FINAL
DRAFT

INTERNATIONAL STANDARD

ISO/IEC FDIS 14476-1

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2001-11-15

Voting terminates on:
2002-01-15

Information technology — Enhanced Communications Transport Protocol: Specification of simplex multicast transport

Technologies de l'information — Protocole de transport de communication amélioré: Spécifications pour le Transport «Simplex Multicast»

Please see the administrative notes on page ii-1

Reference number
ISO/IEC FDIS 14476-1:2001(E)

In accordance with the provisions of Council Resolution 21/1986, this document is **circulated in the English language only**.

Table of Contents

Figures

Tables

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 14476 may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14476-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*, in collaboration with ITU-T. The identical text is published as ITU-T Rec. X.606.

ISO/IEC 14476 consists of the following parts, under the general title *Information technology — Enhanced Communications Transport Protocol*:

— *Part 1: Specification of simplex multicast transport*

— *Part 2: Specification of QoS management for simple multicast transport*

— *Part 3: Specification of duplex multicast transport*

— *Part 4: Specification of QoS management for duplex multicast transport*

— *Part 5: Specification of n-plex multicast transport*

— *Part 6: Specification of QoS management for n-plex multicast transport*

Annexes A and B of this part of ISO/IEC 14476 are for information only.

Summary

This Recommendation | International Standard specifies the Enhanced Communications Transport Protocol (ECTP), which is a transport protocol designed to support Internet multicast applications running over multicast-capable networks. ECTP operates over IPv4/IPv6 networks that have the IP multicast forwarding capability with the help of IGMP and IP multicast routing protocols. ECTP could possibly be provisioned over UDP. ECTP is targeted to support tightly controlled multicast connections.

This first part of ECTP defines the protocol which provides reliability control in the simplex multicast case, adopting a tree-based approach. QoS management functions for the simplex case will be defined in part 2 of the ECTP specification. Further parts of ECTP will define reliability control and corresponding QoS management functions for the duplex case (parts 3 and 4) and the N-plex case (parts 5 and 6).

The sender is at the heart of multicast group communications. A single sender in the simplex multicast connection is assigned the role of the connection owner. The connection owner is responsible for overall connection management by governing connection creation and termination, connection pause and resumption, and join and leave operations.

For tree-based reliability control, a hierarchical tree is configured during connection creation. The sender is the root of the control tree. The control tree can define a parent-child relationship between any pair of tree nodes. This tree-based structure can result in local owners occurring at lower levels in the tree hierarchy as the control structure extends. Each local owner created becomes the root of its own local control tree. The connection owner will then be the root of the overall control tree. Error control is performed for each local group defined by a control tree. Each parent retransmits lost data, in response to retransmission requests from its children.

Introduction

This Recommendation | International Standard specifies the Enhanced Communications Transport Protocol (ECTP), which is a transport protocol designed to support Internet multicast applications running over multicast-capable networks. ECTP operates over IPv4/IPv6 networks that have the IP multicast forwarding capability with the help of IGMP and IP multicast routing protocols, as shown in Figure 1. ECTP could possibly be provisioned over UDP.

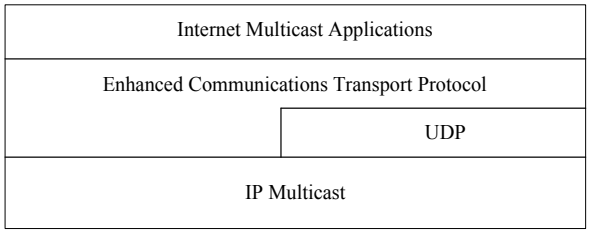


Figure 1 – ECTP Model

ECTP is designed to support tightly controlled multicast connections in simplex, duplex and N-plex applications. This part of ECTP (part 1) specifies the protocol mechanisms for reliability control in the simplex case. ECTP also provides QoS management functions for stable management of the QoS of the connection users. Such QoS management functionality can be achieved with QoS negotiation, monitoring, and maintenance operations. The protocol procedures for QoS management of the simplex case will be defined in the simplex QoS management specification (X.ectp-2 | ISO/IEC 14476-2), which forms an integral part of this Recommendation | International Standard. Further parts of the standard will define control procedures and associated QoS management functions for the duplex case (X.ectp-3 | ISO/IEC 14476-3 and X.ectp-4 | ISO/IEC 14476-4) and for the N-plex case (X.ectp-5 | ISO/IEC 14476-5 and X.ectp-6 | ISO/IEC 14476-6).

In ECTP, all prospective members are enrolled into a multicast group, before a connection or session is created. Those members define an enrolled group. Each receiver in the enrolled group is referred to as an enrolled receiver. In the enrolment process, each member will be authenticated. The group information, including group key and IP multicast addresses and port numbers, will be distributed to the enrolled members during the enrolment process. An ECTP connection is created for these enrolled group members.

ECTP is targeted for tightly controlled multicast services. The sender is at the heart of multicast group communications. A single sender in the simplex multicast connection is assigned the role of the connection owner, designated as top owner (TO) in this specification. The connection owner is responsible for overall connection management by governing connection creation and termination, connection pause and resumption, and join and leave operations.

The sender triggers the connection creation process. Some or all of the enrolled receivers will participate in the connection, becoming designated “active receivers”. Any enrolled receiver that is not active may participate in the connection as a late-joiner. An active receiver can leave the connection. After the connection is created, the sender begins to transmit multicast data. If network problems (such as severe congestion) are indicated by the ECTP QoS management functions (defined in ECTP part 2), the sender suspends multicast data transmission temporarily, invoking the connection pause operation. After a pre-specified time, the sender resumes data transmission. If all of the multicast data have been transmitted, the sender terminates the connection.

ECTP provides the reliability control mechanisms for multicast data transport. ECTP mechanisms are designed to keep congruency with those being proposed in the IETF. To address reliability control with scalability, the IETF has proposed three approaches: Tree based ACK (TRACK), Forward Error Correction (FEC), and Negative ACK Oriented Reliable Multicast (NORM). Each approach has its own pros and cons, and each service provider may take a different approach toward implementing reliability control. ECTP adopts the TRACK approach, because it is more similar to the existing TCP mechanisms and more adaptive to the ECTP framework.

For tree-based reliability control, a hierarchical tree is configured during connection creation. The sender is the root of the control tree. The control tree can define a parent-child relationship between any pair of tree nodes. This tree-based structure can result in local owners (parents) occurring at lower levels in the tree hierarchy as the control structure extends. Each local owner created becomes the root of its own local control tree. The connection owner will then be the root of the overall control tree. Error control is performed for each local group defined by a control tree. Each parent retransmits lost data, in response to retransmission requests from its children.

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – ENHANCED COMMUNICATIONS TRANSPORT PROTOCOL: SPECIFICATION OF SIMPLEX MULTICAST TRANSPORT

1. Scope

This Recommendation | International Standard specifies the Enhanced Communications Transport Protocol (ECTP), which is a transport protocol designed to support Internet multicast applications over multicast-capable IP networks.

This Recommendation | International Standard specifies the ECTP for the simplex multicast transport connection that consists of one sender and many receivers. This Recommendation | International Standard specifies the protocol procedures for the following protocol operations:

a) connection creation with tree creation;

b) multicast data transmission;

c) tree-based reliability control with error detection, retransmission request, and retransmission;

d) late join and leave;

e) tree membership maintenance; and

f) connection termination.

2. Normative references

The following ITU-T Recommendations, International Standards, and IETF standard RFCs contain provisions that, through references in the text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations, Standards, and RFCs 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 | International Standards and RFCs listed below. IEC and ISO members maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU-T maintains a list of currently valid ITU-T documents. The IETF also maintains an index list of all published RFCs.

- ITU-T Recommendation X.601 (2000), Information technology – Multi-Peer Communications Framework
- ITU-T Recommendation X.605 (1998) | ISO/IEC 13252: 1999, Information technology – Enhanced Communications Transport Service Definition

3. Definitions

3.1 Terms defined in ITU-T Rec. X.601

This Recommendation | International Standard is based on the definitions of the multicast groups developed in Multi-Peer Communications Framework (ITU-T Rec. X.601).

a) Enrolled group; and

b) Active group.

3.2 Terms defined in ITU-T Rec. X.605 | ISO/IEC 13252

This Recommendation | International Standard is based on the concepts developed in Enhanced Communications Transport Service (ITU-T Rec. X.605 | ISO/IEC 13252).

a) Transport connection; and

b) Simplex;

3.3 Terms defined in this Recommendation | International Standard

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

a) *application* – represents an Internet multicast application in this specification. It corresponds to a transport service user in the OSI mode. It exchanges transport service primitives with the corresponding transport protocol entity. In the Internet, it communicates with the transport protocol entity via a socket interface;

b) *packet* – represents an unit of transport data, which is equivalent to a segment in TCP/IP and a transport protocol data unit (TPDU) in OSI model. A transport entity communicates with another transport entity by transmitting packets. A transport protocol entity creates a packet, which is encapsulated into an IP datagram and then delivered to the destination entity over networks;

c) *sender* – represents a transport protocol entity that sends the multicast data to the receivers;

d) *receiver* – represents a transport protocol entity that receives the multicast data;

e) *tree* – is a hierarchical logical tree employed for providing scalable reliability control. A tree defines a parent-child relationship between a pair of tree nodes. Sender and receivers are organized into a tree. In the tree hierarchy, a tree node is designated as TO (Top Owner), LO (Local Owner) or LE (Leaf Entity). TO is a single ECTP sender. All the receivers are designated as LOs or LEs;

f) *TO* (*Top Owner*) – is a single sender in the ECTP simplex multicast connection. TO is the root of the tree and manages the overall protocol operations for the connection;

g) *LO* (*Local Owner*) – is a receiver that manages a local group. An LO is responsible for the overall protocol operations for its local group defined by the control tree. For error recovery, it retransmits the multicast data that have been lost by its children. For flow and congestion control, it aggregates the control information for all of its children and then delivers the aggregated information toward TO. In terms of the reliability control operations, TO is also an LO;

h) *LE* (*Leaf Entity*) – is a receiver that has not been designated as an LO. An LE cannot have any children. It is located as a leaf node on the tree;

i) *local group* – consists of a parent and its children in the tree hierarchy;

j) *parent* – is a parent node for a local group. TO or an LO can be a parent; and

k) *child* – is a child node for a local group. An LO or LE can be a child.

4. Abbreviations

4.1 Packet types

CR	Connection Creation Request
CC	Connection Creation Confirm
TJ	Tree Join Request
TC	Tree Join Confirm
DT	Data
ND	Null Data
RD	Retransmission Data
HB	Heartbeat
ACK	Acknowledgment
JR	Late Join Request
JC	Late Join Confirm
LR	Leave Request
CT	Connection Termination

4.2 Miscellaneous

ECTP	Enhanced Communications Transport Protocol
ECTS	Enhanced Communications Transport Service
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet protocol
QoS	Quality of Service
RMT	Reliable Multicast Transport
RFC	Request for Comments
SAP	Session Announcement Protocol
SDP	Session Description Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

5. Conventions

In this Recommendation | International Standard, the key words “MUST”, “REQUIRED”, “SHALL”, “MUST NOT”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “MAY”, and “OPTIONAL” are to be interpreted as described in IETF RFC 2119, and indicate requirement levels for compliant ECTP implementations. Those key words are case-sensitive.

6. Overview

The ECTP is a transport protocol designed to support Internet multicast applications. ECTP operates over IPv4/IPv6 networks that have IP multicast forwarding capability.

This Specification describes the ECTP protocol for the simplex multicast transport connection that consists of one sender and many receivers. ECTP supports the connection management functions, which are based on ITU-T Rec. X.605 | ISO/IEC 13252. The connection management functions include connection creation and termination, connection pause and resumption, and late join and leave. For reliable delivery of multicast data, ECTP also provides the protocol mechanisms for error, flow and congestion controls. To allow scalability to large-scale multicast groups, tree-based reliability control mechanisms are employed which are congruent with those being proposed in the IETF RMT WG.

Figure 2 shows an overview of the ECTP operations.

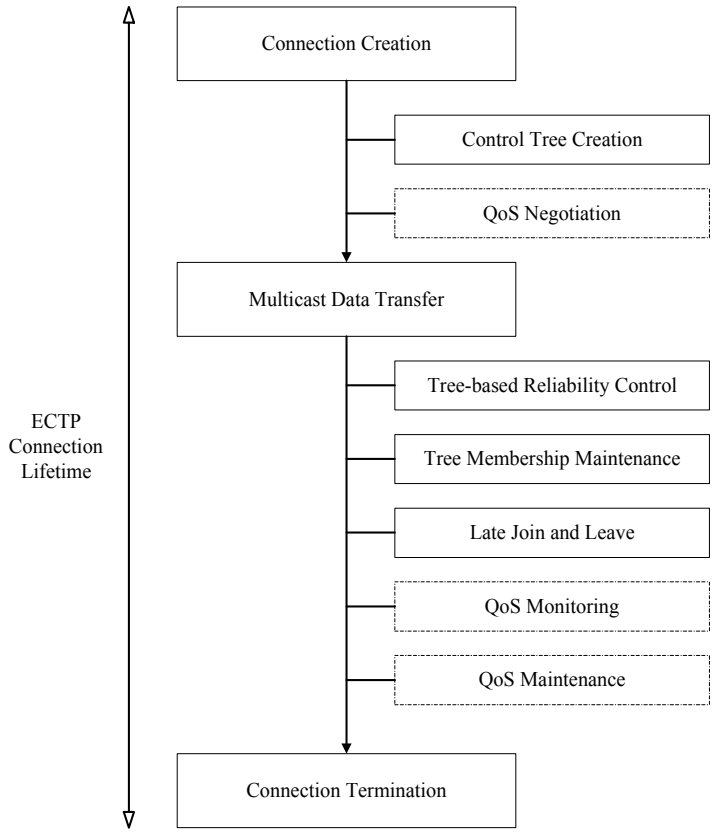


Figure 2 – ECTP Protocol Operations

As shown in the figure, the QoS management operations such as QoS negotiation, monitoring, and maintenance will be specified in part 2 of this Recommendation | International Standard. In particular, QoS maintenance includes the operations for connection pause and resume, and the flow and congestion controls.

Before an ECTP transport connection is created, the prospective receivers are enrolled into the multicast group. Such a group is called an enrolled group (see 8.1). During enrollment, authentication processes may be performed together with group key distribution. The IP multicast addresses and port numbers must be announced to the receivers. These enrollment operations may rely on the well-known SAP/SDP, HTTP (Web Page announcement) and SMTP (E-mail) protocols. The specific enrollment mechanisms are outside the scope of this specification.

An enrolled receiver will be connected to the multicast-capable network with the help of the IGMP and IP multicast routing protocols. Those IGMP and multicast routing protocols will refer to the announced multicast addresses. An ECTP transport connection is created for the enrolled receivers.

ECTP is targeted to support tightly controlled multicast connections. The ECTP sender is at the heart of the multicast group communication. The sender, designated as connection owner (TO), is responsible for the overall management of the connection by governing connection creation and termination, connection pause and resumption, and the late join and leave operations.

The ECTP sender triggers the connection creation process by sending a connection creation message. Some or all of the enrolled receivers may respond with confirmation messages to the sender. The connection creation is completed when the sender receives the confirmation messages from the all of the active receivers, or when a pre-specified timer expires (see 8.2).

Throughout the connection creation, some or all of the enrolled group receivers will join the connection. The receivers that have joined the connection are called active receivers. An enrolled receiver that is not active can participate in the connection as a late-joiner (see 8.6). The late-joiner sends a join request to the sender. In response to the join request, the sender transmits a join confirm message, which indicates whether the join request is accepted or not. An active receiver can leave the connection by sending a leaving request to the sender. A trouble-making receiver, who cannot keep pace with the current data transmission rate, may be ejected (see 8.7).

After a connection is created, the sender begins to transmit multicast data (see 8.3). For data transmission, an application data stream is sequentially segmented and transmitted by means of data packets to the receivers. The receivers will deliver the received data packets to the applications in the order they were transmitted by the sender.

To make the protocol scalable to large multicast groups, ECTP employs the tree-based reliability control mechanisms. A hierarchical tree is configured during connection creation. A control tree defines a parent-child relationship between any pair of tree nodes. The sender is the root of the control tree. In the tree hierarchy, a set of local groups are defined. A local group consists of a parent and zero or more children. The error, flow and congestion controls are performed for each local group defined by the control tree.

Figure 3 illustrates a control tree hierarchy for reliability control, in which a parent-child relationship is configured between a sender (S) and a receiver (R), or between a parent receiver (R) and its child receiver (R).

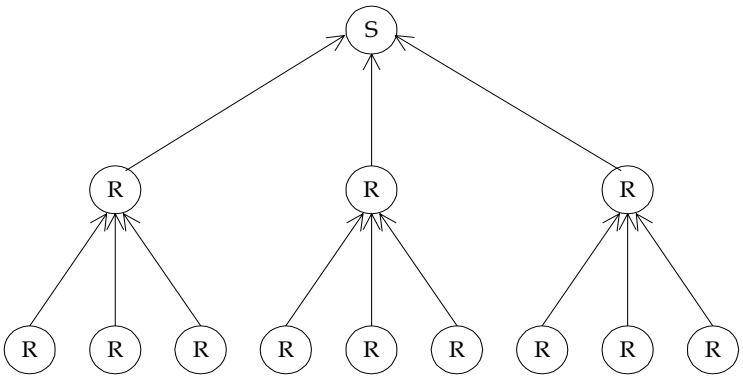


Figure 3 – Control Tree Hierarchy for Reliability Control

ECTP specifies the protocol procedures for tree creation. In the tree creation, a control tree is gradually expanded from the sender to the receivers (see 8.2.2). This is called a top-down configuration. On the other hand, the IETF RMT WG has proposed a bottom-up approach, where the receivers initiate a tree configuration (see Annex B). Those schemes may be incorporated into the ECTP as candidate tree creation options in the future.

Tree-membership is maintained during the connection. A late-joiner is allowed to join the control tree. The late-joiner listens to the heartbeat messages from one or more on-tree parents, and then joins the best parent. When a child leaves the connection, the parent removes the departing child from the children-list. Node failures are

detected by using periodic control messages such as null data, heartbeat and acknowledgement. The sender transmits periodic null data messages to indicate that it is alive, even if it has no data to transmit. Each parent periodically sends heartbeat messages to its children. On the other hand, each child transmits periodic acknowledgement messages to its parent (see 8.8).

In ECTP, error control is performed for each local group defined by a control tree (see 8.4). If a child detects a data loss, it sends a retransmission request to its parent via ACK packets.

An ACK message contains the information that identifies the data packets which have been successfully received. Each child can send an ACK message to its parent using one of two ACK generation rules: ACK number and ACK timer. If data traffic is high, an ACK is generated for the ACK number of data packets. If the traffic is low, an ACK message will be transmitted after the ACK timer expires.

After retransmission of data, the parent activates a retransmission back-off timer. During the time interval, retransmission request(s) for the same data will be ignored. Each parent can remove the data out of its buffer memory, if those have been acknowledged by all of its children.

The flow and congestion control information is delivered from the receivers to the sender, along the control tree. The detailed description of flow and congestion control will be given in ECTP part 2, the QoS management specification for the simplex multicast transport. Based on the monitored flow and congestion control information, the sender will adjust the transmission rate.

During the data transmission, if network problems (for example, severe congestion) are indicated by the QoS management functions specified in ECTP part 2, the sender suspends the multicast data transmission temporarily. In this period, no new data is delivered, while the sender transmits periodic null data messages to indicate that the sender is alive. After a pre-specified time has elapsed, the sender resumes the multicast data transmission (see 8.5).

The sender terminates the connection by sending a termination message to all the receivers, after all the multicast data are transmitted. The connection may also terminate due to a fatal protocol error such as a connection failure (see 8.9)

7. Protocol components

7.1 Nodes

ECTP protocol mechanisms are based on a logical control tree, which defines a parent-child relationship between any pair of tree nodes. Each node on the tree is classified into one of three node types: TO (top owner), LO (local owner), and LE (leaf entity).

a) Top Owner (TO)

TO is the root of the control tree and also a single sender in the simplex multicast connection. TO manages the overall connection management functions including the connection creation and termination. In the connection creation phase, a control tree is configured by interactions between the sender and receivers. After the connection is created, TO sends multicast data to the receivers. TO can temporarily suspend and resume the connection. TO can admit or reject the group members who want to join the existing connection. After all the data is transmitted, TO terminates the multicast transport connection.

b) Local Owner (LO)

In the ECTP connection, some of the receivers may be designated as LOs. Each LO has its children that consist of other LOs and/or LEs. LOs are thus located as interior nodes on the tree. Each LO retransmits the multicast data that have been lost by its children. It also aggregates the information on the flow and congestion control from its children, and forwards the aggregated information toward TO. TO is also an LO in terms of the reliability control operations.

c) Leaf Entity (LE)

A receiver, which has not been designated as LO, is referred to as an LE. An LE cannot have any children. It is thus located as a leaf node on the control tree.

TO is a single sender. LOs and LEs are receivers. In the tree hierarchy, a local group consists of a parent and its children. TO or an LO can be a parent, and an LO or LE can be a child.

In the tree hierarchy, an LO retransmits lost multicast data to its children (error recovery) and forwards the flow and congestion control information to TO. Moreover, each LO has authority to eject a trouble-making child to maintain the stability of the connection. It is thus expected that LOs are given more processing power and responsibility than LEs.

In ECTP, it is presumed that some of receivers have been designated as LOs before the connection is created. This specification does not consider the selection of LOs among flat receivers during the connection. That is, before the connection creation (or in the enrollment phase), each receiver MUST know whether it is an LO or LE. In a very small-sized group or asynchronous networks such as satellite or mobile networks, no LO may be designated. In those environments, all the receivers will be LEs.

An LO may be an end host or a dedicated server. In privately controlled networks, it is probable that dedicated servers function as LOs. In public networks, end hosts may be employed as LOs. In either case, an LO is a receiver and performs the reliability control operations for its local group as a parent.

7.2 Control tree

After a connection is created, TO transmits data to all the receivers by multicast. Each child sends status information on data reception to its parent. The information will thus be delivered to TO along the control tree. The multicast data streams flow from TO to LOs and LEs in the downward direction, while the control information is transferred from LEs to TO via LOs in the upward direction along the control tree.

Figure 4 shows a general structure of an ECTP control tree.

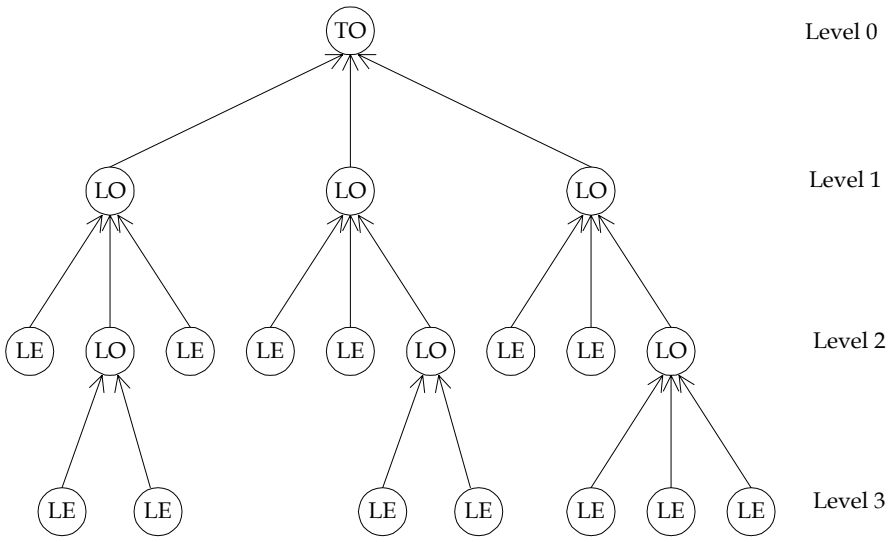


Figure 4 – An ECTP Control Tree

A control tree defines the parent-child relationship between any pair of nodes. The control tree provides each tree node with the following information:

- Who is my parent node ? (at LOs and LEs);

- Who are my children nodes ? (at TO and LOs).

Based on the information described above, a tree node may keep its parent and/or children list. In the list, each element is identified by its transport address, and all the elements are arranged in the order of a pre-specified rule such as IP address or hop distance, etc. The parent list may have one or more elements, some of which will be used as backup parent nodes against a failure of the current parent node.

ECTP provides three options for tree configuration (see 8.2.2). The other options may additionally be defined in the future according to the requirements of multicast applications (see Annex B).

7.3 Addressing

7.3.1 Port

ECTP uses a set of ports to identify different applications in an IP end host. The port number is inserted in each packet header. In general, an IP host can support a number of ports, and each port number is unique within the host. The binding of ports to processes is handled independently by each host.

For a multicast transport connection, at least two ports are used. If implementations use the socket interface, it will be bound to at least two ports. One of them is a port used for multicast data transmission and reception, which MUST be announced to the group members before the connection is created. Another is a port assigned locally within a system, which will be referred to as a destination port number for transmission of unicast control messages.

When a transport connection is released, it is necessary to prevent the current port from being reused by another new connection, because the packets associated with the current port may still exist in the network and flow into the port even after the connection is terminated. It is thus recommended that the relevant port be set to be in a frozen state, if the connection is released. The port being in the frozen state MUST NOT be reused by any other connection for a specific time.

7.3.2 Transport addresses

A transport address is defined as a pair of an IP address and a port number. The multicast transport address consists of an IP multicast address and a port number. TO sends multicast data by setting the multicast transport address as the destination transport address. Each receiver receives the multicast data from TO over the multicast transport address. Such a multicast transport address for data transmission MUST be announced to all the group members in the enrollment phase.

A unicast transport address is identified with a pair of an IP unicast address and a local port number. When TO sends multicast data, it sets the corresponding source transport address to its unicast transport address. The unicast transport address is also used when a node transmits a unicast control message to another node.

7.3.3 Multicast data and control addresses

In ECTP, TO sends data to all receivers by multicast, while LOs send retransmission data and control messages to its local group by multicast.

Depending on the multicast deployment in the network, TO and LOs may share a single IP multicast address or use different IP multicast addresses. For example, the source specific multicast (SSM) routing protocol defines a multicast channel by a pair of an IP multicast address and a unicast address of the source. In SSM, a multicast channel is unique to the sender (see Annex A for more in detail).

In the networks where TO and LOs use different multicast addresses or channels, all of the multicast addresses employed MUST be announced to the group members, before the connection is created. In this case, one of them is used for the multicast data transmission by TO, and the others are for the multicast control by LOs such as the multicast data retransmission and the multicast transmission of control messages.

7.4 Packets

ECTP packets are classified into data and control packets. Data (DT) and Retransmission Data (RD) are the data packets. All the other packets are used for control purposes. Table 1 summarizes the packets used in ECTP. In the table, the transport type 'multicast' represents global multicast using a multicast data address, while the 'local multicast' does local multicast using a multicast control address. The RD and HB control packets are delivered from an LO to its local group (i.e., its children) by local multicast.

Table 1 – ECTP Packets

Packet	Acronym	Transport Type	From	To
Creation Request	CR	Multicast	Sender	Receivers
Creation Confirm	CC	Unicast	Child	Parent
Tree Join Request	TJ	Unicast	Child	Parent
Tree Join Confirm	TC	Unicast	Parent	Child
Data	DT	Multicast	Sender	Receivers
Null Data	ND	Multicast	Sender	Receivers
Retransmission Data	RD	(Local) Multicast	Parent	Children
Acknowledgement	ACK	Unicast	Child	Parent
Heartbeat	HB	(Local) Multicast	Parent	Children
Late Join Request	JR	Unicast	Receiver	Sender
Late Join Confirm	JC	Unicast	Sender	Receiver
Leave Request	LR	Unicast	Parent/Child	Child/Parent
Connection Termination	CT	Multicast	Sender	Receivers

NOTE 1 – Sender is TO, and Receivers are LOs and LEs.

NOTE 2 – Parent is TO or an LO, and Child is an LO or LE.

NOTE 3 – See Table 3 in 9.3 for the detailed packet structure.

8. Protocol procedures

8.1 Operations before the connection creation

Before an ECTP connection is created, every prospective group member has been enrolled to the multicast group. Such a member is called an enrolled group member (see ITU-T Recommendation X.601). Some or all of the enrolled group users will participate in the ECTP connection.

Before an enrolled group member joins the multicast connection, it MUST be attached to the network interface with the help of the IGMP and IP multicast routing protocols. This ensures that the enrolled member listens to the multicast data and control packets from TO and LOs.

An enrolled user gets information on the multicast session, including IP addresses and port numbers, via SDP/SAP, HTTP (Web page) or E-mail. The detailed enrollment mechanisms are outside the scope of the ECTP specification.

To ensure that an enrolled user participates in the ECTP connection, the following transport addresses MUST be announced to the enrolled group, together with the session-specific information:

1) Multicast group (data) transport address: an IP multicast address and a port number

   The IP multicast address and port number combination MUST be unique to an ECTP connection, and it will be used by TO to transmit the multicast data to the receivers;

2) Unicast transport address of TO: an IP unicast address and a port number

   The IP address and the port number correspond to the source IP address and the port number for the multicast data packets, respectively. They will also be referred to as the destination IP address and the port number for the control packets flowing from receivers to TO.

3) Unicast transport address of LO: an IP unicast address and a port number

   This unicast transport address corresponds to the source IP address and port number for the multicast control packets such as TJ and ACK packets. It is also referred to as the destination address and port number for the control packets from the children to LO.

4) Multicast control transport address of an LO: an IP multicast address and a port number

   If TO and LOs use different multicast addresses, each LO also announces its multicast address and port numbers. The IP address and the port number will be used by LO to transmit the multicast control packets such as HB and RD packets to the children;

Using multicast addresses, each enrolled member MUST have been attached to the network, via the IGMP and IP multicast routing protocols, before the connection is created. The member listens to the multicast control and data control messages from TO and LOs.

8.2 Connection creation

8.2.1 Procedures for connection creation

TO triggers the connection creation by sending a CR packet to the enrolled receivers. Some or all of the enrolled receivers will respond with their respective CC packets. TO completes the connection creation by aggregating those CC packets. The receivers that have participated in the connection are called 'active receivers.'

If one or more LOs are employed for the tree creation, each LE sends its CC packet to its parent LO. The parent LO aggregates the CC packets from its children, and then sends an aggregated CC packet to its parent.

The connection creation procedures are summarized as follows:

1) TO transmits a CR packet to all the receivers by multicast.

   TO then activates the *Connection Creation Time* (*CCT*) timer;

2) When a receiver receives the CR packet, it begins to configure the control tree (see 8.2.2);

3) After each receiver joins the tree, it sends a CC packet to its parent by unicast, and then waits for multicast data from TO. Each on-tree parent LO aggregates the CC packets from all of the children and then sends an aggregated CC to its parent;

4) TO aggregates CC packets from its children while the *CCT* timer is valid. If the *CCT* timer expires, TO completes the connection creation for the receivers that have sent CC packets until then.

Figure 5 depicts the generic procedures for connection creation.

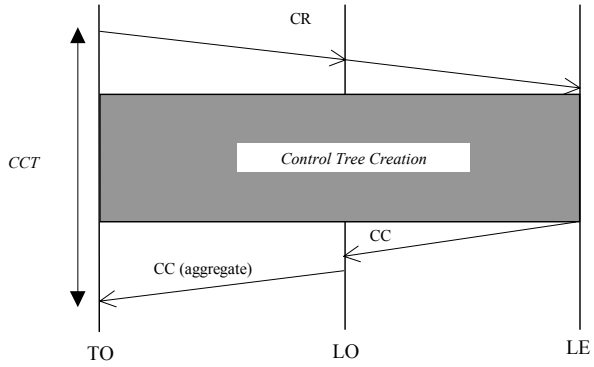


Figure 5 – Connection Creation Procedures

In the figure, the detailed tree creation procedures are described in 8.2.2.

After the connection creation is completed, TO transmits multicast data. The receivers that have not participated in the connection may join the connection as late-joiners (see 8.6).

All the group members, a sender and receivers, activate *Inactivity Time* (*IAT*) timers, when connection creation is indicated. The *IAT* timer is used to protect against abnormal protocol operations. The *IAT* timer is reset each time a new packet arrives. If *IAT* timer expires without reception of any packet, the corresponding node determines that the connection has failed.

8.2.2 Control tree creation

In the connection creation, ECTP configures a hierarchical control tree connecting TO and LEs via zero or more LOs. ECTP provides three options for the tree creation:

a) Option 1: Level 1 Configuration, in which no LO is employed;

b) Option 2: Level 2 Configuration, in which all LOs are connected to TO;

c) Option 3: General Configuration, in which more than two tree levels may be configured;

One of these three options MUST be specified in the connection information element (see 9.2.1). Depending on the network infrastructure, the other options may be defined for the tree creation in the future, which includes the schemes proposed by the IETF RMT WG. The brief sketches for those options are given in Annex B.

The tree creation algorithm automatically constructs a control tree. To ensure the 'no loop configuration' in the tree, the top-down approach is employed with procedural steps. Starting from TO, the tree is gradually expanded by including non-tree LOs and LEs in the stepwise manner.

For the stable operation and maintenance of ECTP protocol mechanisms, Options 1 and 2 are recommended for the tree creation. Figure 6 and 7 illustrate the control tree in Option 1 and 2, respectively.

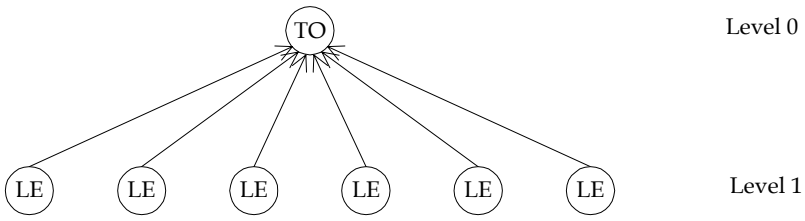


Figure 6 – One-level Tree in Option 1

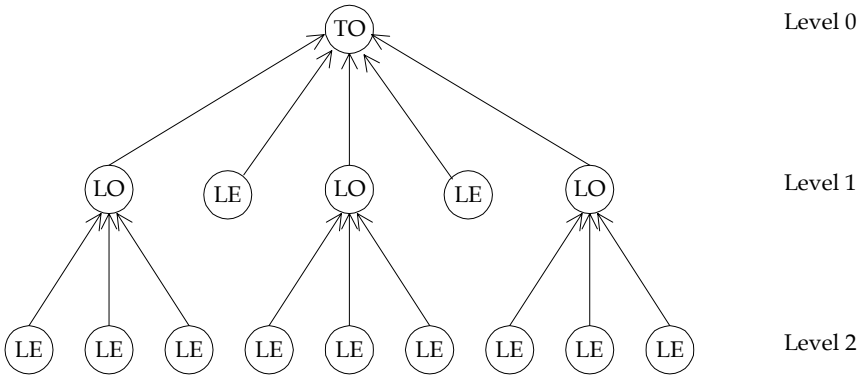


Figure 7 – Two-level Tree in Option 2

Each receiver node starts the tree creation as soon as a CR packet is received. This specification first provides the tree creation procedures for Option 2, and the procedures for Option 1 and 3 will be described later.

In Option 2, all the LOs are connected to TO, and each LE can join an on-tree LO or TO. Figure 8 illustrates the tree creation procedures for Option 2.

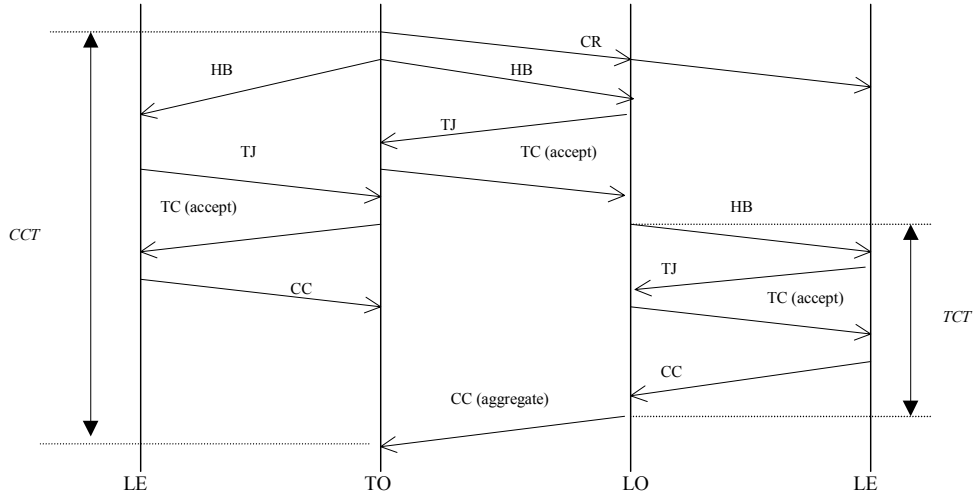


Figure 8 – Tree Creation Procedures

The tree creation procedures are summarized as follows:

1) TO begins to send periodic HB packets over its multicast control address, just after the CR packet is released.

2) Each LO joins the TO by sending a TJ packet, and then activates *Retransmission Time* (*RXT*) timer.

3) TO responds with a TC packet to each LO. The TC packet contains a flag bit to indicate whether the join request is accepted or not.

4) If an LO receives the TC packet with the acceptance flag within the *RXT* time, it becomes an on-tree now.

   In the rejection case, the LO cannot be on the tree. If the TC packet does not arrive within *RXT* time, the LO retransmits a TJ packet to TO.

5) Each on-tree LO begins to advertise periodic HB packets over its multicast control address. Then the on-tree LO activates the *Tree Creation Time* (*TCT*), which is a half of *CCT*.

Each on-tree LO uses a multicast control address to invite its children (see 7.3.3). Each receiver that wants to be connected to the control tree SHOULD join one or more multicast control addresses. This ensures that each LO or LE listens to the HB packets from the candidate parents.

Each LE listens to HB packets from the on-tree TO and LOs. Those candidate parents are recorded into its parent list. When an LE contains one or more entities in its parent list, it selects the best candidate parent. The specific selection rule is an implementation issue, which may be based on the shortest hop distance, the most recently received HB packet, or the lowest IP address, etc. For the parent recovery against the failure, it is desirable that each LE keeps two or more candidate parents in its parent list.

Then an LE joins TO or an on-tree LO as follows:

6) The LE sends a TJ packet to the best parent by unicast, and then activates *RXT* timer.

7) Each on-tree LO responds with a TC packet, which contains the flag bit. The decision of acceptance or rejection is made based on the *Maximum Children Number* (*MCN*).

8) If an LE receives the TC packet with the acceptance flag within *RXT*, it is on the tree now. The on-tree LE sends a CC packet to its parent by unicast. In the rejection case, LE tries to join an alternate on-tree LO. If TC packet does not arrive within *RXT*, LE retransmits a TJ packet to the on-tree LO.

9) If the *TCT* timer expires, then LO aggregates CC packets and sends an aggregate CC to its parent (i.e., TO). If the *CCT* timer expires, TO completes the connection creation and tree creation process.

In Option 1, no LO is employed in the tree. In fact, no tree is configured. All the receivers become the children of TO. This option may be used in multicast connections that do not display the scalability problems.

In Option 3, a general tree is configured with more than two tree levels. The parent-child relationship can be formed between two LOs. Thus an LO may be a child or a parent of another LO. In this option, the tree creation mechanism is the same as that in the option 2, except the following points:

a) When an LO is on the tree, the on-tree LO increases the *Current Tree Level* (*CTL*) value by one. Note that a child LO of TO has *CTL* of '1'. In step (5), for the given *Maximum Tree Level* (*MTL*), an LO sets its *TCT* to "*CCT* * (*MTL* – *CTL*) / *MTL.*"

b) The steps (6), (7), (8) and (9), which are called the 'branching process', will be performed between a parent LO and its children. A new on-tree LO will begin the branching process again to find its children until its *TCT* timer expires; and

c) To ensure that the tree grows from the root to the multiple tree levels, each parent may reserve a portion of the *Maximum Children Number* (*MCN*) value for the children LOs.

The CC packet contains information on the number of active receivers, *Active Receivers Number* (*ARN*). Each LE sets its *ARN* to '1', while a parent aggregates the *ARN* values for its local group by summing up the number of its descendants. In this way, TO can know how many receivers are active in the connection.

8.3 Data transmission

After the connection is created, TO transmits multicast data to all the receivers. TO will generate DT packets by the segmentation procedure. To do this, TO splits a multicast data stream into multiple DT packets. Each DT packet has its own sequence number (see 8.3.2). TO sets the *F* bit of the fixed header to '1' (see 9.1) for the last DT packet of the data stream.

When TO has no data to transmit, TO transmits the periodic ND packets. *Null Data Time* (*NDT*) is a time interval between multicast transmissions of ND packets by TO. The *NDT* timer is activated after the connection is created. Each time a DT or RD packet is transmitted, the *NDT* timer is reset. If *NDT* timer expires, TO transmits an ND packet.

All the data packets received are delivered to the application in the order sent by TO. Each receiver reassembles the received packets. Corrupted and lost packets are detected by using a checksum (see 8.3.1) and sequence number (see 8.3.2). A corrupted packet is considered as a loss. The lost DT packets are recovered in the error control function (see 8.4.2).

ECTP uses flow control based on a fixed-sized window, which is the same as the *ACK Bitmap Size* (*ABS*). The *window size* represents the number of unacknowledged data packets in the sending buffer. Sender can maximally transmit the *window size* data packets at the configured data transmission rate. In ECTP, the transmission rate of multicast data is controlled by the rate-based flow and congestion control mechanisms, which will be specified in ECTP part 2, QoS management specification.

8.3.1 Checksum

This function is used to detect corruption of a received packet. This checksum covers a whole packet including the header, extension elements and/or data parts (see 9.1). The checksum MUST be applied to all packet types. It is calculated and stored by the sending entity, and then verified by the receiving entity.

To compute the checksum for an outgoing packet, the sender first sets the checksum value to '0'. Then the 16-bit one's complement sum of the packet is calculated. The 16-bit one's-complement of this sum is stored in the checksum field of the fixed header.

If the calculated checksum is '0', it is stored as all one bits, i.e., 65535, which is equivalent in one's-complement arithmetic. Note that the transmitted checksum of '0' indicates that the sender did not compute the checksum.

On reception of a packet, each receiver calculates the 16-bit one's-complement sum of the packet. The calculated checksum MUST be all one bits, since the checksum value reflects the checksum stored by the sender. If not, it means a checksum error. In this case, the receiver discards the packet. A corrupted packet is considered as a loss. A lost data packet will trigger the retransmission request (see 8.4.2).

8.3.2 Sequence number

A new DT packet is sequentially numbered by TO. The sequence number is used to detect lost data packets by receivers and to manage the transmission and retransmission buffers by TO and LOs.

On transmitting the CR packet, TO chooses an initial sequence number (*ISN*). The *ISN* is randomly generated other than '0'. The sequence number '0' MAY be used to indicate an inactive connection.

The packet sequence number is increased for each new DT packet. Modulo 2^{32} arithmetic is used and the sequence number wraps back around to '1' after reaching "$2^{32} - 1$".

8.4 Error recovery

The reliability control mechanisms typically consist of error recovery and flow and congestion control operations. Flow and congestion control mechanisms are designed to adjust the data transmission rate based on the monitored status of the receivers and the networks, and these objectives are a good match with the design goals of the QoS management specification. The flow and congestion control operations will thus be specified in the QoS management specification in ECTP part 2, along with QoS monitoring and maintenance operations.

This specification focuses on error recovery, which consists of error detection by receivers, retransmission request by receivers via ACK packet, and retransmissions by parents. In this specification, adjustment of data transmission rates will not be considered (i.e., the sender is assumed to transmit the multicast data at a fixed rate).

8.4.1 Error detection

The header checksum field is used for detection of packet corruption, and the sequence number field is for detection of a packet loss. When a data packet is received, each receiver examines the header checksum. If the checksum field is invalid, the packet is regarded as a corruption and shall be discarded. A corruption is treated as a loss. The loss can be detected as a gap of two consecutive sequence numbers for DT packets. The loss information is recorded into the ACK bitmap, which is attached to the subsequent ACK packets.

8.4.2 Retransmission request

ACK packets are used for the retransmission requests. When a receiver detects a gap in the sequence numbers of received packets, it sets to zero the bit of the ACK bitmap which corresponds to the lost DT packet. The ACK bitmap is included into the acknowledgement element, which is attached to the subsequent ACK packet and delivered to the parent by the ACK generation mechanisms.

For a local group, a parent and its children maintain the following variables to determine the status of DT packets:

a) *Lowest Sequence Number* (*LSN*): If a node is a child, this is the sequence number of the lowest numbered DT packet that the child has not acknowledged. If the node is a parent, this is the sequence number of the lowest numbered DT packet that has not been acknowledged by any of its children;

b) *Highest Sequence Number* (*HSN*): If the node is a child, this is the sequence number of the highest numbered DT packet that has been received. If the node is a parent, then this is the sequence number of the highest numbered DT packet that has been received by any of its children;

To request the retransmissions of lost data, each child makes an acknowledgement element containing the *LSN*, *Valid Bitmap Length* and *ACK Bitmap*. The *Valid Bitmap Length* is set to *HSN* – *LSN* + 1. For an example, for *LSN* = 15 and *HSN* = 22, the *Valid Bitmap Length* = 8. The *ACK Bitmap* specifies a success or a failure of a packet delivery: '1' for success and '0' for failure. A bitmap can represent *Bitmap Length* * 32 packets maximally. Suppose *Bitmap* = 01101111. Then the DT packets with the sequence number 15 and 18 are lost.

Note that an intermediate LO on the tree has two sets of the *LSN* and *HSN* parameters: the one set as a child and the other set as a parent. The parameter values for a child are updated by the status of the data reception from TO, while the parameter values for a parent will be refreshed by the acknowledgement element from the children.

When a parent sends a HB packet to its children, it sets the sequence number field to the *LSN*. The data packets, whose sequence number is smaller than the *LSN*, cannot be recovered.

8.4.3 ACK generation

Each child generates an ACK packet by *ACK Generation Number* (*AGN*) or by *ACK Generation Time* (*AGT*).

Each child sends an ACK packet to its parent every *AGN* number of packets. To do this, a child receives a *Child ID* from its parent in tree configuration, which is contained in the tree membership element. Each child sends an ACK packet to its parent, if the sequence number of a DT packet modulo *AGN* equals *Child ID* modulo *AGN,* i.e., if

Packet Sequence Number % *AGN* = *Child ID* % *AGN*.

Suppose *AGN* = 8 and *Child ID* = 2. The child generates an ACK packet for the DT packets whose sequence numbers are 2, 10, 18, 26, etc. This ACK generation rule is applied when the corresponding DT packet is received or detected as a loss by the child.

When data traffic is low, a receiver may not send an ACK packet for a long time. This could cause a long wait for packet stability at the parent and could also make the receiver appear to have failed. *AGT* is used to ensure that the receivers respond in a timely manner. A receiver sends at least one ACK packet within the *AGT* interval. *AGT* timer is initialized when a child receives the first DT packet, and it is reset each time a new ACK packet is sent.

In summary, when the data traffic is high, ACK packets will be generated by the *AGN* number rule. On the other hand, ACK packets are triggered when *AGT* timer expires, if the traffic load is low.

8.4.4 ACK aggregation

Each parent uses ACK packets to gather status information for the error, flow and congestion controls.

Each time a parent receives an ACK packet from any of its children, it records and updates the status information on which packet(s) have been successfully received by its children. A DT packet is defined as a stable packet if all of the children have received it. The stable DT packets are released out of the buffer memory of the parent. When a parent receives an ACK packet from one of its children, if one or more packet losses are indicated, the parent transmits the corresponding RD packets to all of its children over its multicast control address. (See 8.4.6)

An ACK packet contains information on the flow and congestion control. The parent must aggregate the corresponding control variables for all of its children, and sends the aggregated information to its parent by using its subsequent ACK packet. For any control node (TO or LO) the aggregated information represents the receiving status for all of its descendants including its own children. A more detailed specification of the flow and congestion controls will be given in the QoS management specification in ECTP part 2.

8.4.5 Local RTT measurement

The Round Trip Time (RTT) for a local group, Local RTT, is measured by comparing a HB packet with its corresponding ACK packets. A parent LO sends a HB packet containing a timestamp element to its children every *HB Generation Time* (*HGT*) interval. Each child updates the *Timestamp* that was in the HB packet, before it sends an ACK packet to its parent, as follows:

1) The child records the time when the HB packet arrives as $T_{receive}$;

2) The child records the time when it sends the corresponding ACK packet to its parent as T_{send};

3) The ACK packet delivers the following *Timestamp* value:

$$Timestamp = Timestamp + T_{send} - T_{receive}.$$

Receiving an ACK packet from a child, the parent calculates the RTT from the child by subtracting *Timestamp* from the current time. The RTT is recorded into the children list. The parent determines Local RTT by the minimum RTT value for its children.

Calculation of Local RTT is optional. Local RTT may be used to determine the *Retransmission Back-off Time* (*RBT*) for the local group. After a parent retransmits a data packet, it will ignore any subsequent retransmission requests for the same packet during the *RBT* period (see 8.4.6). The *RBT* may be set to two or three times Local RTT. It is recommended to use *RBT* value large enough to reduce unnecessary duplicated retransmissions. Additional use of Local RTT in ECTP may be defined in the future.

8.4.6 Retransmission

In response to an ACK packet, each parent retransmits RD packets for the data that are requested by any children, if it holds the requested data packets. RD packets are retransmitted over the multicast control address.

After a parent sends an RD packet for the requested data, it activates the *Retransmission Back-off Timer* (*RBT*). During that time, retransmission requests for the same data packet will be ignored.

The maximum number of retransmissions for a lost DT packet shall be bounded to *Maximum Retransmission Number* (*MRN*). The parent ignores further retransmission requests exceeding *MRN*, and removes the corresponding data out of its buffer memory.

8.5 Connection pause and resume

This function is used to suspend multicast data transmissions temporarily. If the connection pause is indicated, TO transmits periodic ND packets with the *F* bit of the fixed header set to '1' (see 9.1). In the connection pause period, TO must not transmit any new DT packet, while control packets including RD and HB can be sent. The connection resume is activated if the sender realizes that connection status is recovered from an abnormal state. If the connection resume is indicated, the sender transmits periodic ND packets with the *F* bit of the fixed header set to '0'. The specific rules to trigger the connection pause and resume will be given in ECTP part 2.

8.6 Late join

To join an existing connection, the late-joiner performs the following procedures:

1) The late-joiner sends a JR packet to TO by unicast. It then activates *Retransmission Time* (*RXT*) timer;

2) On reception of a JR packet, TO responds with a JC packet to the late-joiner. JC packet MUST specify in the flag *F* bit of the fixed header whether the join request is accepted or not: 0 (accept) or 1 (reject);

3) If the late-joiner receives a JC packet with an acceptance flag within *RXT*, then it begins to join the control tree in the tree configuration. If *RXT* timer expires without reception of a TC packet, the late-joiner sends the JR packet to TO again. The number of retransmissions of the JR packet is bounded by *Maximum Retransmissions Number* (*MRN*).

In the tree configuration, the late-joiner joins the tree by sending a TJ packet to an on-tree LO and by receiving a TC packet from the on-tree parent (see 8.8.1 for more in detail).

Figure 9 describes the procedures for the late join operation.

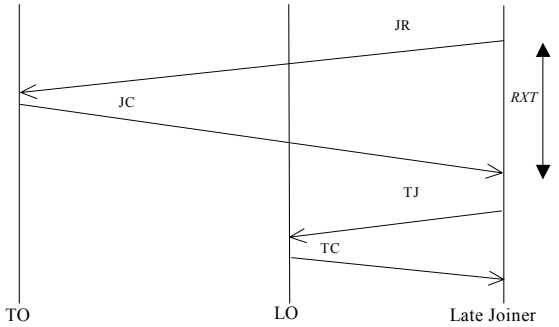


Figure 9 – Protocol Procedures for Late Join

8.7 Leave

This function is used when a receiver leaves an existing connection, or a parent ejects a trouble-making child.

8.7.1 User-invoked leave

According to the leave request of the application, the leaving receiver sends an LR packet to its parent by unicast. The parent then updates its child list.

In case that the leaving receiver is an LO, the protocol behavior may become unstable, because each of its children has to find a new alternate parent. In this case, the reliability may not be guaranteed. For the stable operation of the protocol, it is not recommended that LOs leave the connection.

8.7.2 Troublemaker ejection

TO or LO can eject a trouble-making child. When a troublemaker is detected, the parent sends an LR packet to the troublemaker. It then removes the troublemaker from its child list. An example of troublemaker is a failed child in the tree hierarchy (see 8.8.3.) When a child receives an LR packet from its parent, it MUST leave the connection. In particular, the ejection of an LO is not desirable, because the LO may have one or more children. For stable operation of the protocol, it is not recommended to eject an LO.

The specific rules to define a troublemaker will be discussed in ECTP part 2.

8.8 Tree membership maintenance

After an initial control tree is created in the connection creation, the tree membership is maintained until the connection is terminated. The tree membership maintenance deals with the following issues:

- Tree configuration for late-joiners;
- Tree reconfiguration for leaving receivers; and
- Tree reconfiguration against node failures.

8.8.1 Tree configuration for late joiners

After the connection is created, each on-tree LO advertises periodic HB packets over its multicast control address. When the late-joiner receives a JC packet from TO (see 8.6), it begins to locate a suitable parent.

The late-joiner listens to HB packets from one or more LOs, and information on candidate parents are recorded into the parent list. Again, if the multicast control addresses are different to the multicast data address, each late-joiner must have joined one or more multicast control addresses of TO or LOs, together with the multicast data address, in the enrollment phase.

The late-joiner selects the best candidate parent from its parent list. The selection rule is an implementation issue. The late-joiner sends a TJ packet to the selected candidate parent, and activates *RXT* timer.

The on-tree parent responds with a TC packet to the late-joiner, which contains the flag bit to indicate an acceptance or rejection. The decision is made based on the *MCN*.

If the late-joiner receives the TC packet with the acceptance flag within the *RXT* time, it is on the tree now. In the rejection case, the late-joiner tries to join an alternate candidate parent in the parent list. If the TC packet does not arrive within *RXT* time, the late-joiner retransmits a TJ packet.

8.8.2 Tree reconfiguration for leaving receivers

As described in 8.7, when a child leaves the connection, the parent removes the child out of its child list.

8.8.3 Tree reconfiguration against node failures

To detect a node failure, the *Node Failure Threshold* (*NFT*) is employed. The tree maintenance procedures are different for the node types: TO, LO and LE.

TO is a single sender in the simplex multicast connection. Each receiver detects the failure of TO by the *NDT* interval. If a receiver cannot hear any packet from TO during the interval *NFT* times *NDT*, it determines that TO has failed. Then the receiver leaves the connection.

Each parent LO advertises periodic HB packets after it becomes an on-tree node. A child detects the failure of its parent, if it cannot receive any packets such as HB and RD packets from the parent during the interval *NFT* times *Heartbeat Generation Time* (*HGT*). Then the child begins to find an alternate parent.

A parent detects the failure of a child, if it cannot hear any ACK packets from the child during the interval *NFT* times *ACK Generation Time* (*AGT*) or for the number *NFT* times *AGN* data packets. If a child is detected as a failure, the parent sends an LR packet to the failed child, and clears the child out of its children list.

8.9 Connection termination

TO terminates a multicast transport connection by sending a CT packet to all the receivers by multicast. When the connection termination is indicated, TO shall discard all subsequently received packets and also freezes the local port number. On the receipt of a CT packet, each receiver freezes the local port number.

This function will be initiated after all the multicast data are transmitted. TO also terminates the connection on detection of a fatal protocol error. For an example, if no packet is received during the *IAT* interval, then TO terminates the connection.

9. Packet formats

An ECTP packet MUST contain a fixed header and extension elements or data parts. The fixed header consists of 16 bytes. The extension elements are arranged in the specified order (see 9.2).

The packet format is illustrated below:

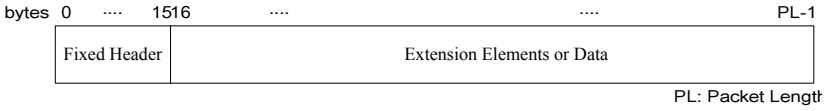


Figure 10 – Packet Format

9.1 Fixed header

The 16-byte fixed header contains parameter fields used in all protocol operations. If any of the fields have an invalid value, this is a protocol error.

The following figure shows the structure of the fixed header, when ECTP operates over IP:

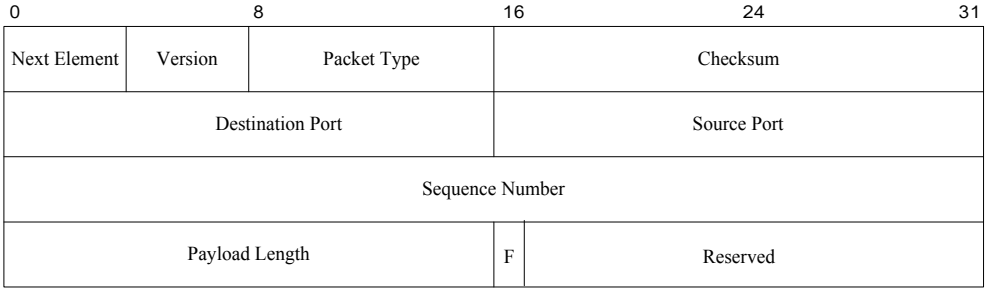


Figure 11 – Fixed Header Format

A fixed header contains the following information:

a) *Next Element* – indicates the type of the next component immediately following the fixed header. The next element field of the last extension element MUST be set to '0000', meaning "no further element" (see 9.2);

b) *Version* – defines the current version of the ECTP protocol. It starts at '1';

c) *Packet Type* – indicates the type of the current packet (see 9.3);

d) *Checksum* – is used to check the segment validity of a packet (see 8.3.1);

e) *Destination* and *Source Ports* – are used to identify the sending and receiving applications. These two values are used as the transport addresses, together with the source and destination IP addresses in the IP header. The port number is 16-bit long.

f) *Sequence Number* – is the sequence number of a data packet in a series of segments. This sequence number is a 32-bit unsigned number that wraps back around to '1' after reaching '$2^{32} – 1$' (see 8.3.2);

g) *Payload Length* – indicates the length of the elements or data part in bytes, following the fixed header. For control packets, it represents the length of extension elements. For data packets, it indicates the length of data parts;

h) *F* – is a flag bit. Depending on the packet types, it has a different purpose:

   1) For the DT packet, *F* = 1 indicates the 'end of stream';

   2) For the JC (join confirm) and TC (tree join confirm) packets, the *F* = 1 indicates that the corresponding join request is accepted. F is set to 0 for the rejection case;

   3) For the ND packet, *F* = 1 indicates the connection pause period. For the other cases, it is set to '0';

   4) For the LR packet, *F* is set to '1' for the user-invoked leave (see 8.7.1), or set to '0' for the troublemaker ejection (see 8.7.2);

   5) For the CT packet, *F* is set to '1' for an abnormal termination, or set to '0' for the normal termination after all the data have been transmitted (see 8.9); and

   6) For the other packets, this field is ignored.

i) *Reserved* – is reserved for future use.

When ECTP operates over UDP, the packet header does not need to specify the source and destination ports, which will be referred to from the UDP header. In this case, the 32-bit field for the source and destination ports will be filled with 'connection ID', which is used for identifying an ECTP connection over UDP in a host. Whether ECTP operates over IP or UDP, the fixed header provides the information commonly used in the ECTP protocol operations.

9.2 Extension elements

The header part contains the fixed header and one or more extension elements. All the header components have the next element field pointing to next components. Since an extension element also has the next element field, the header part can chain multiple extension elements.

According to the extension element type, its next element field is encoded as shown in Table 2. The next element field of the last extension element MUST be '0000'.

Table 2 – Encoding Table of the Extension Elements

Element	Encoding
Connection Information	0001
Acknowledgment	0010
Tree Membership	0011
Timestamp	0100
No element	0000

Each element has its own version value which starts at '1'. If there is a need to define additional or different use of an element in the future, the corresponding version number of the element will be increased by '1'. In the other hand, the version of the fixed header represents the current version of the ECTP protocol. The version of ECTP described in this specification is '1'.

9.2.1 Connection information

This extension element contains information on the multicast transport connection. The element structure is shown below, which has the byte length of '8':

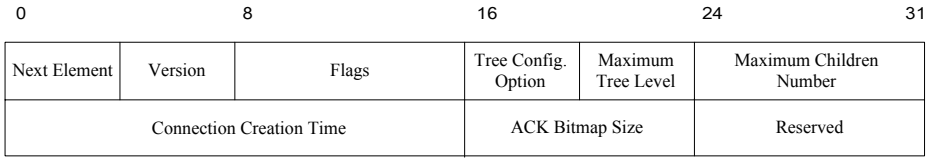


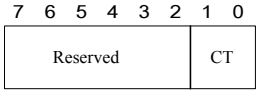
Figure 12 – Connection Information Element

The following parameters are specified:

a) *Next Element* – indicates the type of the next element immediately following this element;

b) *Version* – defines the version of this element usage. It is set to '1' at present;

c) *Flags* – consists of the following fields:



1) *Connection Type* (*CT*) – specifies which type of connection is being established as follows:

   01 – simplex multicast connection;

   The others are reserved for further extension;

2) *Reserved* – is not defined yet, and reserved for future usage.

d) *Tree Configuration Option* (in 4 bits) – specifies the tree configuration option used in the connection. The current version of this specification provides the following options (see 8.2.2):

1) 0001 – Level 1 configuration;

2) 0010 – Level 2 configuration;

3) 0011 – General configuration with more than two levels;

e) *Maximum Tree Level* (*MTL*) – specifies the maximum number of tree levels for the control tree. Values ranging from '1' to '15' are employed. The value '0' indicates that the maximum tree level for the control tree is not restricted;

f) *Maximum Children Number* (*MCN*) – specifies the maximum number of children nodes which a parent can keep on the control tree (see 10.2);

g) *Connection Creation Time* (*CCT*) – specifies a timer to limit the connection creation in units of 10 milliseconds. If this timer expires, TO completes the connection creation even if some of its children have not responded with CC packets. This timer is also used as a basis for an LO to calculate its *Tree Creation Time* (see 8.2.2);

h) *ACK Bitmap Size* – specifies the size of the bitmap in the acknowledgement element, in units of word. This value is not subject to negotiation, and thus all the receiver nodes MUST configure the bitmap field in the acknowledgement element, based on the advertised *ACK Bitmap Size*. The default value is '1', which means that each receiver can contain the information on the receiving status for 32 packets;

i) *Reserved* – is not defined yet, and reserved for future usage.

9.2.2 Tree membership

The 20-byte tree membership element contains the information on the local group, as illustrated below.

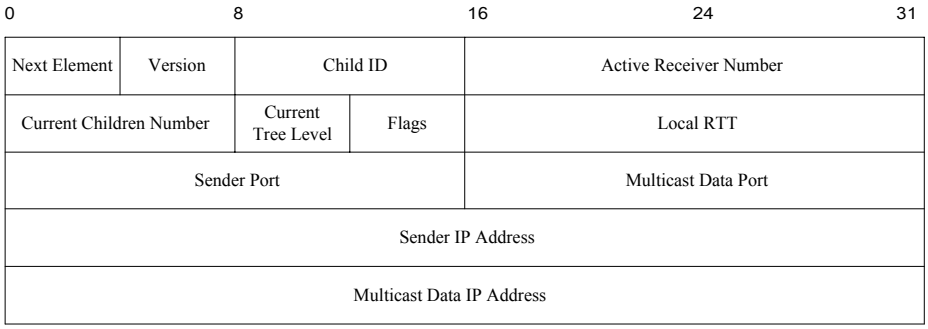


Figure 13 – Tree Membership Element

The following fields are specified:

a) *Next Element* – indicates the type of the next element immediately following this element;

b) *Version* – defines the version of this element usage. It is set to '1' at present;

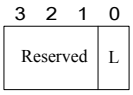
c) *Child ID* – specifies the ID number of a child, which is assigned by its parent in the tree configuration;

d) *Active Receiver Number* (*ARN*) – is the number of active descendants. Each LE sets the *ARN* to '1', and the parent LO aggregates *ARN* values for its children;

e) *Current Children Number* (*CCN*) – is the number of active children for an LO. Each LE sets *CCN* to '0';

f) *Current Tree Level* (*CTL*) – specifies the current tree level. TO is in the level 0, and its children are in the level 1. The *CTL* value is increased by '1' as the tree grows;

g) *Flags* – consists of the following fields:



1) *L* – is a bit flag indicating that the receiver is an LO (1) or LE (0). TO is an LO;

2) *Reserved* – is not defined yet, and reserved for future usage.

h) *Local RTT* – represents the round trip time for a local group in units of 10 milliseconds (see 8.4.5);

i) *Sender Port* – represents the port number of the ECTP sender (TO);

j) *Multicast Data Port* – represents the port number of the multicast data channel;

k) *Sender IP Address* – represents the IPv4 address of the ECTP sender (TO);

l) *Multicast Data Address* – represents the IPv4 address of the multicast data chanel.

9.2.3 Acknowledgment

This element provides the information on error, flow and congestion controls. The element structure is depicted below, which consists of the fixed 8-bytes and the variable-sized *Bitmap* that depends on *ACK Bitmap Size* (see 9.2.1).

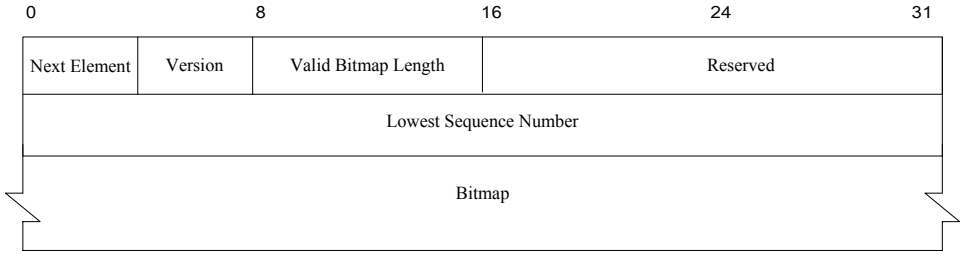


Figure 14 – Acknowledgment Element

The following parameters are specified:

a) *Next Element* – indicates the type of the next element immediately following this element;

b) *Version* – defines the version of this element usage. It is set to '1' at present;

c) *Valid Bitmap Length* – represents the length of the valid bitmap;

d) *Reserved* – is reserved for future usage;

e) *Lowest Sequence Number* (*LSN*) – is the sequence number of the lowest numbered data packet not yet received;

f) *Bitmap* – represents which data packets have been lost. It contains *Valid Bitmap Length* bits, starting from the *LSN* sequence number. The invalid bits in the *Bitmap* are set to '0'.

9.2.4 Timestamp

The Network Timestamp Protocol (NTP) is used for specifying timestamps (see IETF RFC 1119). NTP timestamps are represented as a 64-bit unsigned fixed-point number. The integer part is in the first 32 bits and the fraction part in the last 32 bits. If the NTP system is not used, ECTP uses the timestamp calculation algorithm TCP does. In this case, only the first 32 bits for the integer part will be valid. A flag bit is employed to indicate which timestamp mechanism is used. The flag bit is set to '0' for use of TCP, while it is set to '1' for use of NTP.

The structure of the timestamp element is depicted below.

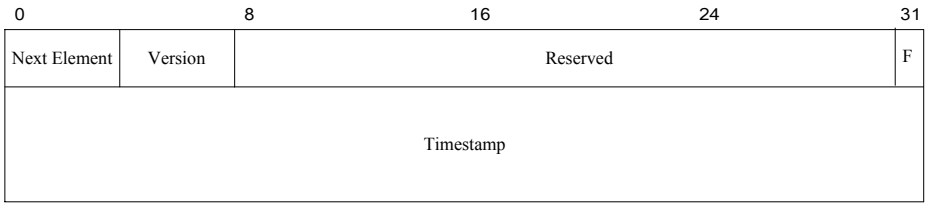


Figure 15 – Timestamp Element

The following fields are specified:

a) *Next Element* – indicates the next element type immediately following this timestamp element;

b) *Version* – defines the version of this element usage. It is set to '1' at present;

c) *F* – is set to '0' for use of the TCP timestamp mechanism, while it is set to '1' for use of NTP;

d) *Reserved* – is not defined yet, and reserved for future usage; and

e) *Timestamp* (in 8 bytes) – contains the timestamp value.

9.3 Packet structure

The encoding value and extension elements for each packet are shown in Table 3. Those extension elements are attached to the fixed header in the order of the connection information, tree membership, acknowledgement, and timestamp elements, if any.

Table 3 – Encoding and Extension Elements for ECTP Packets

Packet Type	Encoding	Extension Element				Data
		Connection Information	Tree Membership	Acknowledgement	Timestamp	
CR	0000 0001	O				
CC	0000 0010		O			
TJ	0000 0011					
TC	0000 0100		O*			
DT	0000 0101					O
ND	0000 0110					
RD	0000 0111					O
ACK	0000 1000		O	O	O	
HB	0000 1001		O		O	
JR	0000 1010					
JC	0000 1011	O*				
LR	0000 1100					
CT	0000 1101					

Note - In the table, O* means that the element is attached only if the corresponding request is accepted.

Note that this table provides just a guideline for the packet structure. In the future, as the protocol version grows, the mapping of the extension elements to the packet type is subject to change.

9.3.1 Creation request (CR)

TO creates an ECTP connection by sending a CR packet to all the receivers over the multicast data address. The CR packet format is as follows:

CR = Fixed header + Connection Information element.

In the fixed header, the next element is encoded as '0001' to indicate the connection information element. The packet type is set to '0000 0001'. The destination port represents the group port number for multicast data transmission to all the receivers. Again, in the enrollment phase, this port number MUST be announced to all the receivers (see 7.3.2). The source port is a local port number of TO for the connection, which will be used as the destination port number for unicast transmission by the children of TO or the late joiners. The sequence number is set to *ISN* that is assigned by the TO (see 8.3.2). This lets each receiver know the sequence number of the first DT packet that will be transmitted. The payload length is set to 8 (bytes) for the connection information element. The *F* bit is ignored.

In the connection information element, TO must specify all the fields with inputs from the application user, or with the default values.

9.3.2 Creation confirm (CC)

In response to the CR packet, each receiver sends a CC packet to its parent by unicast. The CC packet format is:

CC = Fixed header + Tree Membership elements.

In the fixed header, the next element is encoded as '0011' to indicate the tree membership element. The packet type is set to '0000 0010'. In the fixed header, the destination port is the local port number of the parent, which is the source port number contained in the corresponding CR (for parent TO) or HB packet (for parent LO). The source port of the CC packet is a local port number of the receiver.

In the tree membership element, all the fields other than Local RTT are filled with the information obtained in the tree configuration. The specific value depends on the node type: LO or LE.

9.3.3 Tree join request (TJ)

A receiver joins a control tree by sending a TJ packet to an on-tree LO during the tree configuration phase. The TJ packet format is as follows:

TJ = Fixed header.

The destination port is the local port number of the on-tree LO, which is contained in the corresponding HB packet. The source port is a local port number of the receiver.

9.3.4 Tree join confirm (TC)

An on-tree TO or LO responds with a TC packet to the TJ packet. The TC packet format is as follows:

TC = Fixed header + Tree Membership element; or

TC = Fixed header.

In the fixed header, the destination port is the port number of the receiver that sent the TJ packet, and the source port is the local port number of the on-tree parent. The flag bit *F* is set to '0' for acceptance or '1' for rejection.

If the tree join request is accepted, the TC packet contains a tree membership element. The *Child ID* and *CTL* fields MUST be specified.

9.3.5 Data (DT)

TO transmits a multicast stream with DT packets. The DT packet format is as follows:

DT = Fixed header + Data part.

The destination port is the port number of the multicast group. The source port is the local port number of TO. Each new DT packet is sequentially numbered in the sequence number field. The *Payload Length* is filled with the length of the data part in bytes. For the last data packet in a multicast stream, the *F* bit of the fixed header is set to '1'.

9.3.6 Null data (ND)

When TO has no data to transmit or it is in the connection pause period, it transmits an ND packet every *Null Data Time* (*NDT*) interval. The ND packet format is as follows:

ND = Fixed header;

The destination port is the port number of the multicast group. The source port is the local port number of TO. The sequence number field is filled with the sequence number of the DT packet most recently transmitted. In the connection pause period, the *F* bit of the fixed header is set to '1'. In the other case, it is set to '0'.

9.3.7 Retransmission data (RD)

According to the retransmission request via an ACK packet, each parent LO transmits the RD packets to its children over the multicast control address. The RD packet format is as follows:

RD = Fixed header + Data part;

The destination port is the port number of the multicast control address (see 7.3.3), and the source port is the local port number of the LO. The other fields are the same as those of the corresponding DT packet.

9.3.8 Acknowledgement (ACK)

Each child sends an ACK packet to its parent by unicast. The ACK packet format is as follows:

ACK = Fixed header + Tree Membership + Acknowledgement + Timestamp elements.

In the fixed header, the destination port is the local port number of the parent LO, and the source port is the local port number of the child.

The tree membership element MUST specify the *Child ID*, *ARN* and *Flags* fields.

All the fields in the acknowledgement element MUST be specified.

The timestamp element contains the timestamp value for the *Local RTT* (see 8.4.5).

9.3.9 Heartbeat (HB)

Each parent LO sends a HB packet to its children every *Heartbeat Generation Time* (*HGT*) interval.

The HB packet format is as follows:

HB = Fixed header + Tree Membership + Timestamp element.

In the fixed header, the destination port is the port number of the multicast control address (see 7.3.3), and the source port is the local port of the LO.

The tree membership element MUST specify the *CCN*, *CTL* and *Local RTT* fields.

The timestamp value is the time when the HB packet is transmitted.

9.3.10 Late join request (JR)

A late-joiner can join the connection by sending a JR packet to TO. The JR packet format is as follows:

JR = Fixed header.

The destination port is the port number of TO, and the source port is the local port number of the late-joiner.

9.3.11 Late join confirm (JC)

TO responds with a JC packet to the JR packet. The JC packet is as follows:

JC = Fixed header + Connection Information element; or

JC = Fixed header.

In the fixed header, the destination port is the local port number of the late-joiner, and the source port is the local port number of TO. The flag bit *F* is set to ‘0’ for acceptance or ‘1’ for rejection.

If the tree join request is accepted, the JC packet contains the connection information element.

9.3.12 Leave request (LR)

When a child wants to leave the connection, it sends an LR packet to its parent. The LR packet is also used for a parent to eject a trouble-making child. The LR packet format is as follows:

LR = Fixed header.

The destination port is the local port number of the parent LO or a trouble-making child, and the source port is a local port number of the leaving child or the parent LO, which depends on who triggers the LR packet.

The *F* bit of the fixed header is set to '1' for the user-invoked leave (see 8.7.1), or set to '0' for the troublemaker ejection (see 8.7.2)

9.3.13 Connection termination (CT)

TO terminates the connection by sending a CT packet to all the receivers. The CT packet format is as follows:

CT = Fixed header.

The destination port is the group port number for multicast data address (see 7.4). The source port is a local port number of TO.

The *F* bit of the fixed header is set to '1' for an abnormal termination, or set to '0' for normal termination after all the data have been transmitted (see 8.9)

10. Timers and variables

10.1 Timers

All the timers specified in ECTP are defined in units of 10 milliseconds. In the implementation, each timer may be employed in units of 50 or 200 milliseconds, depending on the number of the ticks per second in the system.

a) ACK Generation Time (*AGT*): Each child sends an ACK packet to its parent if *AGT* timer expires. Each receiver reactivates this timer each time it generates an ACK packet. The specific *AGT* value depends on the implementation.

b) Connection Creation Time (*CCT*): The connection creation time is constrained by *CCT* timer, which is specified by TO (see 8.2). *CCT* is represented in 10 milliseconds.

c) Heartbeat Generation Time (*HGT*): Each parent sends a HB packet to its children every *HGT* interval (see 8.8). *HGT* may be set to a multiple of *AGT*.

d) Inactivity Time (*IAT*): If a node has not received any packet during *IAT* interval, it shall determine that the network is disconnected. Each node refreshes its *IAT* timer, each time it receives a packet. The *IAT* value depends on the implementation (see 8.2).

e) Null Data Time (*NDT*): When TO has no data to transmit or it is in the connection pause period, it sends a ND packet every *NDT* time interval. *NDT* depends on the implementation (see 8.3).

f) Retransmission Back-off Time (*RBT*): After a parent retransmits a lost data packet requested by its child, it activates the *RBT* timer (see 8.4.6). Retransmission requests for the same data packet will be ignored while the timer is valid.

g) Retransmission Time (*RXT*): A node activates the *RXT* timer after it transmits a control packet by unicast. If the responding control packet has not arrived before the *RXT* timer expires, then the control packet is retransmitted. The specific *RXT* value depends on the implementation (see 8.2.2 and 8.6).

h) Tree Creation Time (*TCT*): In the tree creation, each LO activates its *TCT* timer to complete the tree configuration for its local group. The *Tree Creation time* is calculated by using *CCT* (see 8.2.2).

10.2 Operation variables

a) ACK Generation Number (*AGN*): Each child generates an ACK packet for *AGN* data packets (see 8.4.3). The *AGN* number is set to a 1/4 of *ACK Bitmap Size*.

b) Active Receiver Number (*ARN*): The *ARN* is specified in the tree membership element (see 8.4.4 and 9.2.2).

c) Current Children Number (*CCN*): The *CCN* is the number of active children for a local group (see 9.2.2). The value is calculated by a parent, and announced to the children via the tree membership element. In the tree configuration, a non-tree child may refer to this value to select the best on-tree parent LO.

d) Current Tree Level (*CTL*): The *CTL* represents the current tree level, in which a receiver is located in the tree. This value is increased each time a new tree branch is created (see 8.2.2). The *CTL* value is specified in the tree membership element (see 9.2.2).

e) Maximum Children Number (*MCN*): The number of children that a parent has is constrained by *MCN*. To ensure a desirable growth of the tree, a portion of *MCN* can be reserved for some child LOs (see 8.2.2). The *MCN* value is specified in the connection information element.

f) Maximum Retransmission Number (*MRN*): The number of retransmissions for the control and RD packets is bounded by *MRN*. The specific value depends on the implementation (see 8.4.6 and 8.6).

g) Maximum Tree Level (*MTL*): The number of tree levels for the ECTP control tree is constrained by *MTL*, which is specified in the connection information element (see 8.2.2 and 9.2.1).

h) Node Failure Threshold (*NFT*): A tree node failure is indicated by the *NFT* value (see 8.8.3). The specific value depends on the implementation.

Annex A. Network considerations

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

ECTP is a transport layer protocol that runs over IP networks. The multicast transport protocols assume that the underlying networks have IP multicast forwarding capability. Many multicast routing protocols have so far been proposed. Those protocols can be classified into the source based tree protocols and the shared tree protocols. The source based tree protocols include the DVMRP, MOSPF, PIM-DM and SSM. The shared tree protocols include the CBT, PIM-SM and BGMP.

Since ECTP is an end-to-end transport protocol, its protocol mechanisms are designed to be independent of any specific multicast routing protocol. However, the use of an IP multicast address will be affected by the underlying multicast routing protocols. Note in the ECTP simplex multicast connection that TO and LOs are required to provide the multicast transmission for their receivers or children, and thus they will use one or more IP multicast addresses (see 7.3.3). Based on the current IP multicast routing protocols, some possible scenarios for use of multicast addresses in an ECTP connection are described as follows:

1) When a shared tree protocol such as CBT or PIM-SM is deployed in the networks, a single multicast address is shared by TO and LOs. In this case, the TTL-scoped multicasting is required for each parent to restrict its multicast control traffic such as HB and RD packets within its local group;

2) In the shared or source based tree protocols, TO or LO uses its own multicast address. In this scenario, TO transmits multicast data over its multicast data address, while each LO sends control packets such as HB and RD packets over its multicast control address. Therefore, two or more multicast addresses will be used for the ECTP connection; and

3) In the SSM (Source Specific Multicast) routing protocol, which has been perceived as a promising solution for IP multicast routing by many ISP, a multicast channel or address is defined as a pair of a multicast address (G) and a unicast address of the source (S). That is, a pair of (S, G) defines a multicast channel. If SSM is used in the network, TO and LOs share a single multicast address, while the multicast channels of TO and LOs can be identified by their source address together with the common multicast address.

In any case, the information on the multicast transport addresses, including IP multicast addresses and port numbers, MUST be announced to all the receivers in the enrollment phase.

Annex B. Tree configuration mechanisms considered in IETF RMT WG

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

ECTP provides three options for tree configuration (see 8.2.2). This Annex describes a brief sketch for the tree configuration mechanisms proposed in IETF RMT WG. Some of them may be incorporated into the ECTP specification as a candidate tree creation option in the future.

The IETF has proposed four tree configuration schemes, in which a control tree is built in a "bottom-up" or receiver-initiated fashion. Each node initiating a tree join request uses its metric(s) to the sender to make a decision as to which node is the best parent. According to the metrics employed or the methods for obtaining the metrics, the four options are proposed.

Tree construction, regardless of metric, proceeds generically as follows.

1) Receivers of a session use standard out-of-band mechanisms for discovery of a session's existence via SAP or HTTP. In this way, a receiver discovers the multicast group address, the sender's address, and other information necessary for logical tree construction.
2) Each receiver then determines the best parent LO to use for the session, and binds with it for service. If a receiver is also an LO, then it may use mechanisms to find an LO for itself.
3) All LOs must determine their distance to the sender using the metric required for the session.
4) Once an LO determines its own metric to the sender, it discovers other potential parents and their metrics as well. In this way, the LO can make a choice as to where it should graft itself into the control tree.
5) When an appropriate parent LO has been chosen, an LO must bind to the chosen parent. Once an LO receives a bind from a child, then the LO must also bind with other LOs in order to form the control tree (rooted back to sender).
6) During a session, a receiver or LO may change to a different parent LO for some reasons.

Depending on the tree creation option, a different metric may be employed. Metric to the sender is used to rank and determine which neighbor is an appropriate parent.

a) Static Metric

   A list of neighbors available to a receiver, optionally together with their distances, are provided by a well-known server or location. Each receiver determines the neighbors' distances, then picks the best one to be its parent LO.

b) Expanding Ring Search Metric

   Receivers learn their approximate distance to the sender through a beacon of the sender (e.g., a CR packet in ECTP). Once this metric is learned, it can be advertised to neighbors through the use of an expanding ring search (ERS). Each receiver multicasts its request by using an ERS. On-tree LOs respond to the requests of receivers with their own TTL from the sender. Each receiver increases the TTL of its requests until a response is received, then picks the best parent LO.

c) Routing Metric

   Nodes learn their distance from the source through GRA (Generic Router Assist). Receivers have pre-configured the relationships with potential parents, based on the underlying multicast routing tree.

d) Point-of-Contact Metric

   Nodes learn their distance from the source, then query a designated node, the POC, for neighbors using this distance. The POC returns one or more choices of parent LOs from which the node chooses.

Bibliography

The following IETF RFCs are useful to understand this ECTP specification.

IETF RFC 768, User Datagram Protocol, *Internet Standard*, August 1980

IETF RFC 791, Internet Protocol, *Internet Standard*, September 1981

IETF RFC 793, Transmission Control Protocol, *Internet Standard*, September 1981

IETF RFC 1112, Host Extensions for IP Multicasting, *Internet Standard*, August 1989

IETF RFC 1119, Network Time Protocol, *Internet Standard,* May 1990

IETF RFC 2119, Key Words for Use in RFCs to Indicate Requirement Levels, *Best Current Practice*, March 1997

IETF RFC 2236, Internet Group Management Protocol, Version 2, *Proposed Standard,* November 1997

IETF RFC 2327, SDP: Session Description Protocol, *Proposed Standard,* April 1998

IETF RFC 2362, Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, *Experimental,* June 1998

IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, *Draft Standard,* December 1998

IETF RFC 2887, The Reliable Multicast Design Space for Bulk Data Transfer, *Informational*, August 2000

IETF RFC 2974, SAP: Session Announcement Protocol, *Experimental,* October 2000

IETF RFC 3048, Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer, *Informational*, January 2001

ICS 35.100.40

Price based on 33 pages