basic RMCP-2 message types.

12.3 Secure RMCP-2 control types and code values

Table 36 lists the control types and their corresponding code values.

Table 36 – Control types for secure RMCP-2

Control type	Meaning	Code value (hexadecimal)	Message types containing the control type
SERV_USER_IDENT	Service User Identification	0x22	SUBSREQ
AUTH	Authentication	0x23	RELREQ
AUTH_ANS	Authentication Answer	0x24	RELANS
SMA_PROPOSE	Security profile values proposed by the SMA	0x11	SECAGREQ
GK_MECH_CAPAB	Group Key Mechanism Capabilities	0x12	SECAGREQ
EN_DEC_CAPAB	Encryption/Decryption algorithm Capabilities	0x13	SECAGREQ
AUTH_ALG_CAPAB	Hash/MAC Algorithm Capabilities	0x14	SECAGREQ
GK_MECH	Group Key Mechanism	0x15	SECLIST
AUTH_MECH	Authentication Mechanism	0x16	SECLIST
CON_EN_DEC_ALG	Contents Encryption/Decryption Algorithm	0x17	SECLIST
GK_EN_DEC_ALG	Group Key Encryption/Decryption Algorithm	0x18	SECLIST
AUTH_ALG	Hash/MAC Algorithm	0x19	SECLIST
GK_MECH_DELIVER	Group Key Mechanism Delivery Request	0x1A	SECALGREQ
AUTH_MECH_DELIVER	Authentication Mechanism Delivery Request	0x1B	SECALGREQ
CON_EN_DEC_DELIVER	Contents Encryption/Decryption Algorithm	0x1C	SECALGREQ
GK_EN_DEC_DELIVER	Group Key Encryption/Decryption Algorithm Delivery Request	0x1D	SECALGREQ
AUTH_ALG_DELIVER	Hash/MAC Algorithm Delivery Request	0x1E	SECALGREQ
SEC_RETURN	Security Return	0x1F	SECAGANS
KEY_INFO	Key Information	0x20	RELANS KEYDELIVER
ACL_LIST	Access Control List	0x21	HRSANS

Table 37 lists the sub-control types and their corresponding code values.

Table 37 – Sub-control types for secure RMCP-2

Sub-control Type	Meaning	Code Value (hexadecimal)	Message types containing the control type
KEY_MATERIAL	Key material to generate the key	0x01	RELANS KEYDELIVER
ACL_DATA	ACL_list	0x02	HRSANS

12.4 Code values related to the RMCP-2 security policy

Table 38 lists the EN_DEC_ID, CON_EN_DEC_ID and GK_EN_DEC_ID codes for the security policy.

Table 38 – EN_DEC_ID, CON_EN_DEC_ID and GK_EN_DEC_ID codes

Code	Meaning	Reference
0x01	AES CBC Mode 128-bit key	ISO/IEC 18033-3:2005
0x02	AES CTR Mode 128-bit key	ISO/IEC 18033-4:2005
0x03	PKCS #1	ISO/IEC 18033-2:2006
0x04	The SEED Encryption Algorithm	ISO/IEC 18033-3:2005
1x01	Values greater than 1x00 are reserved for other modes of AES and SEED defined by the SM	ISO/IEC 18033-3:2005
1x02		
1x03		

NOTE – EN_DEC_ID, CON_EN_DEC_ID and GK_EN_DEC_ID are located in separate fields of the secure RMCP-2 messages. Although the values for the EN_DEC_ID, CON_EN_DEC_ID and GK_EN_DEC_ID parameters may differ, the meaning of each code, as listed above, is identical wherever it is used.

Table 39 lists the AUTH_ID codes for the security policy.

Table 39 – AUTH_ID codes

Code	Acronym	Meaning	Reference
0x01	HMAC-SHA1	Hash Message Authentication Code – US Secure Hash Algorithm 1	ISO/IEC 9797-2
0x02	HMAC-MD5	Hash Message Authentication Code – Message-Digest Algorithm 5	ISO/IEC 9797-2
0x03	MD5	Message-Digest Algorithm 5	ISO/IEC 9797-2

Table 40 lists the GP_ATTRIBUTE codes for the security policy.

Table 40 – GP_ATTRIBUTE codes

Code	Attribute	Meaning
0x01	OPEN	A service user identifier is not required by an RMA before subscribing to the secure RMCP-2 session
0x02	CLOSED	A service user identifier is required by an RMA before subscribing to the secure RMCP-2 session (see 10.1.1.5)

Table 41 lists the GK_MECHA codes for the RMCP-2 security policy.

Table 41 – GK_MECHA Codes

Code	Attribute	Meaning
0x00	STATIC	Only one Group Key is used per one session
0x01	PERIODIC	Group Key is updated periodically
0x02	BACKWARD	Group Key is updated whenever any member leaves the group
0x04	FORWARD	Group Key is updated whenever any member joins the group
0x03	PERIODIC+BACKWARD	
0x05	PERIODIC+FORWARD	
0x06	BACKWARD+FORWARD	
0x07	PERIOIDC+FORWARD+BACKWARD	

Table 42 lists the GK_NAME codes for the RMCP-2 security policy.

Table 42 – GK_NAME codes

Code	Acronym	Meaning	Reference
0x01	KDC	Group Key Management Protocol (GKMP) Architecture	IETF RFC 2094
0x02	GKMP	Group Key Management Protocol (GKMP) Specification	IETF RFC 2093
0x03	MIKEY	Multimedia Internet KEYing	IETF RFC 3830
0x04	GSAKMP	Group Secure Association Key Management Protocol	IETF RFC 4535
0x05	LKH	Key Management for Multicast: Issues and Architectures	IETF RFC 2627

Table 43 shows the AUTH_ATTRIBUTE code for the RMCP-2 security policy.

Table 43 – AUTH_ATTRIBUTE code

Code	Value	Meaning
0x01	MEMBERSHIP	Membership of the session is authenticated using the Membership Authentication procedure defined in Annex E

Table 44 shows the AUTH_NAME code for the RMCP-2 security policy.

Table 44 – AUTH_NAME code

Code	Acronym	Meaning	Reference
0x01	MEM_AUTH	Membership authentication	The procedure is defined in Annex E

12.5 Miscellaneous code values

Table 45 lists two additional result codes that record reasons for rejecting the subscription of an RMA due to a missing or unrecognized SERV_USER_ID in the SUBSREQ message in cases where the session supports closed groups. These result codes are specific to the secure RMCP-2 protocol, and they supplement the code values in Table 25 that are also used in the secure RMCP-2 protocol.

Table 45 – Additional result codes for the RMCP-2 return value

Result code	Meaning
0x41	SERV_USER_ID missing
0x42	SERV_USER_ID not recognized

Table 46 lists the SEC_RETURN and Auth_result codes for the RMCP-2 security policy.

Table 46 – SEC_RETURN and Auth_result codes

Code	Value	Meaning
0x01	OK	Authentication satisfactory
0x02	ERROR	Error found on authentication
0x03	RETRANSMISSION_REQ	Retransmission Requested
0x04	FAIDED CONFIGURATION	Applies only to SEC_RETURN in the SECAGANS message

Table 47 lists the KEY_TYPE codes for key delivery.

Table 47 – KEY_TYPE codes

Code	Value	Meaning
0x01	Ks	Session Key
0x02	Kg	Group Key
0x03	Kc	Contents Encryption Key

Annex A

Tree configuration algorithm

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

A.1 Bootstrapping rule

An MA that joins an RMCP-2 session for the first time should retrieve bootstrapping information from the SM to attach to the existing tree. Because none of the MAs has any information about the tree, each MA needs to gather information about the existing tree. To independently construct the RMCP-2 tree, the SM gives the bootstrapping information to the newly joined MAs. Hence, the bootstrapping information that is managed by the SM should be as reliable and optimized as possible. The bootstrapping information basically consists of a series of MA lists managed by the SM. Because the amount of bootstrapping information is limited, the information cannot list all members. Rather, the limited information should include only the most optimized to describe the session.

Among the MAs acquired from bootstrapping information, the most optimized MA will be a MA with high forwarding capability, short network delays and high possibility of successful attachment. However, the SM cannot tell the exact network distance between MAs, SM only gives information about MA's capabilities for pre-configured network speed and space for downstream. In an RMCP-2 session, MAs are listed in the following order of preference:

1) Dedicated MA;
2) MA having lower tree depth;
3) MA having higher bandwidth.

In addition, each MA should know how many downstream nodes are allowed.

1) Available Room for new CMA.

In view of all these considerations the bootstrapping information, which contains a list of candidate MA parents, should be managed by the SM as follows:

```
if it is dedicated MA
    give highest priority


else
    priority = available number of CMA * pw_cma                               +
                 possible_forwarding_bandwidth * pw_bandwidth                 +
                 diff_hop_rate * pw_hop


pw_cma = policy based weight factor for cma                    (%/cma)
pw_bandwidth = policy based weight factor for bandwidth                (%/bit/s)
pw_hop = policy based weight factor for hop                    (%/level)
```

If all the dedicated MAs in a session have enough room for downstream MAs, or if the network administrator wants to keep every MA leaf of the MA, the SM only sends information on the DMAs. In addition, the SM should guarantee that all the MAs that appear in this information are alive.

To ensure that the list of MAs is up to date, the SM periodically checks the status of the MAs and uses the following rule to keep the status information up to date.

```
if when MA_LIST_PROB timer expires
    probe and update MA's status listed in MA_LIST

if when there is a successful subscription
    probe and update the status of the successfully subscribed MA
```

If the size of an RMCP-2 session is quite small or an SM wants to tightly control a session, the SM gives a complete list of MAs to every new MA.

```
if RMCP-2 session is tightly controlled by SM
else if the number of MA_LIST is less than the maximum size of one SUBSANS msg
        send all MA_list in SM's database
```

A.2 Neighbour discovering rule

Because the bootstrapping information from the SM is only a portion of the whole RMCP-2 session, the information is insufficient for each MA to find its best PMA. In addition, the MA cannot recognize its nearest neighbours. Thus, each MA, regardless of whether it has already attached itself to the session, should explore its neighbours by exchanging their NLs. This mechanism also enables the MA to measure the network distance and the status of each MA.

The NL used for neighbour discovery is constructed as follows:

```
include DMA to the MA_LIST_FOR_ND

if the session operates based on DMA
        break;
else
        include its root_path to the MA_LIST_FOR_ND
        include its directly attached CMA list to the MA_LIST_FOR_ND
        include its probed and non-probed MA list
        until the size of PPROB message satisfies

the MA_LIST_FOR_ND is completed
```

The network condition for the two MAs that participate in the neighbour discovery can be calculated as follows:

```
delay = RTT/2
bandwidth = packet size received / (RTT/2)
```

A.3 HMA selection rule

When there are two or more MAs in a same multicast-enabled area, HMA contention problem may occur. In case of HMA contention, each MA tries to send HANNOUNCE message to become a new HMA so every MA in the same multicast-enabled area may have duplicate HANNOUNCE messages from different MAs. The following rule is used to detect HMA contention.

```
if duplicated HANNOUNCE of valid Auth and same SID arrives from different MAID
    decide HANNOUNCE is collided
```

The following rule solves the problem of any HMA contention:

```
if they have different session join time
        choose earlier session joiner as HMA
else
    choose lower MAID as HMA
```

A.4 CMA acceptance rule

Upon receiving a new RELREQ from an MA, a PMA should decide whether to accept the relay request. The decision rule is as follows:

```
new RELAY request has arrived

if it has enough room for new CMA
        if Matched QoS && policy &&Matched data profile && data condition
            accept the MA's relay request
else
            deny the MA's relay request
```

A.5 Parent decision rule

Each MA, including the new MAs, should select from among the probed MAs the MA that has the minimum cost. The selected MA then becomes a PMA candidate. Whenever an MA joins an RMCP-2 session for the first time, the MA regards the PMA candidate as its PMA; otherwise, the candidate PMA is reserved for the parent switching. The rule for calculating the cost and for selecting the best PMA is expressed as follows:

```
if there is a MA in the same multicast-enabled area
    if the MA is in the same local LAN
        select the MA as its candidate PMA

else
    find the MA having minimum cost
    cost = diff_delay_rate          * wt_delay +
        diff_bandwidth_rate         * w_bandwidth +
        diff_hop_rate               * w_hop

if there are two or more candidate PMAs having the same cost
    select the node which has the minimum difference between the two MAIDs

*)  sum(wt_delay, w_bandwidth, w_hop) = 1
    w_delay         = weight factor for delay
    w_bandwidth     = weight factor for bandwidth
    w_hop           = weight factor for tree depth
```

The cost of selecting the PMA can be calculated as follows: RMCP-2 uses a weighing factor to configure the most optimized data delivery tree, and the weighing factor should be given by a network administrator or the session creator. The following information of MA A and MA B is assumed to have been acquired by MA C:

	MA A	**MA B**
Delay	10 ms	11 ms
Bandwidth	100 Mbit/s	90 Mbit/s
tree depth	level 5	level 7

With information on these measurements, the MA C can distinguish which MA is closer to itself. The following examples show how the MA C calculates the cost on the basis of the weighing factor:

	Case 1	**Case 2**	**Case 3**
Comparison of MA A and MA B	cost = (10-11)/E(10,11)*0.5 + (100–90)/E(100,90)*0.4 + (5-7)/E(5,7) * 0.1 = **–0.039**	cost = (10-11)/E(10,11)* 0.4 + (100–90)/E(100,90)*0.4 + (5-7)/E(5,7) * 0.2 = **–0.063**	cost = (10-11)/E(10,11)*0.4 + (100–90)/E(100,90)*0.6 + (5-7)/E(5,7) * 0.0 = **0.025**
Decision	Choose MA A	Choose MA A	Choose MA B
Case 1) weighing factor (w_delay/w_bandwidth/w_hop) = (0.5/0.4/0.1) Case 2) weighing factor (w_delay/w_bandwidth/w_hop) = (0.4/0.4/0.2) Case 3) weighing factor (w_delay/w_bandwidth/w_hop) = (0.4/0.6/0.0)			

When the cost of two PMA candidates is equal, the MA uses the following rule to choose one of the two candidates:

```
if there are two or more candidate PMAs having the same cost
    select the node which has the minimum difference between the two MAIDs
```

A.6 Tree improvement rule

Because each MA cannot know the exact information on the entire network topology, the MA's parent decision may not be most optimized. Therefore, each MA should gradually enhance the RMCP-2 session with respect to the parent switching mechanism. The following rule calculates when the parent switching is triggered:

```
If | (perceived QoS – new QoS) / perceived QoS | > Stability (policy)
        trigger parent_switching

*)  Stability factor is given by administrator when the session is created.
    Stability = 0 ~ 100% (the larger stability factor, the less parent switching)
```

A.7 PMA's kicking-out rule

The PMA can expel one of its CMAs whenever the number of CMAs allowed by an MA decreases or whenever a CMA of the PMA makes trouble. The PMA uses the following rule when making an expulsion decision:

```
if(maximum # of CMA < current # of CMA)
    select the worst CMA, send LEAVREQ to the CMA

else if(relaying QoS degraded by CMA)
    send LEAVREQ to the CMA
```

Annex B

Real-time data delivery scheme

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

B.1 Overview

Whenever an MA needs to transfer real-time data to multiple users, it adopts an IP-IP tunnelling scheme for high throughput. This subclause describes how the IP tunnelling method is used for real-time data delivery. Figure B.1 shows the general architecture of IP tunnelling.

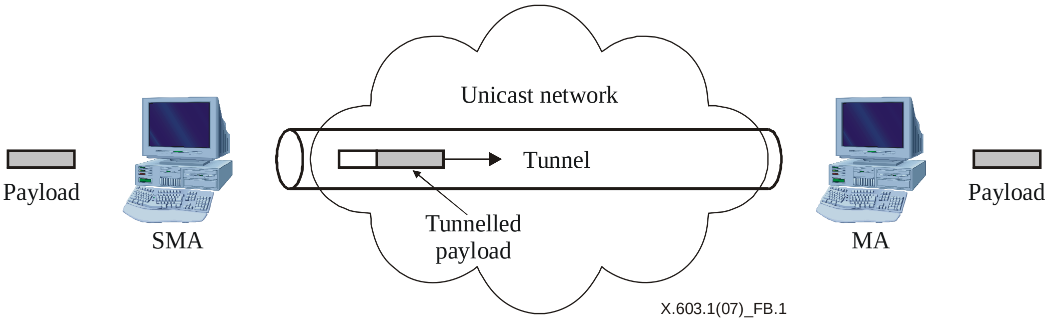


Figure B.1 – IP tunnelling scheme

B.2 IP-IP tunnel mechanism for RMCP-2 real-time data delivery

After exchanging a series of RMCP-2 control messages, a multicast data delivery path is constructed over the control path. The MA constructs the data delivery path to its subordinate MAs. The control module provides a data module with the IP address of subordinate MAs and an encapsulation scheme, and this information is contained in the data profile of the SUBSREQ message for the construction of the data delivery table. The data module of each MA stores the address of subordinate MAs in the delivery table. In this method, a real-time data delivery channel between PMA and CMA gradually constructs real-time data delivery channel from SMA to each leaf MAs. After the data delivery channel is set up, the group application operates as if it belongs to the IP multicast network, as shown in Figure B.2.

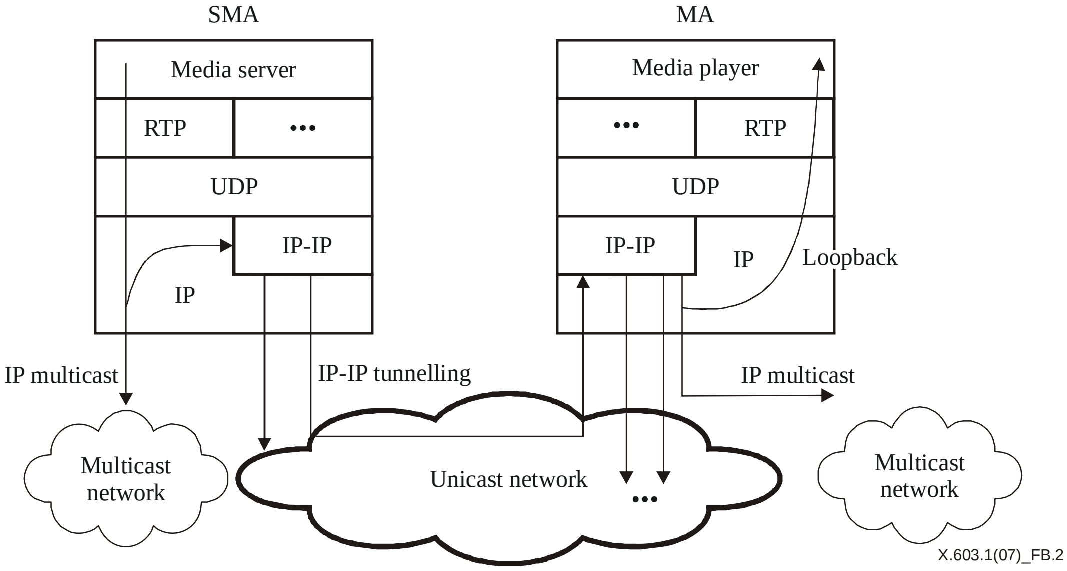


Figure B.2 – Real-time data channel with IP-IP encapsulation

The SMA encapsulates the IP multicast data packets in the unicast data packets and transmits the encapsulated data to the downstream MAs by unicasting them over the unicast network. The SMA also multicasts the IP multicast data packets to a multicast-enabled area. Upon receiving the tunnelled data packets, each CMA de-capsulates the packets into the IP multicast data packets.

When CMAs are in the same multicast region, a PMA simply forwards the multicast data to the multicast region. If one or more CMAs are in the unicast region, a PMA should encapsulate the IP multicast data packets and then transmits the tunnelled data to its CMAs.

Annex C

Reliable data delivery scheme

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

C.1 Overview

The scheme described here is an overlay multicast data delivery scheme that can handle reliable data. The nodes of the parent-child relationship exchange data profiles to find a set of available data. To make it feasible, each node opens the TCP connection for reliable data delivery. Once the data delivery channel is set, each node receives data from its parent and then forwards the received data to its downstream, if any. In this manner, the data from the root can reach the leaf nodes via multiple intermediate nodes.

By using data profiles, each node can search for a node with the necessary data, and, if necessary, any node that uses this scheme can change its upstream node. The following subclause describes the protocol sequence of the overlay multicast scheme for simplex reliable data delivery.

C.2 Operation

C.2.1 Channel connection

Figure C.1 shows the procedure of channel connection and follows the following four steps:

1) The CMA sends the PMA a data profile that contains the new joiner's local address and the sequence number to receive. When the new joiner has no information on the data delivery, the sequence number to receiver will be set to NEWEST.
2) After receiving the CMA's data profile, the PMA replies with data profile which contains the listening address and current sequence number.
3) Two MAs, which exchange channel information, establish TCP connections between them and then allocate a channel ID for the established connection.
4) The PMA sends encapsulated user data with the channel ID, which is allocated by the PMA itself, as well as the sequence number.

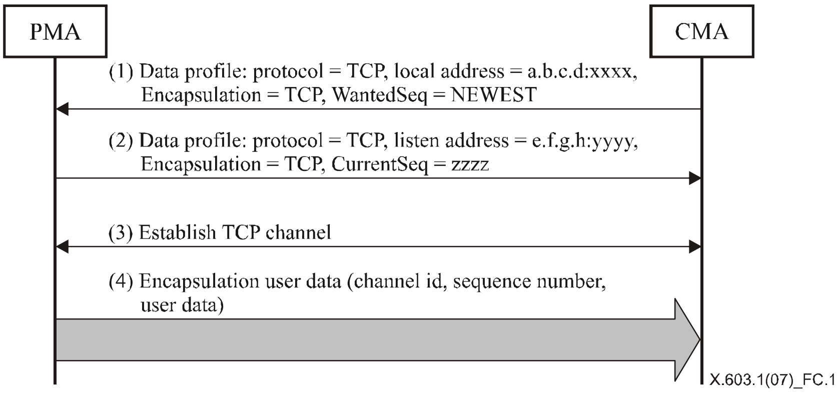

PMA
CMA
(1) Data profile: protocol = TCP, local address = a.b.c.d:xxxx,
Encapsulation = TCP, WantedSeq = NEWEST
(2) Data profile: protocol = TCP, listen address = e.f.g.h:yyyy,
Encapsulation = TCP, CurrentSeq = zzzz
(3) Establish TCP channel
(4) Encapsulation user data (channel id, sequence number,
user data)
X.603.1(07)_FC.1

Figure C.1 – Procedure for channel connection

C.2.2 Channel disconnection

As shown in Figure C.2, the procedure for disconnecting channels has the following three steps:

1) A TCP connection is established between two MAs.
2) The PMA sends encapsulated user data with the channel ID, which is allocated by the PMA itself, along with the sequence number.
3) Either MA can eliminate the TCP channel by calling for the TCP to be closed.

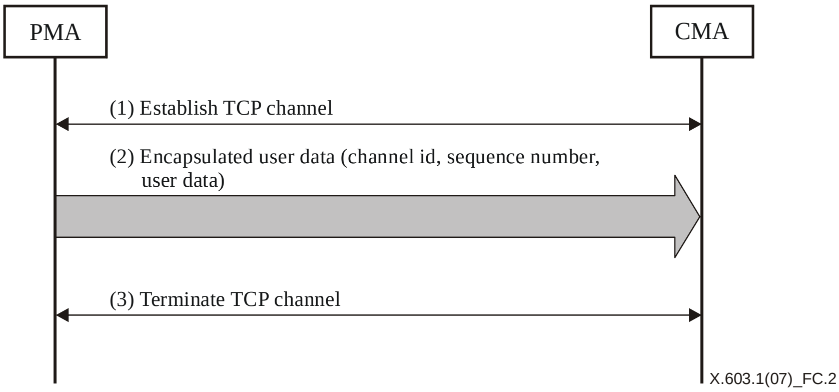

PMA
CMA
(1) Establish TCP channel
(2) Encapsulated user data (channel id, sequence number, user data)
(3) Terminate TCP channel
X.603.1(07)_FC.2

Figure C.2 – Procedure for channel disconnection

C.2.3 Channel switching

As shown in Figure C.3, the procedure for channel switching has the following seven steps:

1) A TCP connection is established between two MAs.

2) The PMA sends encapsulated data with the channel ID, which is allocated by the PMA itself, along with the sequence number and user data.

3) The CMA sends a data profile, which contains its local address and sequence number to receive, to the new PMA.

4) Upon receiving data profile from a new CMA, the new PMA replies with data profile which contains listening address, current sequence number and buffered sequence number.

5) If the wanted sequence is between the buffered sequence number and the current sequence number, the CMA disconnects the TCP channel with an old PMA.

6) The new PMA and CMA, which are connected by TCP, allocate new channel identification for the connection.

7) The new PMA sends encapsulated user data with the channel id allocated by itself and sequence number from the wanted sequence number.

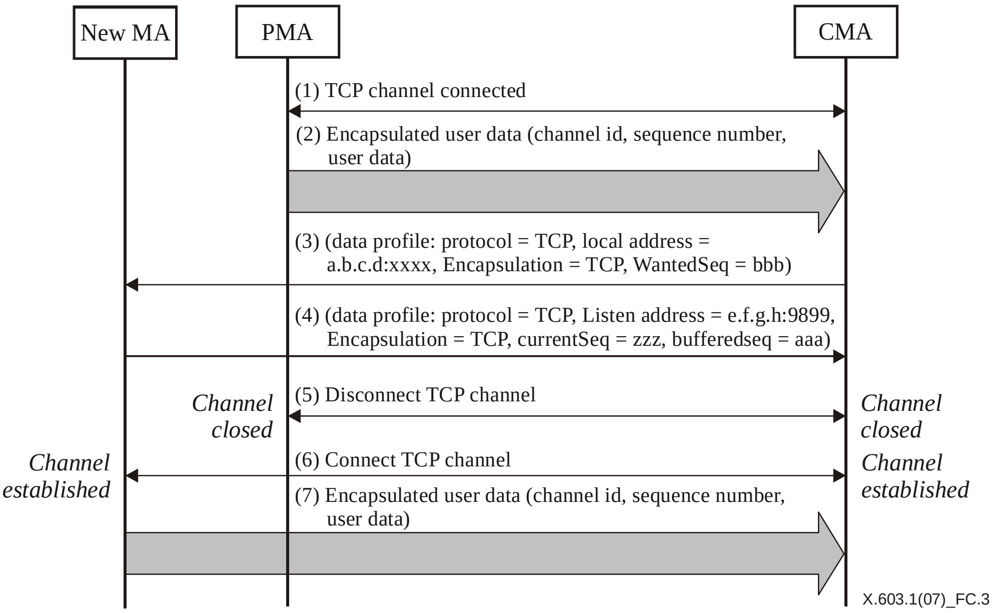

New MA
PMA
CMA
(1) TCP channel connected
(2) Encapsulated user data (channel id, sequence number, user data)
(3) (data profile: protocol = TCP, local address = a.b.c.d:xxxx, Encapsulation = TCP, WantedSeq = bbb)
(4) (data profile: protocol = TCP, Listen address = e.f.g.h:9899, Encapsulation = TCP, currentSeq = zzz, bufferedseq = aaa)
Channel closed
(5) Disconnect TCP channel
Channel closed
Channel established
(6) Connect TCP channel
Channel established
(7) Encapsulated user data (channel id, sequence number, user data)
X.603.1(07)_FC.3

Figure C.3 – Procedure for channel switching

C.3 Data encapsulation format

Figure C.4 shows the user data message for reliable data delivery:

a) Reserved: Reserved for future use and set to zero now.

b) Length: Total byte length of current message.

c) Channel ID: Identification of the data channel between the data hops.

d) Sequence number: The sequence number allocated by an SMA of the current service data unit; this value can be allocated globally in a round robin manner.

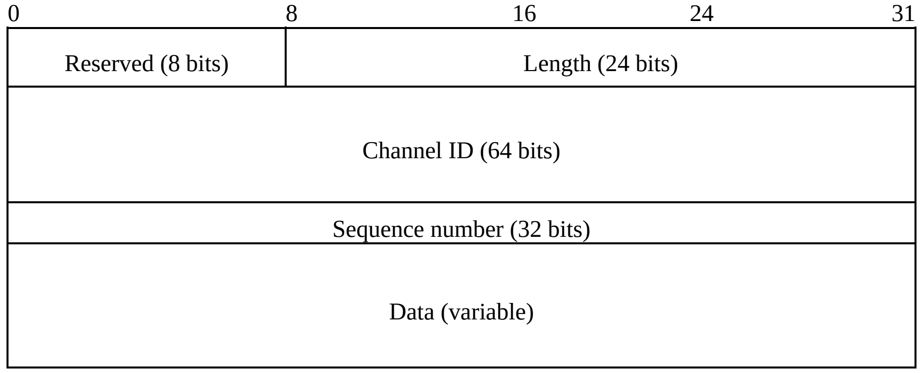


Figure C.4 – Data encapsulation format

C.4 Data profile

When the data tunnelling scheme is used in RMCP-2, the following format should be used for the data profile:

"Protocol = TCP, Listen address = a.b.c.d:9899, Encapsulation= TCP, CurrentSeq=xxxx, BufferedSeq=yyyy, WantedSeq=zzz"

Annex D

RMCP-2 API

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

This annex specifies the application programming interfaces (APIs) for RMCP-2. The APIs described in this annex can be used in applications that utilize the capabilities of RMCP-2.

The RMCP-2 APIs follow the Berkeley socket APIs. However, to differentiate the RMCP-2 APIs from existing Berkeley socket functions, the RMCP-2 APIs are prefixed with 'rmcp2_' (for example, rmcp2_socket).

D.1 Overview

D.1.1 APIs

Table D.1 summarizes the API functions in RMCP-2:

Table D.1 – Summary of RMCP-2 APIs

Category	Name	Description
MA control	*rmcp2_socket()*	Creates a new RMCP-2 socket.
	rmcp2_bind()	Associates a set of information about session, such as session id, role, local and group addresses, data profile, etc.
	rmcp2_connect()	Joins RMCP-2 session.
	rmcp2_close()	Terminates connection and releases socket.
	rmcp2_setsockopt()	Sets socket and protocol options to RMCP-2 MA control module.
	rmcp2_getsockopt()	Gets socket and protocol options from RMCP-2 MA control module.
	rmcp2_recv()	Delivers received data to application.
	rmcp2_send()	Sends application data to a RMCP-2 group.
Data delivery	*rmcp2_recv()*	Delivers received data to application.
	rmcp2_send()	Sends application data to a RMCP-2 group.
Session management	*rmcp2_session_open()*	Creates a new RMCP-2 session.
	rmcp2_session_close()	Terminates RMCP-2 session and releases resources allocated.
	rmcp2_member_out()	Kicks the trouble maker out from the session.
	rmcp2_status_report()	Examines the condition of a specific RMCP-2 session.
	rmcp2_char_change()	Sets or changes RMCP-2 session characteristics.

D.1.2 Use of RMCP-2 API

Figure D.1 illustrates the use of RMCP-2 APIs and shows API sequences in terms of the SM and two MAs.

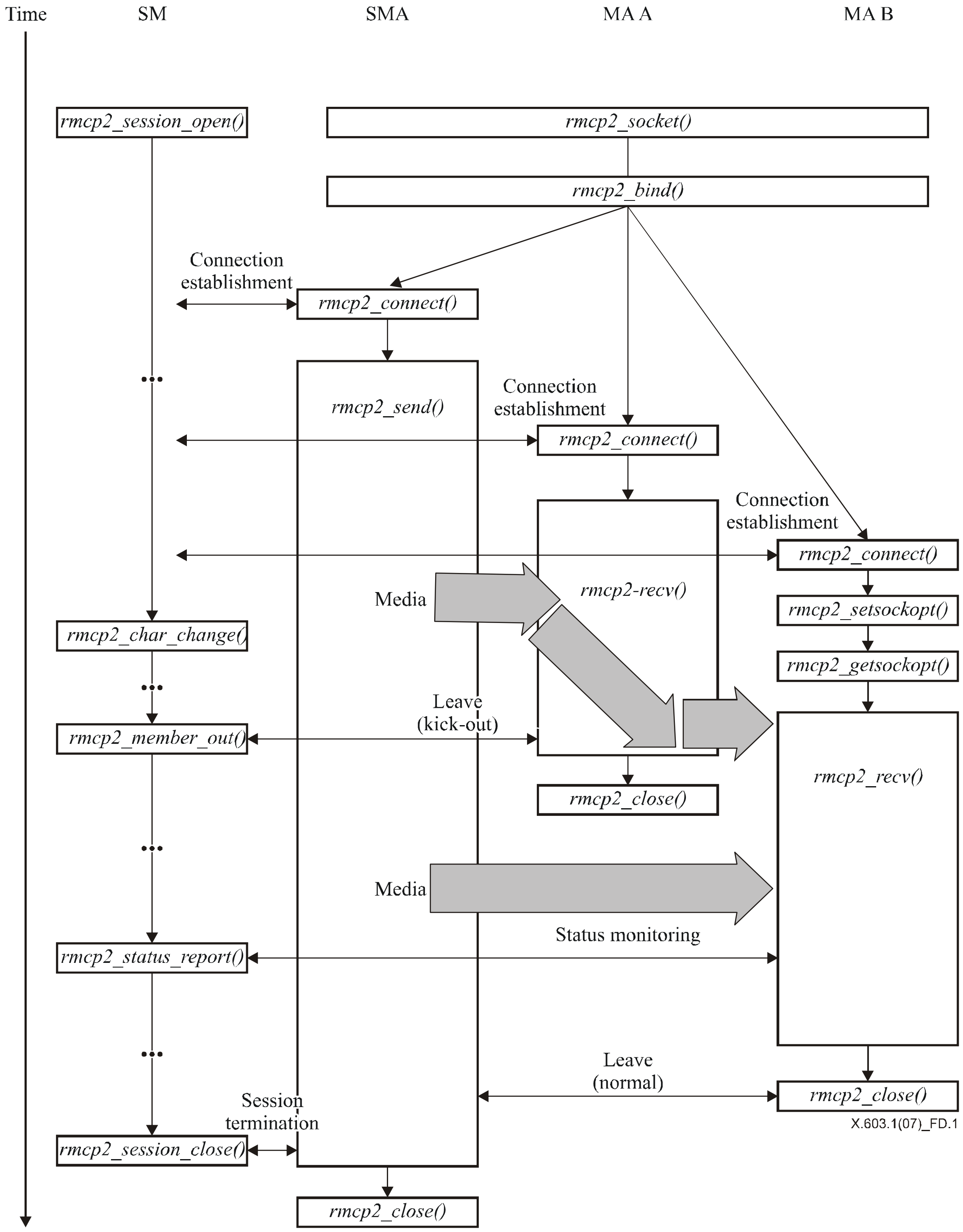


Figure D.1 – Usage of RMCP-2 APIs

D.2 RMCP-2 API functions

D.2.1 Functions related to MA control

This subclause defines a series of RMCP-2 APIs that are related to the MA. RMCP-2 applications use the functions defined here to join and leave RMCP-2 sessions. The functions to send and receive data are defined in the next subclause.

int rmcp2_socket (void)

Under the RMCP-2 protocol, an application asks an RMCP-2 MA to start an RMCP-2 session by calling the *rmcp2_socket()* function. If successful, this function returns a non-zero RMCP-2 socket identifier; otherwise, it returns a negative value with error codes.

*int rmcp2_bind(int sd, session_profile *profile, int profile_len)*

Whenever an RMCP-2 application wants to impose some information about a session, it can call the *rmcp2_bind()* function. This function sets the MA information that is crucial for joining an RMCP-2 session. Some of the most important details of a session are as follows:

a) session ID;

b) its role inside RMCP-2 session, such as sender-side MA, leaf MA;

c) a specific MA address to be operated;

d) a group address;

e) data profile which it wants to use;

f) other vendor-specific information, etc.

After a successful bind, this function returns zero; otherwise, it returns a negative return value with the proper error code.

*int rmcp2_connect(int sd, struct sockaddr *sm_addr, int addrlen)*

Only after a bind is successful, an RMCP-2 application can start to join an RMCP-2 session. By calling the *rmcp2_connect()* function, each RMCP-2 application can invoke the MA to subscribe to and join an RMCP-2 session. The arguments associated with this function are the address of the SM and the session ID to join an RMCP-2 session. After successfully joining a session, this function returns zero; otherwise, it returns a negative value with an error code.

int rmcp2_close(int sd)

To leave a session, an RMCP-2 application calls the *rmcp2_close()* function. By calling the *rmcp2_close()* function, an application can make an RMCP-2 MA initiate a departure procedure and then free the protocol control block. After successful session leave, this function returns a zero; otherwise, it returns a negative integer with the appropriate error code.

*int rmcp2_setsockopt(int sd, int opt_type, char *opt, int optlen)*

The *rmcp2_setsockopt()* function enables an application to set or change one or more protocol parameters. If it is successful, this function returns zero; otherwise, it returns a negative integer with the appropriate error code.

*int rmcp2_getsockopt(int sd, int opt_type, char *opt, int *optlen)*

An application that wants to know one or more protocol parameters from the MA calls the *rmcp2_getsockopt()* function by offering an *opt_type* and an empty **opt* which is large enough to hold the resultant information from the MA. If it is successful, this function returns zero; otherwise, it returns a negative integer with the appropriate error code.

D.2.2 Functions related to MA's data delivery

This subclause defines a series of RMCP-2 APIs that are related to RMCP-2 data delivery. These APIs are used by applications to send or receive RMCP-2 data traffic.

*int rmcp2_recv(int sd, char *buf, int len, int flags)*

A receiving application that wants to receive data from an RMCP-2 session calls the *rmcp2_recv()* function and copies the received data of *len* from the MA data module. If it is successful, this function returns zero; otherwise, it returns a negative value with an error code.

*int rmcp2_send(int sd, char *buf, int len, int flags)*

To send data to an RMCP-2 session, an RMCP-2 application calls the *rmcp2_send()* function. However, because RMCP-2 only supports a one-to-many data delivery service, the MA of the sending application must be an SMA. This function copies the data of *len* to the MA data module. If it is successful, this function returns the number of bytes that it sends; otherwise, it returns a negative value with an error code.

D.2.3 Functions related to session management

A session manager application (SM application) can initiate, manage or terminate an RMCP-2 session by calling one of the APIs defined in this subclause. To clarify that ambiguity between *an application that uses the RMCP-2 SM* and the *RMCP-2 SM* itself, the term *SM application* is used to refer to the application that uses the RMCP-2 SM.

*SID rmcp2_session_open(session_profile *session_profile)*

An SM application that wants the SM to start an RMCP-2 session calls the *session_open()* function with session profile. The *session_profile* argument should be packed with sufficient session information to create and manage an RMCP-2 session. Upon receiving the session profile, the SM allocates enough room for a specific RMCP-2 session. After the successful creation of a session, this function returns the created session ID; otherwise, it returns zero with an appropriate error code.

int rmcp2_session_close(SID session_id)

An SM application that wants the SM to terminate an RMCP-2 session calls the *session_close()* API. This function asks the SM to start the procedure for terminating an RMCP-2 session and then frees enough room for the RMCP-2 session. After the session has been successfully terminated, this function returns a non-negative value; otherwise, it returns a negative value with an error code.

int rmcp2_member_out(SID session_id, MAID maid)

Whenever a session member, or MA, causes critical problems or violates session policy, the SM application may expel the trouble maker from the session. If an SM application wants an SM to expel a specific member from the session, then it calls the *member_out()* API with a session ID, along with the ID of the member to be expelled.

*int rmcp2_status_report(SID session_id, int command, char *result, int *result_len)*

The *status_report()* function enables an SM application to examine the condition of a session. This function is usually called with arguments such as the session ID, the operation commands and buffer for the results. If it is successful, this function returns zero; otherwise, it returns a negative value and an error code.

*int rmcp2_char_change(SID session_id, int command, char *opt, int optlen)*

The *rmcp2_ char _change setsockopt()* function enables an SM application to set or change the characteristics of an RMCP-2 session, such as the AUTH information. If it is successful, this function returns zero; otherwise, it returns a negative integer with an appropriate error code.

Annex E

Membership authentication mechanism

(This annex forms an integral part of this Recommendation | International Standard)

E.1 Overview

The secure RMCP-2 membership authentication is based on the three-pass authentication procedure in ISO/IEC 9798-3:1998. This procedure, as applied to secure RMCP-2, is described below and is illustrated in Figures E.1 and E.2. The variables used are listed in Table E.1.

Membership authentication checks whether a node is a session member; it plays the role of a member of the RMCP tree or local group of the MM region and assumes that any node trying to authenticate membership for the RMCP tree or the group is verified by SM in the RMCP-2 session in advance, since the procedure is executed based on the password information of the node. To configure the RMCP tree, PMA and CMA perform this procedure when CMA wants to be a child node of PMA. Likewise, DMA and RMA authenticate their counterparts to transmit multicast data to the regular members joining the MM group.

E.2 Authentication procedure

The membership authentication is initiated on a RELREQ message containing an AUTH control in the RELREQ message (see 11.2.3). PMA and DMA can be servers, and CMA and RMA, client parties. The client requests the server to authenticate a membership using some authentication materials: identifier (IDc), random number (r_c), and encrypted value by hashed 'auth' ($E_k(g^A)$ mod p). The server then sends its authentication materials: IDs, r_s, $Ek(g^B)$ mod p, and Vs(Vector value). Finally, authentication is finished successfully when the client sends vector value Vc. The authentication procedure is based on the Diffie Hellman algorithm. Here, A and B are arbitrary values, and K_{MAS} as a shared key between the client and the server encrypts Kg in the local group of the MM region. Here, the random number *r* should be securely generated on cryptographically secure pseudo random bit generator (CGSPRBG) such as PKCS#1, Micali−Schnorr and Blum−Blum−Shub pseudo random bit generators. 'm_auth' is value created by a message digest algorithm for message integrity. The value is made by symmetrical or asymmetrical secure MAC functions. 'auth' is AUTH-information of the SUBSREQ message.

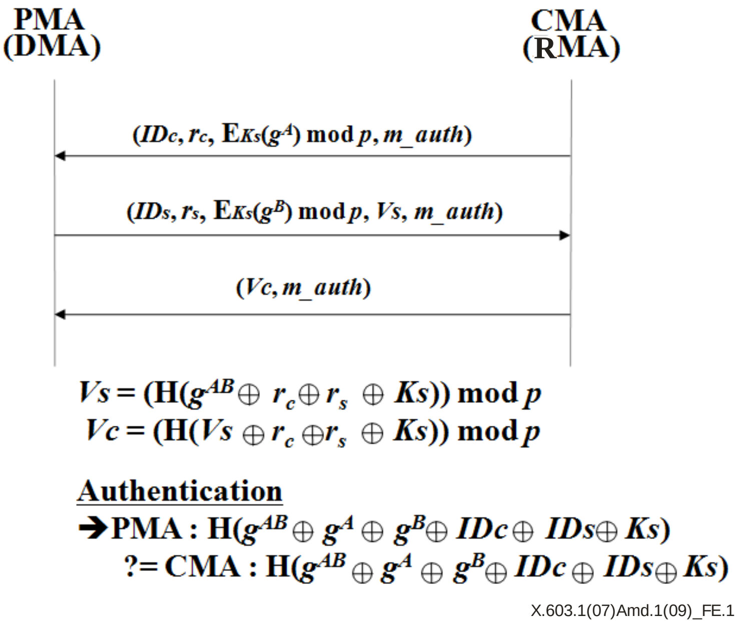
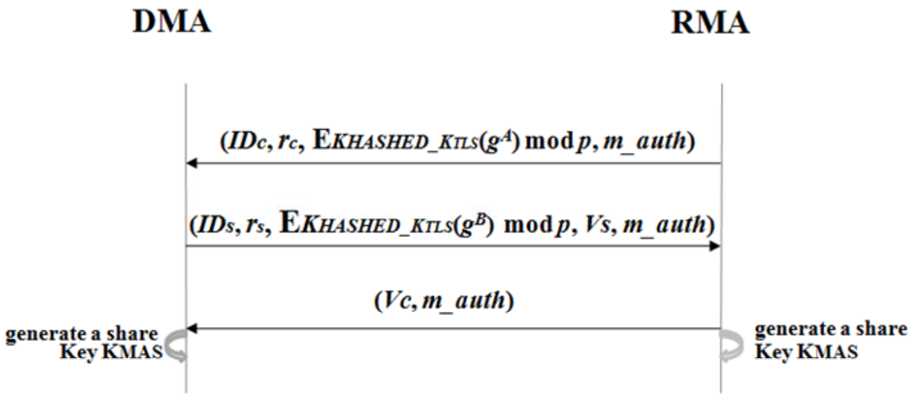

PMA (DMA)
CMA (RMA)
$(IDc, rc, E_{Ks}(g^A) \bmod p, m\_auth)$
$(IDs, rs, E_{Ks}(g^B) \bmod p, Vs, m\_auth)$
(Vc, m_auth)
$Vs = (H(g^{AB} \oplus r_c \oplus r_s \oplus Ks)) \bmod p$
$Vc = (H(Vs \oplus r_c \oplus r_s \oplus Ks)) \bmod p$
Authentication
$\rightarrow$PMA : $H(g^{AB} \oplus g^A \oplus g^B \oplus IDc \oplus IDs \oplus Ks)$
?= CMA : $H(g^{AB} \oplus g^A \oplus g^B \oplus IDc \oplus IDs \oplus Ks)$
X.603.1(07)Amd.1(09)_FE.1

Figure E.1 – Membership authentication between PMA and CMA


DMA
RMA
(IDc, rc, EKHASHED_KTLS(gA) mod p, m_auth)
(IDs, rs, EKHASHED_KTLS(gB) mod p, Vs, m_auth)
(Vc, m_auth)
generate a share Key KMAS
generate a share Key KMAS

$$Vs = (\mathrm{H}(g^{AB} \oplus r_c \oplus r_s \oplus Ks)) \bmod p$$

$$Vc = (\mathrm{H}(Vs \oplus r_c \oplus r_s \oplus Ks)) \bmod p$$

$$K_{MAS} = \mathrm{H}(g^{AB} \oplus g^A \oplus g^B \oplus IDc \oplus IDs \oplus K_{HASHED\_KTLS})$$

Authentication

$$\rightarrow \mathrm{DMA} : \mathrm{H}(g^{AB} \oplus g^A \oplus g^B \oplus IDc \oplus IDs \oplus K_{HASHED\_KTLS})$$

$$?= \mathrm{RMA} : \mathrm{H}(g^{AB} \oplus g^A \oplus g^B \oplus IDc \oplus IDs \oplus K_{HASHED\_KTLS})$$

Figure E.2 – Membership authentication between DMA and RMA

Table E.1 – Definition of variables for membership authentication

Variables/Functions	Definitions
E(*x*)	Encryption function on defined multicast security policy
H(*x*)	Hash function on defined AUTH_ALG of multicast security policy
Mod	Modulation operator
IDc	Identifier of client-side; CMA and RMA
IDs	Identifier of server-side; PMA and DMA
r_c	Random number from client-side; CMA and RMA
r_s	Random number from server-side; PMA and DMA
G	Generator on Diffie-Hellman algorithm
A	Arbitrary value by client-side; CMA and RMA
B	Arbitrary value by server-side; PMA and DMA
P	Defined value on Diffie-Hellman algorithm
Vs	Vector value on Diffie-Hellman algorithm from server-side; PMA and DMA
Vc	Vector value on Diffie-Hellman algorithm from client-side; CMA and RMA

Successful authentication is indicated by Auth_result with a value of 0x01 in the AUTH_ANS control of the RELANS message.

Annex F

Bibliography

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

F.1 Informative references

– IETF RFC 2093 (1997), *Group Key Management Protocol (GKMP) Specification*.

– IETF RFC 2627 (1999), *Key Management for Multicast: Issues and Architectures*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2011