

I n t e r n a t i o n a l T e l e c o m m u n i c a t i o n U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.135

(08/2007)

SERIES T: TERMINALS FOR TELEMATIC SERVICES
Data protocols for multimedia conferencing

User-to-reservation system transactions within T.120 conferences

ITU-T Recommendation T.135



International
Telecommunication
Union

Documents
Document Sharing Hub

ITU-T T-SERIES RECOMMENDATIONS
TERMINALS FOR TELEMATIC SERVICES

Facsimile – Framework	T.0–T.19
Still-image compression – Test charts	T.20–T.29
Facsimile – Group 3 protocols	T.30–T.39
Colour representation	T.40–T.49
Character coding	T.50–T.59
Facsimile – Group 4 protocols	T.60–T.69
Telematic services – Framework	T.70–T.79
Still-image compression – JPEG-1, Bi-level and JBIG	T.80–T.89
Telematic services – ISDN Terminals and protocols	T.90–T.99
Videotext – Framework	T.100–T.109
Data protocols for multimedia conferencing	T.120–T.149
Telewriting	T.150–T.159
Multimedia and hypermedia framework	T.170–T.189
Cooperative document handling	T.190–T.199
Telematic services – Interworking	T.300–T.399
Open document architecture	T.400–T.429
Document transfer and manipulation	T.430–T.449
Document application profile	T.500–T.509
Communication application profile	T.510–T.559
Telematic services – Equipment characteristics	T.560–T.649
Still-image compression – JPEG 2000	T.800–T.849
Still-image compression – JPEG-1 extensions	T.850–T.899

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

ITU-T Recommendation T.135

User-to-reservation system transactions within T.120 conferences

Summary

ITU-T Recommendation T.135 introduces a protocol that enables a reservation application in a user terminal to communicate electronically with the reservation system a service provider, in order to reserve resources for multimedia conferences, modify or cancel previous reservations, and perform other related transactions. The architecture model defined by this Recommendation is designed for in-band transactions – that is, transactions taking place between a reservation system and a T.120 compliant user terminal joined to an ongoing conference.

This revised version of T.135 introduces a number of clarifications to the previous version.

Source

ITU-T Recommendation T.135 was approved on 29 August 2007 by ITU-T Study Group 16 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

© ITU 2008

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

1	Scope	1
2	References.....	1
3	Definitions	2
4	Abbreviations.....	3
5	Conventions	4
6	Introduction	5
7	T.135 model.....	5
	7.1 System model	5
	7.2 Mode of operation	7
8	Service description.....	17
	8.1 Service summary	17
	8.2 Service description	18
9	Protocol specification	49
	9.1 Encoding of URST PDUs.....	49
	9.2 URST ASN.1 Module	50

ITU-T Recommendation T.135

User-to-reservation system transactions within T.120 conferences

1 Scope

This Recommendation provides a standard protocol for multimedia conference reservation transactions between users and service providers, whilst users are utilizing a T.120-based terminal.

Its primary intent is to enable conferencing users to perform operations such as reserving resources for their next conference, or directly interacting from the conference they are participating in to modify the conference's resource use.

The architecture model and the protocol specified in this Recommendation make provision of a collection of parameters which allow conferencing users to deal with multiple service providers simultaneously.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- [ITU-T H.231] ITU-T Recommendation H.231 (1997), *Multipoint control units for audiovisual systems using digital channels up to 1920 kbit/s.*
- [ITU-T H.242] ITU-T Recommendation H.242 (2004), *System for establishing communication between audiovisual terminals using digital channels up to 2 Mbit/s.*
- [ITU-T H.243] ITU-T Recommendation H.243 (2005), *Procedures for establishing communication between three or more audiovisual terminals using digital channels up to 1920 kbit/s.*
- [ITU-T H.261] ITU-T Recommendation H.261 (1993), *Video codec for audiovisual services at $p \times 64$ kbit/s.*
- [ITU-T H.262] ITU-T Recommendation H.262 (2000), *Information technology – Generic coding of moving pictures and associated audio information: Video.*
- [ITU-T H.320] ITU-T Recommendation H.320 (2004), *Narrow-band visual telephone systems and terminal equipment.*
- [ITU-T Q.931] ITU-T Recommendation Q.931 (1998), *ISDN user-network interface layer 3 specification for basic call control.*
- [ITU-T T.50] ITU-T Recommendation T.50 (1992), *International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange.*
- [ITU-T T.120] ITU-T Recommendation T.120 (2007), *Data protocols for multimedia conferencing.*
- [ITU-T T.122] ITU-T Recommendation T.122 (1998), *Multipoint communication service – Service definition.*

- [ITU-T T.123] ITU-T Recommendation T.123 (2007), *Network specific data protocol stacks for multimedia conferencing*.
- [ITU-T T.124] ITU-T Recommendation T.124 (2007), *Generic Conference Control*.
- [ITU-T T.125] ITU-T Recommendation T.125 (1998), *Multipoint communication service protocol specification*.

3 Definitions

This Recommendation defines the following terms:

- 3.1 application protocol:** Any standard or non-standard protocol utilized by an application.
- 3.2 application protocol entity:** The name given to any instance of an application protocol within the frame of the T.120 communication infrastructure.
- 3.3 conference resource:** Some part of a multimedia conferencing device that can be used in a conference.
- 3.4 multimedia conference:** A number of multimedia conferencing devices that are joined together and that are capable of exchanging audiographic and audiovisual information across various communication networks. In this Recommendation, a multimedia conference is assumed to be T.120-based for some or all of its data part.
- 3.5 multimedia conferencing device:** A device used in a multimedia conference which may include terminals, multiport terminals, MCUs and other machines.
- 3.6 multipoint:** The ability to exchange any or all of audio, video or data among multiple multimedia conferencing devices simultaneously as opposed to point-to-point where audio, video and/or data is exchanged between two directly connected nodes.
- 3.7 multipoint control unit:** Commonly referred to as an MCU or bridge; a multiport device that serves to connect terminals and other MCUs in a multipoint fashion. An MCU is not primarily intended as an end-point for user communication.
- 3.8 multiport terminal:** A user terminal also capable of bridging T.120 information. Refer to clause 3 of [ITU-T T.124] for the full definition of a multiport terminal.
- 3.9 registered site:** Given a particular reservation system, a registered site is a site which is known by the reservation system (its description is stored in (one of) the reservation system database(s)), and that can be unambiguously identified by it by the means of its *Site ID*.
- 3.10 reservation application:** An application utilizing a T.135 APE.
- 3.11 reservation connection:** A data point-to-point connection established between a client and a server reservation application and by the means of which a user has logged into a reservation system.
- 3.12 reservation device:** Generic name given to components of a reservation system when it is designed as a distributed architecture.
- 3.13 reservation domain:** The set of multimedia conferencing devices managed by a reservation system. The term "management" is used in the sense of direct control of scheduling resources and, for some cases, real-time control of conferencing devices.
- 3.14 reservation server node:** A special reservation device that is capable of representing the service provider which device is a component in a multimedia conference.
- 3.15 reservation system:** A system capable of managing the entire related reservation domain.
- 3.16 reservation transaction:** An interaction between two reservation applications which is globally based on a query/response model.

3.17 service provider: An organization owning or controlling one or more reservation systems and using them to provide multimedia conference bridging services.

3.18 service subscriber: A person, an organization or any intermediate entity known [i.e., recorded in (one of) the database(s)] by a reservation system and recognized by it as a *registered* user of the service provider. The subscriber is unambiguously identified within the reservation system of the service provider reservation system by means of its *Subscriber ID*.

3.19 service user: A person, an organization or any intermediate entity using the services provided by a service provider.

3.20 site: A global piece of information describing a member of a multimedia conference. Site information includes information relative to the multimedia conferencing device used, to the network to which the device is directly connected [including the device network address(es)], to the physical location of the device, and to the human participants using this device.

3.21 site ID: An unambiguous identification by which a site is recognized as a registered site by a reservation system.

3.22 subscriber ID: An unambiguous identification by which a user is recognized as a service subscriber by a reservation system.

3.23 user terminal: The terminal that a user actually utilizes to participate into multimedia conferences.

4 Abbreviations

This Recommendation uses the following abbreviations:

APE	Application Protocol Entity
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
CSDN	Circuit Switched Digital Network
GCC	General Conference Control
GSTN	Global Switched Telephone Network
GUI	Graphical User Interface
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MCE	Multimedia Conferencing Equipment
MCS	Multipoint Communication Service
MCU	Multipoint Control Unit
PDU	Protocol Data Unit
PSTN	Public Switched Telephone Network
RD	Reservation Device
RSN	Reservation Server Node
RSN-RA	Reservation Server Node Reservation Application
URST	User-to-Reservation System Transactions

UT	User Terminal
UT-RA	User Terminal Reservation Application

5 Conventions

The primitive parameters of the abstract services defined in this Recommendation use the following key:

M	Parameter is mandatory.
C	Parameter is conditional.
O	Parameter is optional.
Blank	Parameter is absent.
(=)	Value of the parameter is identical to the value of the corresponding parameter of the preceding primitive, where preceding is defined relative to the order: request, indication, response, confirm.
(=RQ)	Value of the parameter is identical to the value of the corresponding parameter in a preceding primitive, where RQ = request, IN = indication, RS = response, and CF = confirm.

Service primitives are categorized in up to four types: Request, Indication, Response, and Confirm. Some primitives support all of these types, while others do not. These four types are defined as follows:

Request primitives:	Those that are sourced from a Reservation Application to a URST APE to initiate a URST transaction or command a specific service.
Indication primitives:	Those that are sourced from a URST APE to a Reservation Application either as a result of a received Request or Indication PDU or a URST APE initiated action.
Response primitives:	Those that are sourced from a Reservation Application to a URST APE in response to an Indication primitive which requires a response.
Confirm primitives:	Those that are sourced from a URST APE to a Reservation Application as a result of a received Response PDU or directly in response to a Request primitive.

PDUs are categorized into three types. PDU names all include the words "Request", "Indication", or "Response" to indicate the intended use of the PDU. These are defined as follows:

Request PDUs:	Those that require a Response PDU in return. A URST APE receiving a Request PDU from the peer side generates an Indication service primitive toward the Reservation Application concerned.
Indication PDUs:	Those that do not require a response (e.g., those that are for informational purposes). A URST APE receiving a Request PDU from the peer side generates an Indication service primitive toward the Reservation Application concerned.
Response PDUs:	Those which are in response to a particular Request PDU. A Response PDU is sent to the peer side by an URST APE either automatically or as a result of a corresponding Response service primitive passed by a Reservation Application. A URST APE receiving a Response PDU from the peer side generates a Confirm service primitive toward the Reservation Application concerned.

6 Introduction

In order to book a multimedia conference, a user may utilize a T.120 terminal (abbreviated by the term UT as *User Terminal* in this Recommendation) to reserve the necessary resources and indicate the facilities needed at conference runtime. This Recommendation provides an application protocol specification and a related service description enabling a reservation application within a UT to interact with a reservation application within a service provider entity referred to as the *Reservation System* in this Recommendation. Although the protocol described in this Recommendation is processed by a T.120 *Application Protocol Entity* (APE) which makes use of MCS and GCC services, it may be used in other communication infrastructures. However, this issue is out the scope of this Recommendation.

The *User-to-Reservation System Transactions* (URST) service supplied by a URST APE is connection-oriented in the sense that it allows a user identified by a subscriber identity to log into a reservation system. It enables the user to schedule conferences and modify or cancel previous reservations. Conferences can be scheduled over one or more MCUs controlled by a single reservation system, but the user is free to introduce "external" MCUs in the list of sites defining the conference. The URST service also makes provision of querying facilities such as listing and reading scheduled and ongoing conferences, inquiring of resources availability and accessing sites directory service.

Beside the set of functions defined by this Recommendation, the URST protocol and service also provision means for proprietary dialogues between a user and a service provider.

This Recommendation does not specify any graphic user interface since reservation applications developed for UTs are assumed to provide that functionality. In other words, the URST service description refers to the services offered by the URST protocol, how these are used by an application to present a set of services to a user is outside the scope of this Recommendation.

7 T.135 model

The model assumes that the UT utilized to access the reservation system is T.120-compliant and involved in an ongoing T.120 conference. Although it is assumed this conference is a users conference in the common sense of the term, the model does not preclude service providers from operating special T.120 conferences dedicated to providing access to reservation services for users. This issue is considered a service provider option and is outside the scope of this Recommendation.

7.1 System model

The reservation system is pictured as a functional entity which processes and responds to users conference reservation requests, that is search, allocate and control the necessary *Multipoint Control Unit* (MCU) and other network resources. It is assumed that the reservation system includes T.120-compliant electronic endpoints, called *Reservation Server Nodes* (RSNs), which participate in user conferences and by this means provide access to reservation services. Apart from RSNs, the exact architecture of the reservation system is not specified by this Recommendation. It may be more or less complex and consist of one or more electronic devices as well as human operators. However, it should be noted that the URST service and protocol allow users to interact over ongoing conferences, which intuitively suggests an automated process.

From an architectural standpoint, the RSN can be an internal part (or a proxy agent) of an MCU or multipoint terminal operated by the service provider, or any management device integrally or partially dedicated to the purpose of reservation. Those possibilities are choices of the service provider and left outside the scope of this Recommendation. In any case, from the reservation terminal viewpoint, the RSN is seen as the reservation system. In this Recommendation, non-RSN components of a reservation system are referred to as *Reservation Devices* (RDs).

Figure 7-1 shows examples of configuration. RSNs are indicated by gray boxes. Thick lines represent T.120 links while dotted ones correspond to unspecified ones. Case a) illustrates the case of the RSN being (located within) an MCU. Case b) illustrates the case of the RSN being (located within) a dedicated equipment.

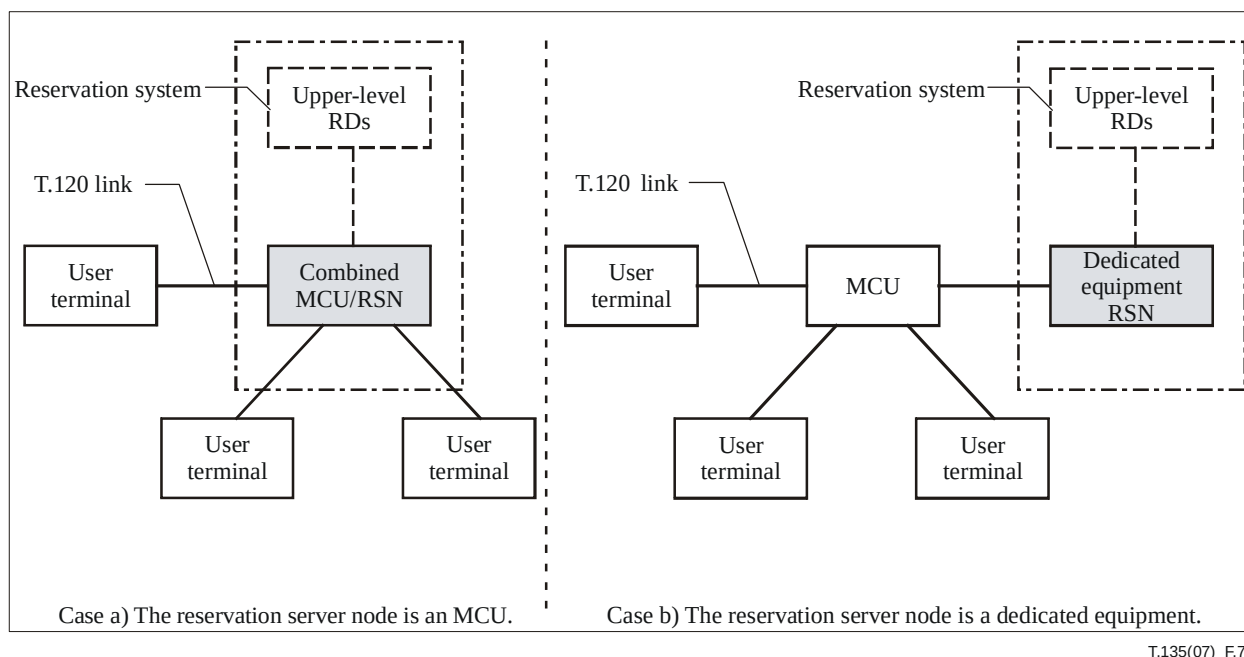


Figure 7-1 – The reservation server node in the system model

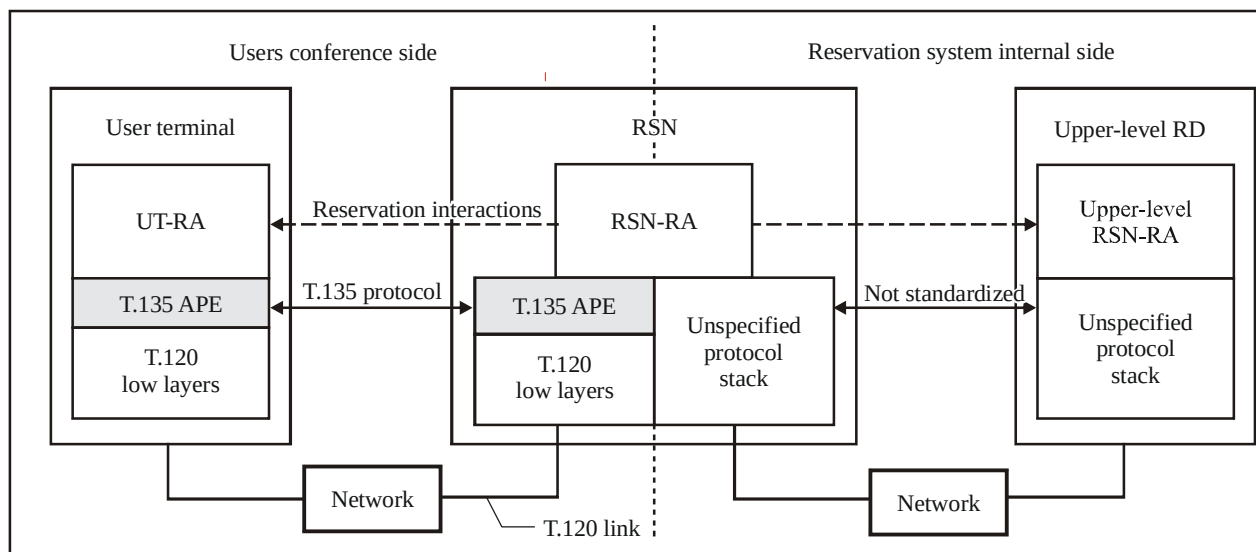
Although for simplicity Figure 7-1 pictures a single RSN participating into the users conference, it should be noted that the model allows more than one RSN in the conference. Further, distinct RSNs representative of distinct reservation systems (and therefore distinct service providers) may be joined to a users conference.

The RSN is a T.120 device that includes a *Reservation Server Node Reservation Application* (RSN-RA) which makes use of the protocol operated by a URST APE. By means of the URST protocol, the RSN-RA dialogs with a *User Terminal Reservation Application* (UT-RA) located in the UT that also relies on an URST APE. Depending on implementation choice, a URST APE can be an independent entity offering its services to the reservation application(s), or integrated within a reservation application. The delivery to a reservation application by the URST APE of the information contained in URST *Protocol Data Units* (PDUs) is modelled by the URST service primitives specified in this Recommendation. The model assumes that the URST APE is responsible for communicating with the underlying MCS and GCC entities.

The role played by the RSN-RA depends on the reservation system architecture. Unless it is completely autonomous, the RSN will bridge the users conference domain toward some upper level RD that contains an entity modelled as an upper-level RSN-RA capable of processing reservation transactions. The reservation protocol and the transport stack used between the RSN and the upper-level RD are outside the scope of this Recommendation. Whatever role is performed by the RSN-RA in combination with upper-level RSN-RAs of upper-level RDs, the RSN shall project to the users conference as a whole compliant with the set of functions defined by the URST protocol.

Figure 7-2 shows the communication infrastructure model of T.135. Gray boxes represent the parts that are subject to this Recommendation, while white boxes are parts outside its scope. A dashed line shows the virtual frontier between the user's conference domain and the back side of the RSN. For simplicity, Figure 7-2 pictures the reservation system internal side with a single upper-level RD.

The dotted arrowed line shows the virtual reservation dialog that takes place between an upper-level RSN-RA within it and the UT-RA within the user's terminal. In a real configuration, the internal side of the reservation system may include a chain of several RDs.



T.135(07)_F.7.2

Figure 7-2 – System model and scope of T.135

7.2 Mode of operation

Interactions between a UT-RA and an RSN-RA are of a point-to-point nature and modelled as query/answer operations, referred to as *Reservation Transactions* in this Recommendation. Reservation transactions take place through a logical object named a *Reservation Connection*. In this Recommendation the UT-RA is always the originator of reservation connections, as well as the querying side of all defined reservation transactions. However, provision is made to allow other schemes in future revisions.

A reservation connection consists of a successful attempt of the UT-RA to "log" into the reservation system under a *Subscriber ID*. Reservation transactions can take place once the following actions have been performed in sequence:

- the UT-RA in the UT has recognized the RSN-RA;
- the UT-RA has successfully opened a reservation connection with the RSN-RA.

A URST APE makes use exclusively of the *MCS-Send-Data* service for sending and receiving URST PDUs and each side sends data onto the peer's MCS UserID channel.

7.2.1 Identification of reservation applications

Once joined to the users conference, an RSN-RA that wishes to make itself available to other nodes shall first obtain an MCS User ID through MCS service *MCS-ATTACH-USER*, and then enrol in the URST *Registration Session* (refer to [ITU-T T.121]) through GCC service *GCC-Application-Enrol*. According to the guidelines of [ITU-T T.121], the enrolment in the registration session shall be done with the *Enrol/Un-enrol* flag set to "Enrol" and the *Active/Inactive* flag set to "Inactive". However, the RSN-RA **MUST** pass its MCS User ID through parameter *Application User ID* of primitive *GCC-Application-Enrol-request*.

In order to open a reservation connection, a UT-RA must follow the same process. However, since this version of T.135 defines reservation connections as always opened by UT-RAs, a UT-RA is free to wait until it needs to open a reservation connection for getting an MCS User ID and enrol in the registration session.

A reservation application shall use its associated *Non-Collapsing Application Capabilities* (refer to [ITU-T T.124]) to make peers aware of its type – that is, if it is located within an RSN representing a reservation system (i.e., a service provider) or within a UT. The standardized non-collapsing capabilities presented in Table 7-1 are defined for this purpose.

Table 7-1 – Standardized non-collapsing capabilities

Name	ID	Associated application data	Dependency
Reservation Application Type	0	One octet long, interpreted as an unsigned integer value that shall be set to: • 0 for RSN-RAs • 1 for user UT-RAs	Mandatory
Reservation System Name	1	Undefined length, interpreted as a string of characters encoded according to the rules of [ITU-T T.50].	Mandatory for RSN-RAs, irrelevant otherwise.
Reservation System ID	2	Octet string which length is comprised within the range [4...255], encoded as an ASN.1 value of the type <i>H221NonStandardIdentifier</i> as specified in [ITU-T T.124].	Optional for RSN-RAs, irrelevant otherwise.

All these capabilities and their assigned application data are initialization parameters to the reservation application. The list of RSN-RAs enrolled in the URST registration session and their associated standard capabilities can be retrieved by a UT-RA via the *RS-Check-Reservation-Systems* service.

The character string application data assigned to the *Reservation System Name* capability is intended to be presented to the user by the UT-RA, in order to allow him to identify the reservation system (i.e., the service provider). Therefore its content shall be human-interpretable.

The content of the octet string application data assigned to the *Reservation System ID* capability is intended to be machine-interpretable. Its purpose is to allow automation of reservation systems identification. By prior agreement, a service provider may indicate to a user an alphanumeric pattern identifying its reservation system. The user may then configure his UT-RA to automatically recognize and discriminate the pattern among others. The octet string shall contain an ASN.1 parameter of the type *H221NonStandardIdentifier* defined in [ITU-T T.124], the first four octets of its value field being the country and service provider code, similar to the specification given in Annex A of [ITU-T H.221] for *NS-cap* and *NS-comm* escape sequences.

When an RSN-RA is enrolled in the registration session, peer UT-RAs may attempt to open a reservation connection with it. An RSN-RA will make itself unavailable by un-enrolling the registration session.

7.2.2 Reservation connections

7.2.2.1 Connection handles and identifiers

The URST service allows a reservation application to handle multiple reservation connections simultaneously. Between a reservation application and the underlying URST APE, each opened reservation connection is locally identified by parameter *Connection Handle* of URST service

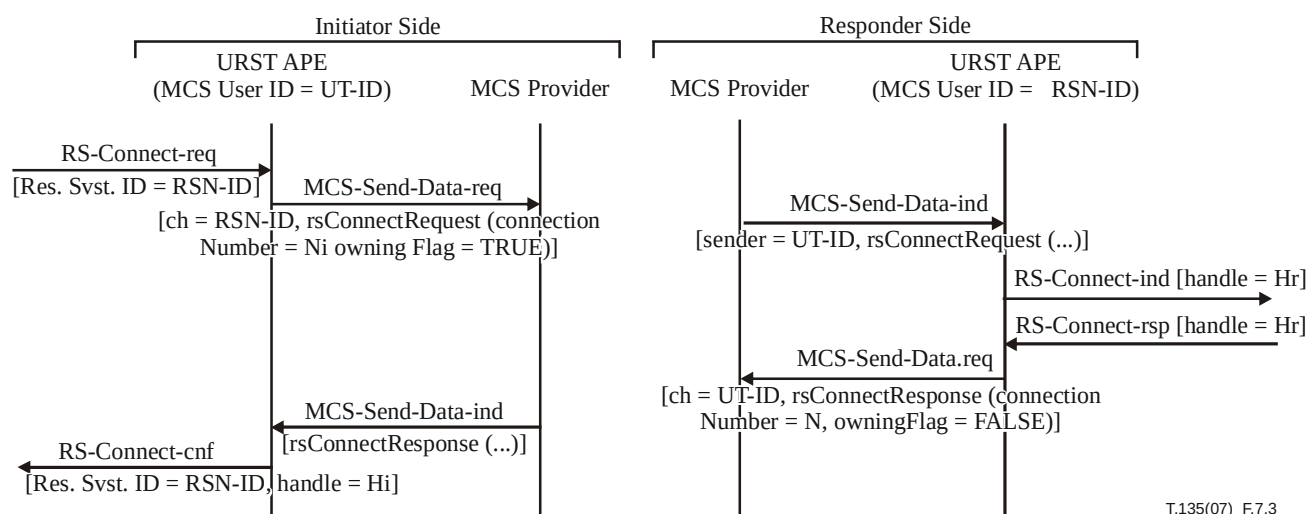
primitives. The format and mechanism used to construct connection handles is considered as a local matter by this Recommendation.

At the protocol level (i.e., between two distant URST APEs), the identification of a reservation connection is done through the parameter *connectionID* of URST PDUs. This parameter is of the ASN.1 type *ConnectionID* which consists in an integer comprised within the range [0...65535], the *connection number*, and a boolean flag, the *owning flag*. The owning flag indicates to the receptor of a URST PDU whether or not the connection number was allocated by the sender of the PDU (i.e., if the concerned reservation connection was opened by the sender of the PDU)¹.

The URST APE at the opening side of a reservation connection is responsible for allocating the connection number. At both sides of the connection, URST APEs are responsible of allocating the connection handle and maintaining the mapping with its correspondent connection number.

7.2.2.2 Opening a reservation connection

On reception of the *RS-Connect-request* service primitive from the reservation application, the URST APE at the initiator side shall first allocate a connection number and its associated connection handle. It shall then send an *rsConnectRequest* PDU toward the peer side, by transmitting it through the peer URST APE's MCS UserID channel, and construct the *connectionID* parameter with the connection number allocated and the owning flag set (i.e., indicating TRUE). The URST APE at the peer side shall respond by transmitting an *rsConnectResponse* PDU toward the initiator side through its MCS UserID channel, reusing the same reference number to construct the value of parameter *connectionID*, but unsetting the owning flag (i.e., indicating FALSE). The connection handle allocated at the opening side is indicated through parameter *Connection Handle* of primitive *RS-Connect-Response*, on reception of the *rsConnectResponse* PDU.



T.135(07)_F.7.3

Figure 7-3 – Opening a reservation connection

7.2.2.3 Authentication scenarios

URST service and protocol allow feeble authentication based on simple passwords or strong authentication schemes based on unidirectional or challenge/response algorithms.

The authentication scheme may consist of multiple passes, each consisting of an *rsConnectRequest* PDU from the initiator side to the responding side and an *rsConnectResponse* PDU from the

¹ This rule is intended to allow the definition of reservation system initiated connection in future revisions of this Recommendation.

responding side to the initiator side. When multiple passes are used, only parameter *resultCode* in the last *rsConnectResponse* PDU will indicate the success or failure of the connection process. Parameter *resultCode* in intermediate *rsConnectResponse* PDUs shall always be set to the value "authenticationInProgress".

Figure 7-4 shows the initialization sequence of a reservation connection.

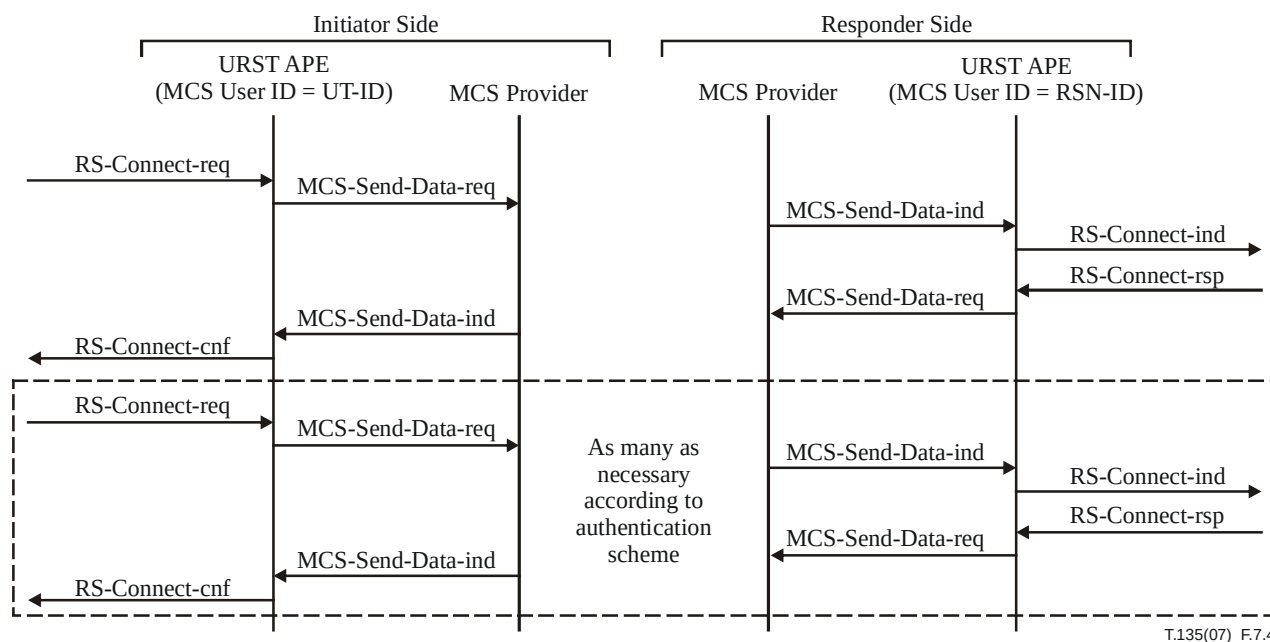


Figure 7-4 – Reservation connection initialization sequence

7.2.2.4 Closing a reservation connection

A reservation connection is closed when any of the UT-RA and the RSN-RA issues an *RS-Disconnect-request* service to its local URST APE, when any of the involved user terminal or RSN disconnects from the conference, or when the conference is terminated.

In the first case, the URST APE at the closing side shall send an *rsDisconnectIndication* PDU toward the peer side and drop the corresponding connection handle and protocol-level reference number. On receiving the *rsDisconnectIndication* PDU, the URST APE at the peer side will then generate an *RS-Disconnect-indication* primitive to the above reservation application, drop the corresponding connection handle.

In the other cases, each URST APE shall raise an *RS-Disconnect-indication* primitive to the local reservation application and drop the corresponding connection handle and protocol-level reference number.

7.2.3 Reservation transactions

7.2.3.1 Transaction identifiers and handles

The URST service allows a reservation application to handle multiple reservation transactions simultaneously. Between a reservation application and the underlying URST APE, each reservation transaction is locally identified by parameter *Transaction Handle* of URST service primitives. The format and mechanism used to construct transaction handles is considered as a local matter by this Recommendation.

At the protocol level (i.e., between two distant URST APEs), the identification of a reservation transaction is done through the parameter *transactionID* of URST PDUs. This parameter is of the ASN.1 type *TransactionID* which consists in an integer comprised within the range [0...65535], the

transaction number, and a boolean flag, the *owning flag*. The owning flag indicates to the receptor of a URST PDU whether or not the transaction number was allocated by the sender of the PDU (i.e., if the transaction was initiated by the sender of the PDU).

The URST APE at the initiator side of a reservation transaction is responsible for allocating the transaction number. At both sides of the transaction, URST APEs are responsible of allocating the transaction handle and maintaining the mapping with its correspondent transaction number.

The service primitives listed hereafter initiate reservation transactions:

- *RS-Conference-Detail-Inquire-request*;
- *RS-Conference-Cancel-request*;
- *RS-Conference-Check-Availability-request*;
- *RS-Conference-List-Inquire-request*;
- *RS-Conference-Modify-request*;
- *RS-Conference-Reserve-request*;
- *RS-Site-Delete-request*;
- *RS-Site-Directory-Inquire-request*;
- *RS-Site-Modify-request*;
- *RS-Site-Record-request*;
- *RS-Non-Standard-Transaction-request* (see 7.2.3.4).

When the new transaction number and handle are allocated, the URST APE replies immediately by generating an *RS-Transaction-Allocation-indication* to the reservation application, indicating the new identifier through parameter *Transaction Handle* of this primitive, prior to sending the appropriate *rsXXXRequest* PDU to the peer URST APE.

7.2.3.2 Transactional sequences – Transactions procedures

Any transaction is started by an *rsXXXRequest* PDU sent by the URST APE at the originator side, on request of the reservation application. The peer URST APE replies to the request PDU by sending back the corresponding *rsXXXResponse* PDU. The *rsXXXResponse* PDU contains a parameter *acknowledgmentType* that defines three classes of transactions, *explicitly acknowledged*, *acknowledged* and *unacknowledged* transactions.

An unacknowledged transaction is normally terminated when, in turn, the URST APE at the originator side receives the corresponding *rsXXXResponse* response PDU. Receiving the response PDU, the URST APE will generate the corresponding *RS-XXX-confirm* service primitive to the reservation application and drop the transaction identifier. At the responding side of an unacknowledged transaction, the URST APE may drop the corresponding transaction identifier as soon as the *rsXXXResponse* PDU is sent toward the peer side.

Acknowledged transactions are normally ended in a similar way, but with the following difference. When receiving the *rsXXXResponse* PDU indicating an acknowledged transaction, the URST APE at the originator side shall generate the corresponding *RS-XXX-confirm* service primitive to the reservation application, and automatically send a *rsAcknowledgeTransactionIndication* PDU to the peer side before dropping the transaction identifier. At the peer side, the URST APE shall generate an *RS-Transaction-Acknowledge-indication* on reception of the *rsTransactionAcknowledgeIndication* and shall drop the transaction identifier at this instant.

Explicitly acknowledged transactions are normally ended in the following way. When receiving the *rsXXXResponse* PDU indicating an explicitly acknowledged transaction, the URST APE at the originator side shall indicate it to the reservation application by parameter *Explicit Acknowledgment Required* of the corresponding confirm service primitive. The URST APE shall then wait to receive

an *RS-Transaction-Acknowledge-request* primitive from the reservation application. When this occurs, the APE shall then send an *rsTransactionAcknowledgeIndication* PDU to the peer side, before dropping the transaction identifier. At the peer side, the URST APE shall generate a *RS-Transaction-Acknowledge-indication* on reception of the *rsTransactionAcknowledgeIndication* before dropping the transaction identifier.

The following standard transactions are always of the acknowledged type:

- *RS-Conference-Cancel*;
- *RS-Site-Delete*;
- *RS-Site-Modify*;
- *RS-Site-Record*.

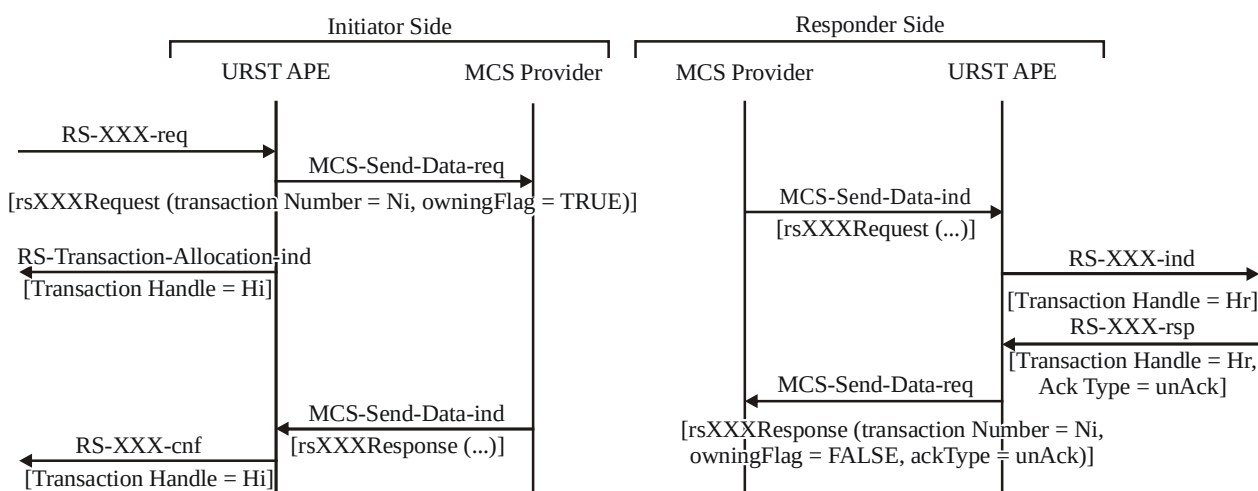
The following standard transactions are either of the acknowledged or the explicitly acknowledged type:

- *RS-Conference-Modify*;
- *RS-Conference-Reserve*.

Other standard transactions of T.135 are of the unacknowledged type, while proprietary transactions or sub-transactions may be of any type (see 7.2.3.4).

A reservation application at the originator side of a transaction may explicitly request its termination by issuing an *RS-Transaction-Cancel-request* service primitive. If so, the URST APE will send an *rsTransactionCancelIndication* PDU to the peer side, and drop the transaction identifier. If the URST APE receives the response PDU afterward (i.e., if it receives a response PDU which transaction identifier indicates that its own side was the initiator of the transaction and that matches no allocated identifier), it shall ignore the PDU. A URST APE receiving a *rsTransactionCancelIndication* PDU shall generate an *RS-Transaction-Cancel-indication* to the reservation application.

Figures 7-5 to 7-7 show the typical sequences for each type of standardized transaction.



T.135(07)_F.7.5

Figure 7-5 – Standardized unacknowledged transactions sequence

Reservation System Controlled Parameters, and are indicated by the mark "(RC)" following their names in the tables of clause 8.

In a response service primitive, the reservation system will indicate that it has "turned around" a group of such parameters, by setting parameter *Explicit Acknowledgment Required*. In such a case, the URST APE at the RSN will generate the response PDU with parameter *acknowledgmentType* indicating that the explicit acknowledgment is required. The explicit acknowledgment will be performed as explained in 7.2.3.2.

7.2.3.4 Proprietary information exchanges

The *RS-Non-standard-Transaction* and *RS-Non-Standard-Data* services of T.135 enable sending and receiving three types of PDUs, the *rsNonStandardRequest*, the *rsNonStandardResponse* and the *rsNonStandardIndication* PDU, referred to as non-standard PDUs in regard to their name, although they are specified in this Recommendation.

Non-standard PDUs contain the optional parameter *protocolKey* that may be used to identify a standard or non-standard protocol or data format used in their data field. The format of this parameter is similar to the one defined for an *Application Protocol Key* in [ITU-T T.124] (see 3.3 of [ITU-T T.124]). Parameter *protocolKey* is reflected by parameter *Protocol Key* of the *RS-Non-Standard-Transaction* and *RS-Non-Standard-Data* service primitives.

As described in 7.2.3.4.1 to 7.2.3.4.4, the *RS-Non-standard-Transaction* service is used for handling proprietary transactions or *sub-transactions* while the *RS-Non-Standard-Data* service is used for *transaction-related information delivery* or *asynchronous information delivery*.

7.2.3.4.1 Proprietary transactions

Proprietary transactions are performed through the *rsNonStandardRequest* and *rsNonStandardResponse* PDUs, respectively generated via service primitives *RS-Non-standard-Transaction-request* and *RS-Non-standard-Transaction-response*, and indicated via *RS-Non-standard-Transaction-indication* and *RS-Non-standard-Transaction-confirm*.

On reception of an *RS-Non-standard-Transaction-request* primitive, the URST APE shall allocate a transaction identifier for the proprietary transaction and indicate it to the reservation application by generating an *RS-Transaction-Allocation-indication* primitive to the reservation application, in the same manner as for standard transactions. Parameters *Acknowledgment Type* and *Explicit Acknowledgment Required* of *RS-Non-standard-Transaction-response* and *RS-Non-standard-Transaction-indication* enable proprietary transactions to be of the explicitly acknowledged, acknowledged or unacknowledged type.

Figure 7-8 shows the use of the *RS-Non-Standard-Transaction* service for proprietary transactions.

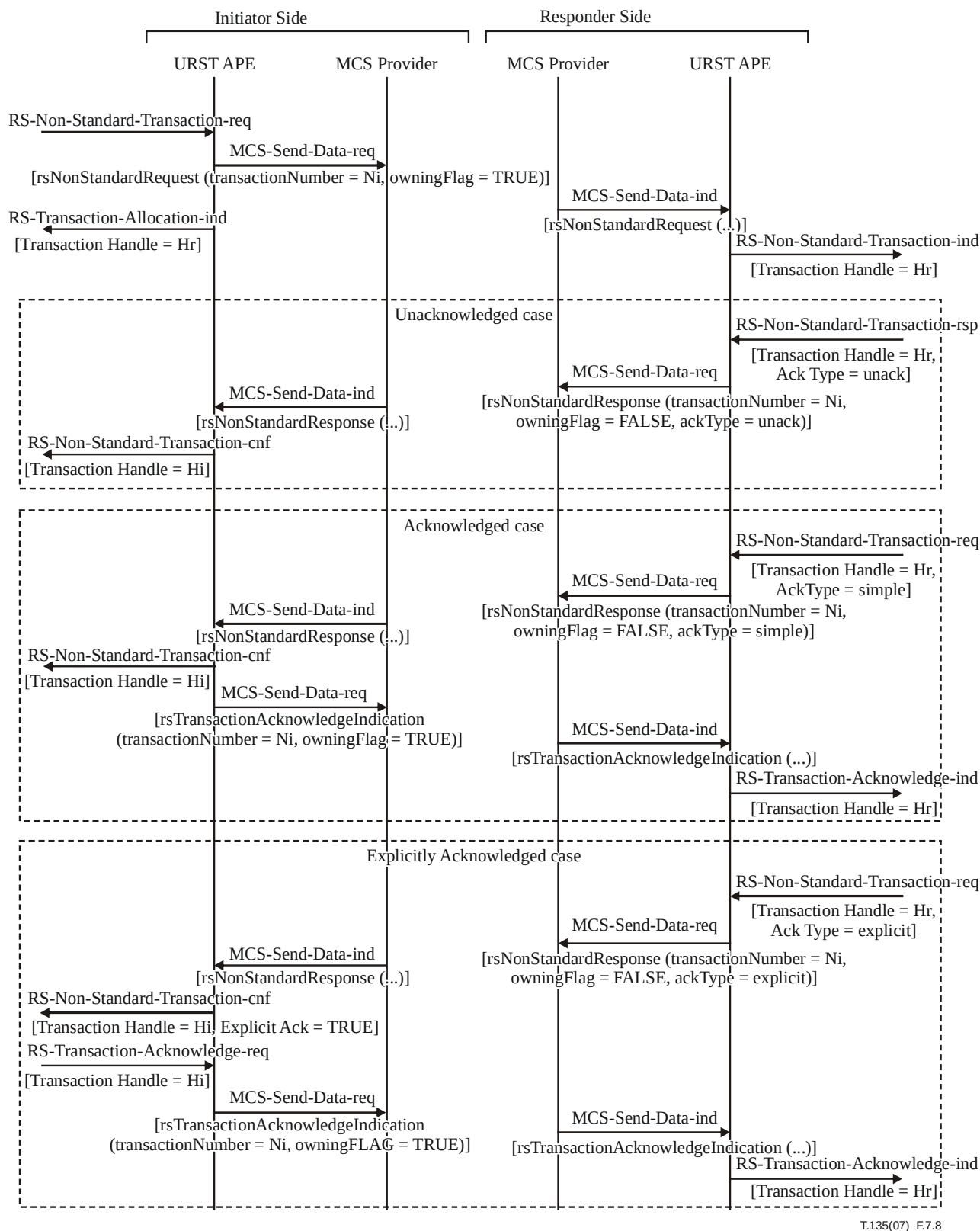


Figure 7-8 – Proprietary transactions typical sequences

7.2.3.4.2 Proprietary sub-transactions

Before answering a transaction request, the responding application may require other pieces of information from the peer side. To do so, it is allowed to issue a *proprietary sub-transaction* before replying to the original transaction. A proprietary sub-transaction is performed in the same manner as a proprietary transaction (see 7.2.3.4.1), but is also characterized by a parent transaction

identifier linking it to its parent transaction. Parameter *Parent Transaction ID* of the *RS-Non-Standard-Transaction* service primitive is provisioned for that purpose. Sub-transactions are terminated in the same way as transactions (see 7.2.3.2), and are automatically aborted if their parent transaction is terminated.

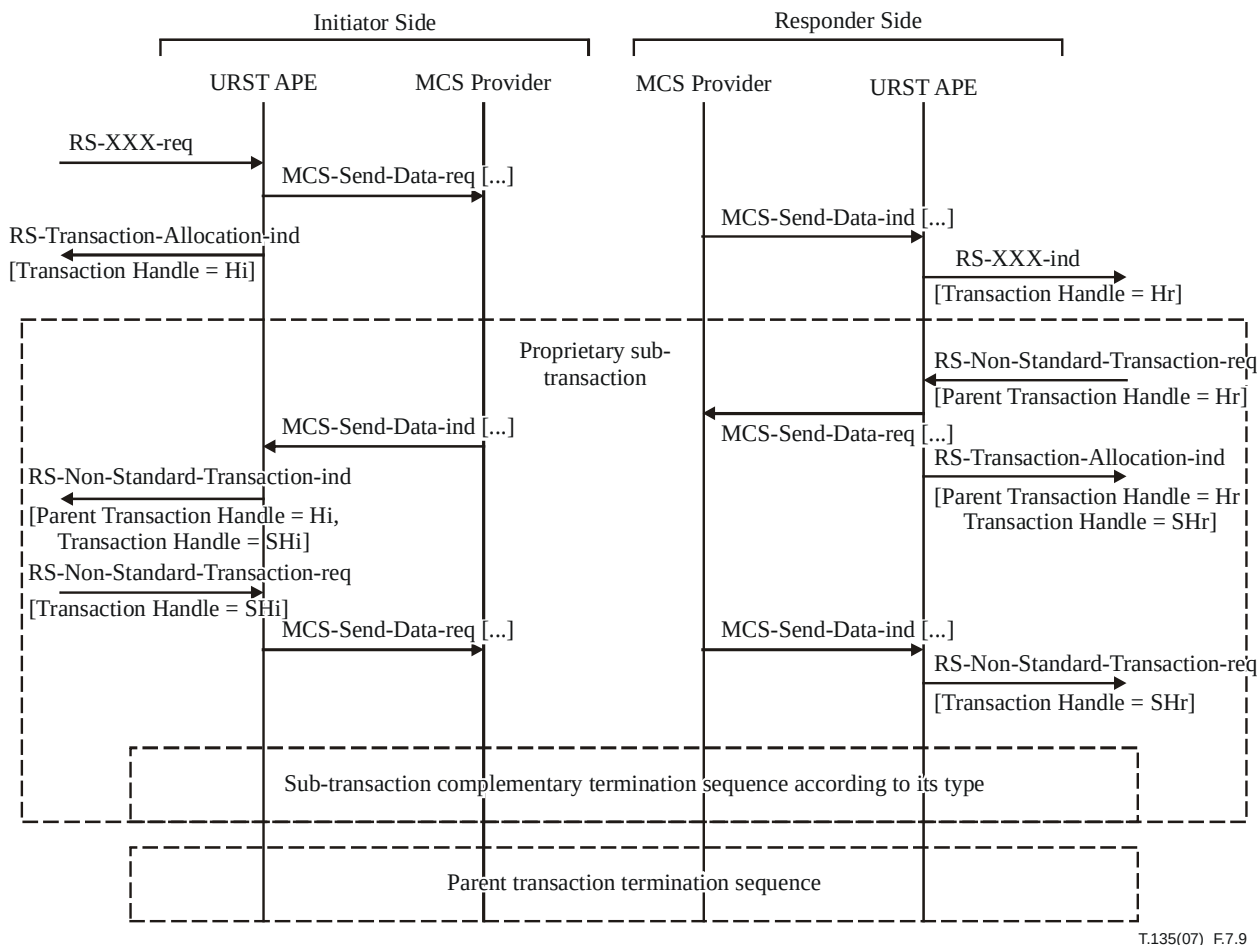


Figure 7-9 – Proprietary sub-transaction typical sequence

7.2.3.4.3 Transaction-related information delivery

While a transaction or sub-transaction is ongoing, both sides may also send asynchronous pieces of information related to the transaction or sub-transaction. Asynchronous information have no transaction identifier, but have a parent transaction identifier which shall be set to the identifier of the transaction or sub-transaction to which it is related. Asynchronous information are sent through the *rsNonStandardIndication* PDU which is generated via the *RS-Non-Standard-Data-request* service primitive.

7.2.3.4.4 Asynchronous information delivery

Alternatively to transaction-related information delivery, each side of a reservation connection may, at any time, send asynchronous pieces of proprietary information not related to any ongoing transaction or sub-transaction. This is always done through the *rsNonStandardIndication* PDU with no parent transaction identifier, which is generated by the *RS-Non-Standard-Data-request* service with parameter *Parent Transaction ID* not supplied. In the same manner as for transaction-related information delivery, parameter *protocolKey* may be used to identify a protocol or data format.

The mechanisms around non-standard PDUs are intended to facilitate proprietary dialogs between a reservation system and a user, when standard transactions of this Recommendation are not

sufficient. For instance, this enables a reservation system to project proprietary facilities to a user by starting a sub-transaction of an *RS-Conference-Reserve* transaction.

7.2.4 Other behaviours

7.2.4.1 Conducted mode

URST APEs are transparent to the conducted mode. That is, they shall work in the same manner whilst the conference is in the conducted or non-conducted mode.

7.2.4.2 Distant invocation

URST APEs shall not make use of the *GCC-Application-Invoke-request* service.

8 Service description

8.1 Service summary

Table 8-1 lists URST service primitives. When applicable, the last columns indicates if the corresponding transaction is of the Unacknowledged (U), Acknowledged (A) and/or Explicitly Acknowledged (EA) type.

Table 8-1 – URST service primitives

Service primitive	Subclause	Description	Transaction type		
			U	A	EA
RS-Check-Reservation-Systems	8.2.2	Retrieving the list of reservation systems present in the conference	Not applicable		
RS-Conference-Detail-Inquire	8.2.9	Importing the full description of a scheduled or ongoing conference	√		
RS-Conference-Cancel	8.2.7	Cancelling a scheduled conference.		√	
RS-Conference-Check-Availability	8.2.10	Obtaining a list of available dates with time intervals for a conference described in terms of bandwidth, number of participating sites and all other parameters related to resources allocation	√		
RS-Conference-List-Inquire	8.2.8	Retrieving a list of scheduled and/or ongoing conferences under a summarized form	√		
RS-Conference-Modify	8.2.6	Modifying a scheduled conference		√	√
RS-Conference-Reserve	8.2.5	Scheduling a conference		√	√
RS-Connect	8.2.3	Establishment of reservation connection	Not applicable		
RS-Disconnect	8.2.4	Releasing a reservation connection	Not applicable		
RS-Non-Standard-Data	8.2.16	Non transaction proprietary data exchange (Transaction-related Information Delivery or Asynchronous Information Delivery)	Not applicable		
RS-Non-Standard-Request	8.2.15	Proprietary transaction or sub-transactions	√	√	√
RS-Site-Delete	8.2.13	Deleting site record from a reservation system database		√	