International Telecommunication Union

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

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/.

CONTENTS

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 multiport 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.

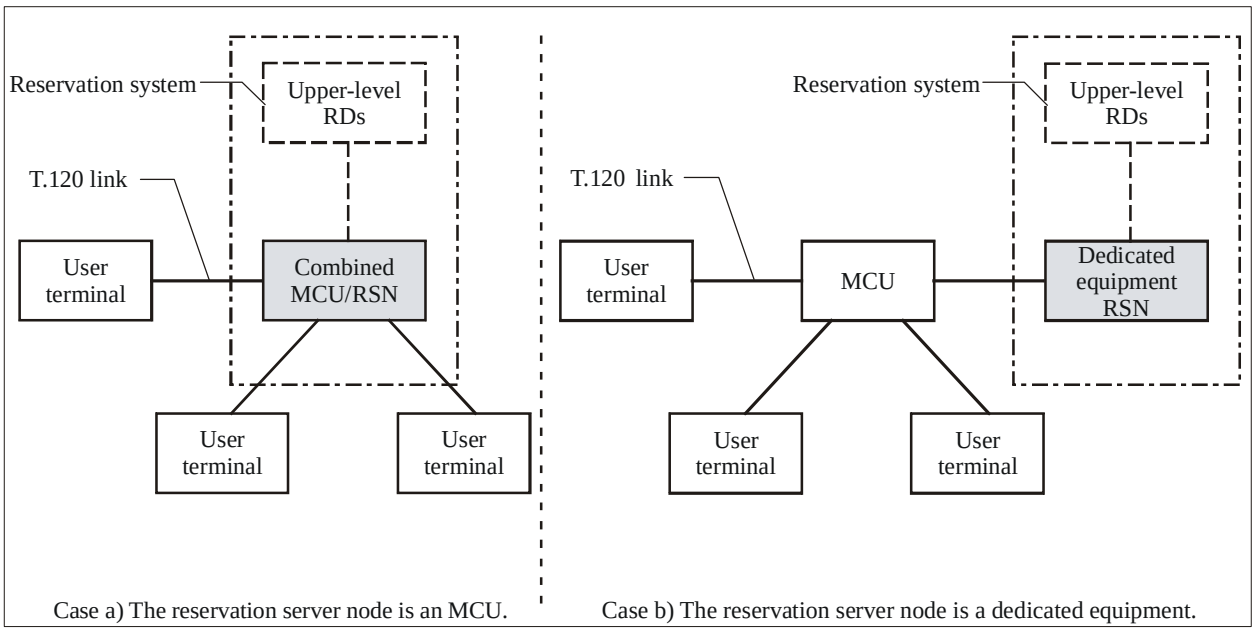

Figure 7-1 – The reservation server node in the system model

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.

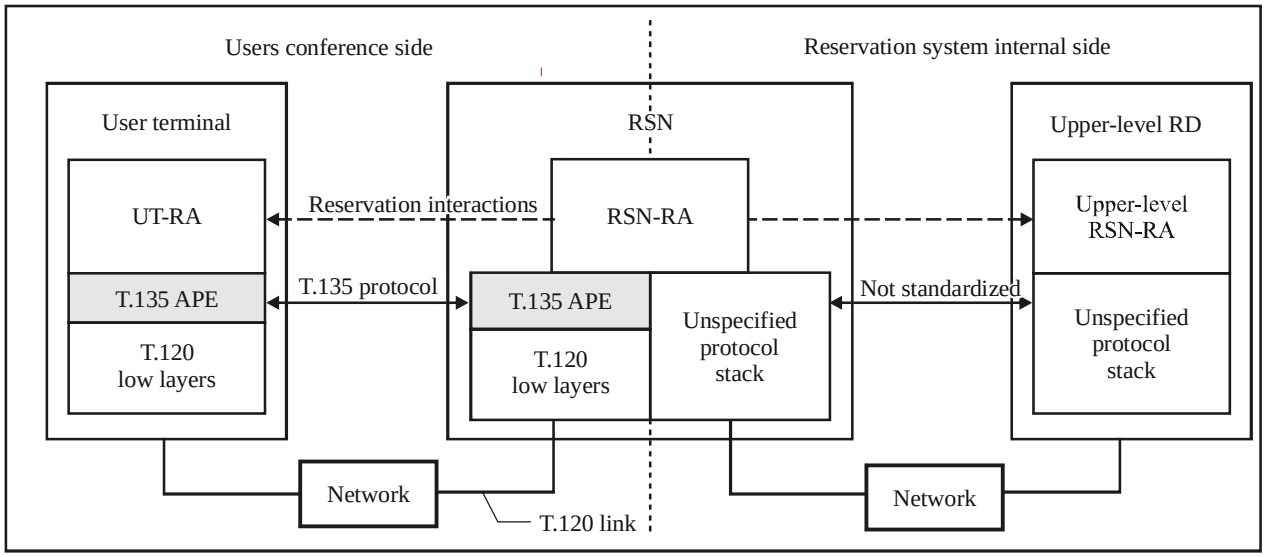

Figure 7-2 – System model and scope of T.135

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 *H221NonStandardIdentifier* 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.

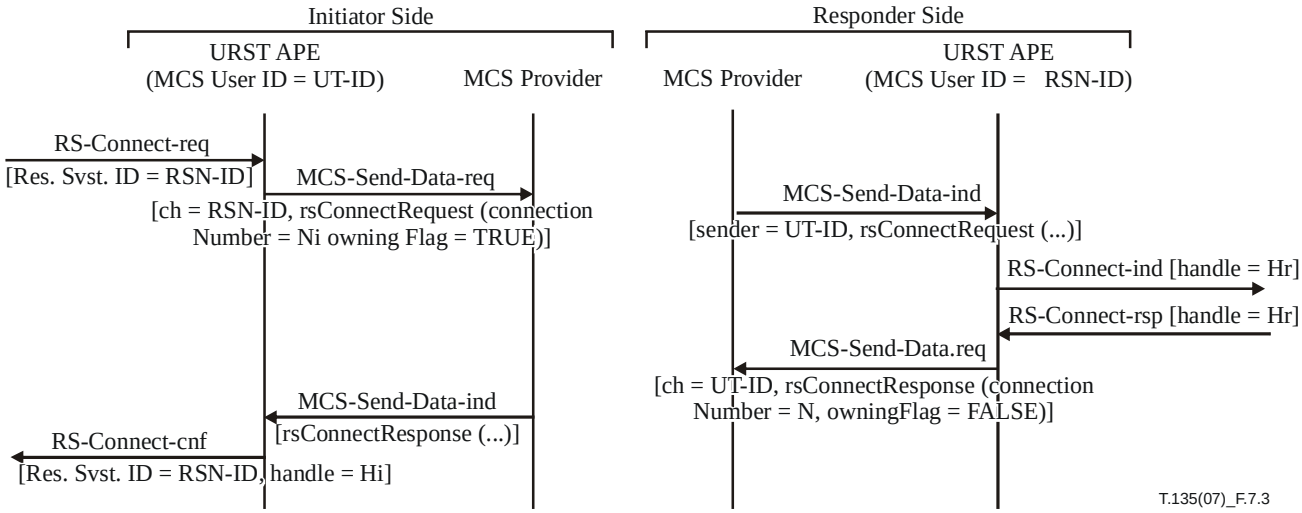

Figure 7-3 – Opening a reservation connection

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.

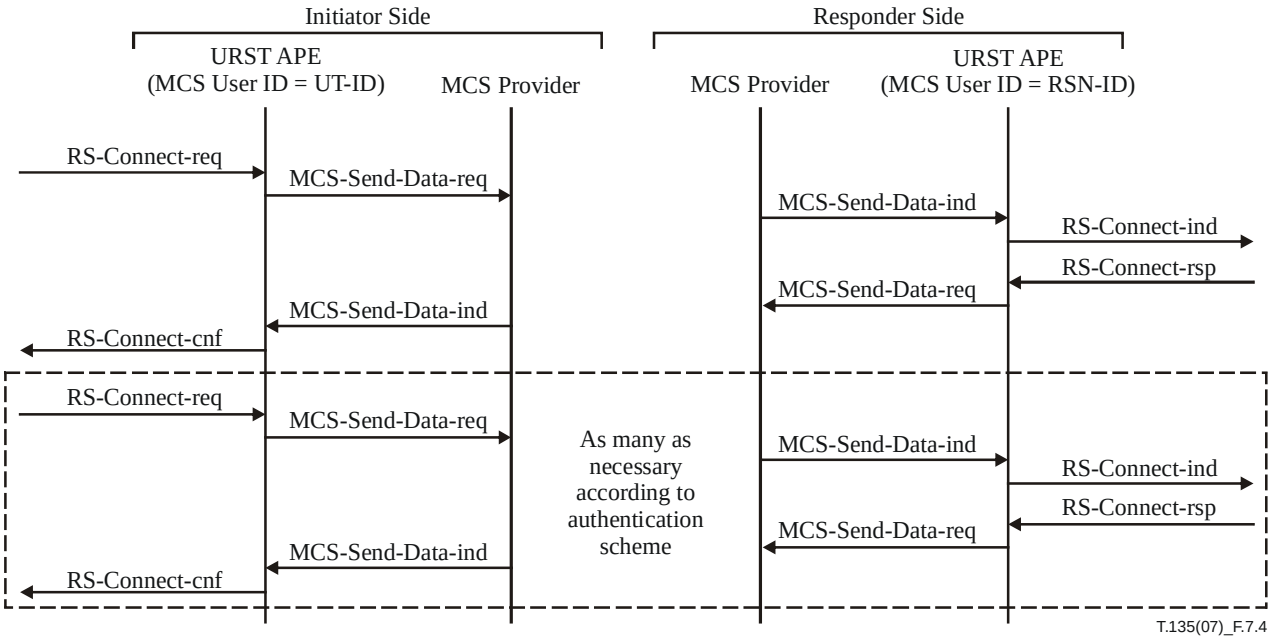

Figure 7-4 – Reservation connection initialization sequence

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.

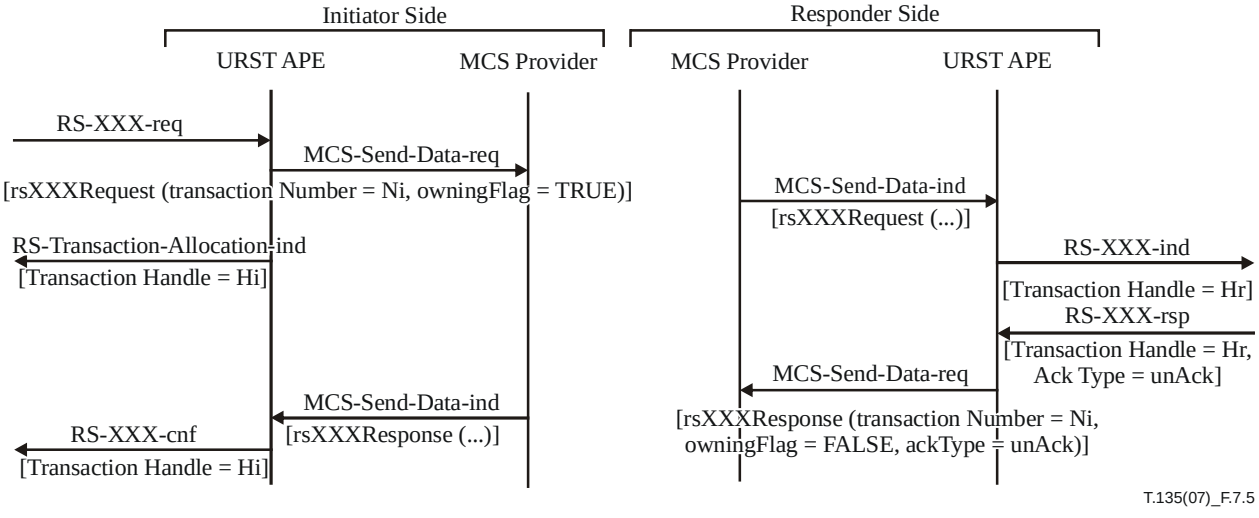

Figure 7-5 – Standardized unacknowledged transactions sequence

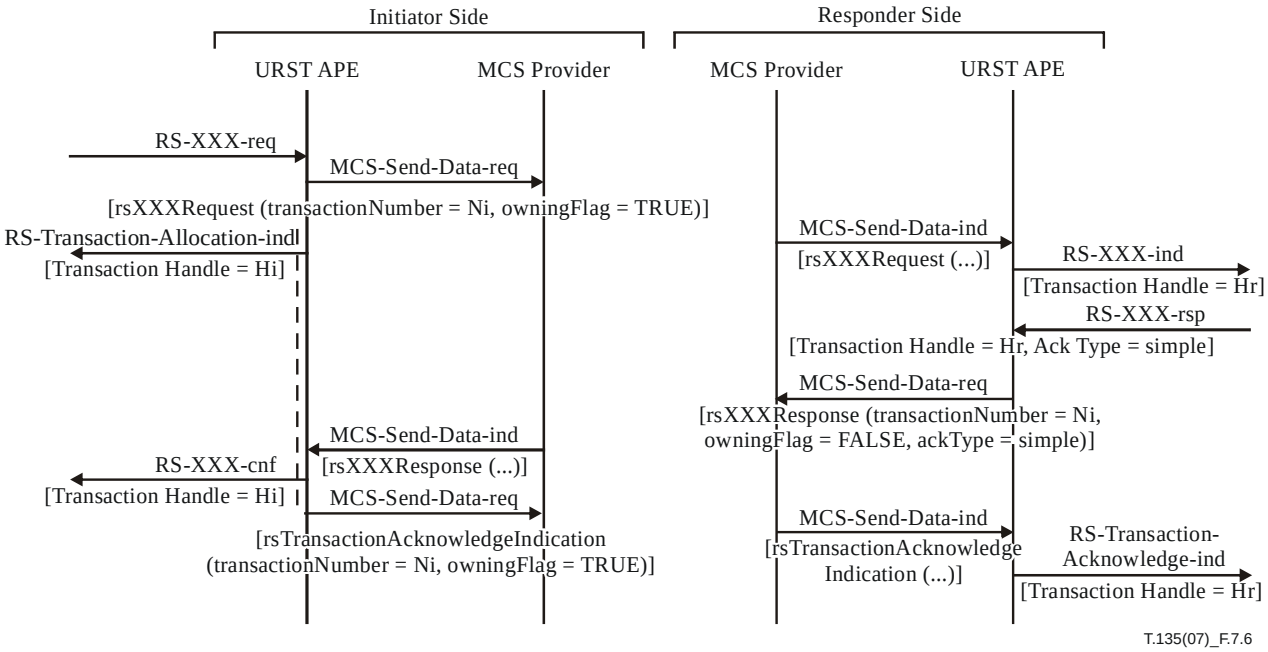

Figure 7-6 – Standardized acknowledged transactions sequence

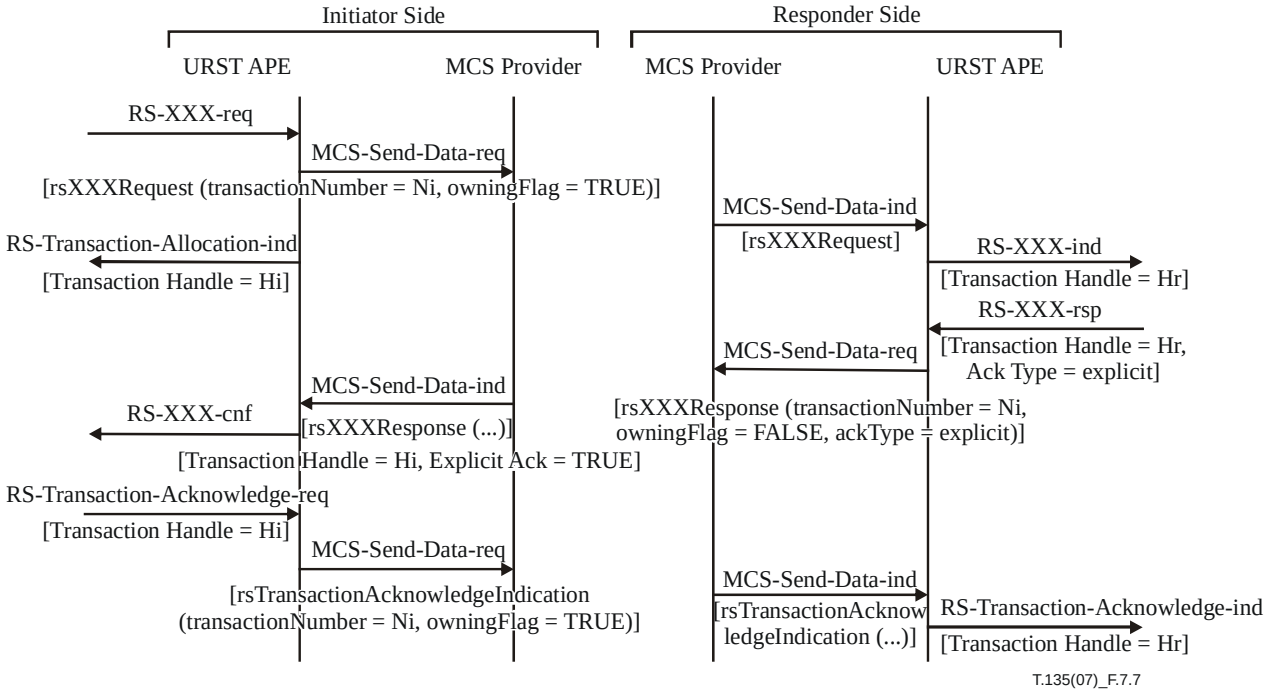

Figure 7-7 – Standardized explicitly acknowledged transactions sequence

7.2.3.3 Reservation System Controlled Parameters

Most of the parameters of standard acknowledged transactions listed in 7.2.3.2 are intended to be input by the user, whichever presentation of those parameters the UT-RA will make. However, most of those parameters represent a user preference rather than a strict requirement over the transaction. Parameter *Video Switching Method* of the *RS-Conference-Reserve* transaction is an example of those. In order to not constrain the reservation system to refuse a transaction systematically because such a user preference does not match its resources' characteristics, the reservation system is allowed to accept a transaction whilst changing the value of the corresponding parameter in the response primitive/PDU of the transaction. Such parameters are therefore

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.

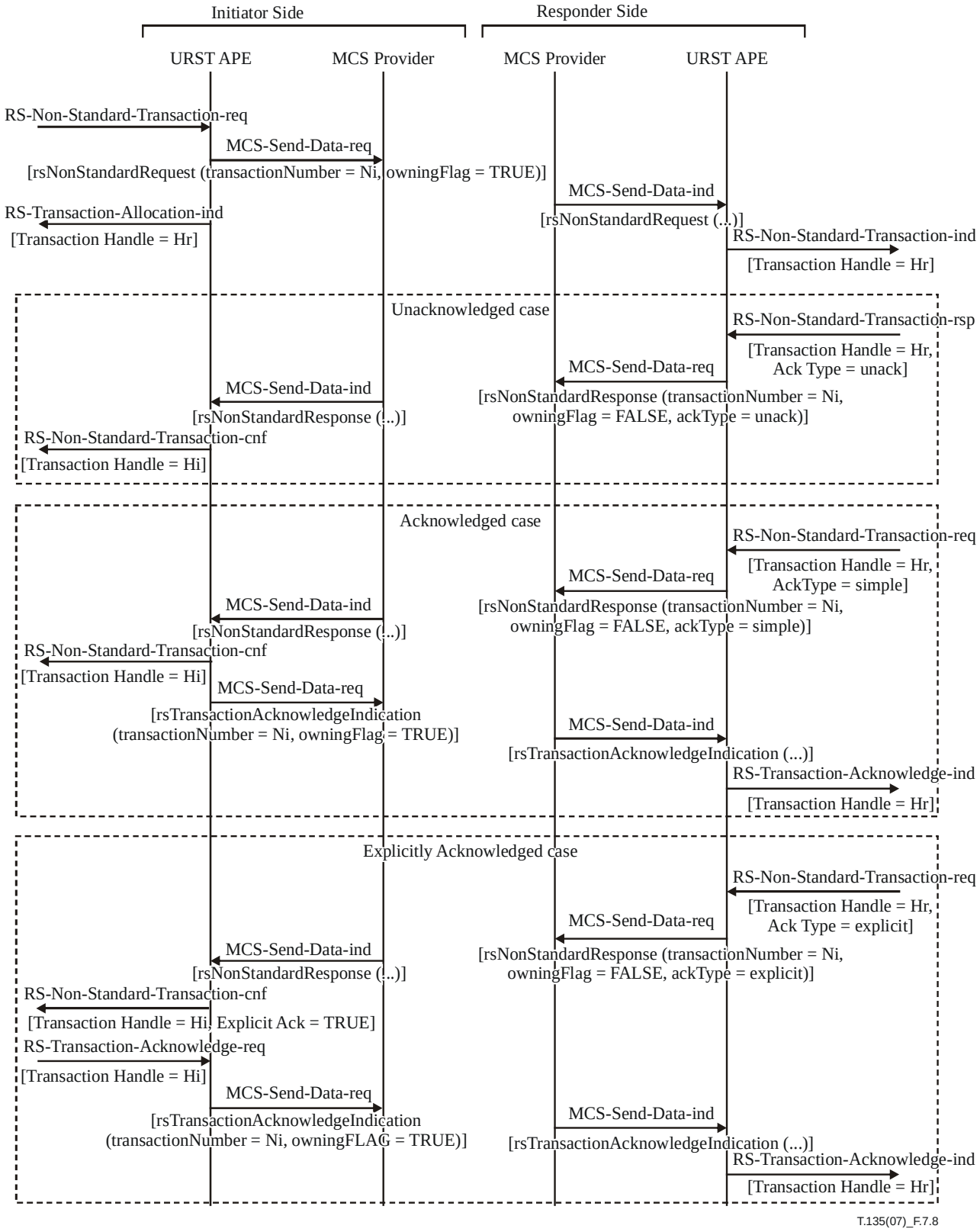

Figure 7-8 – Proprietary transactions typical sequences

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.

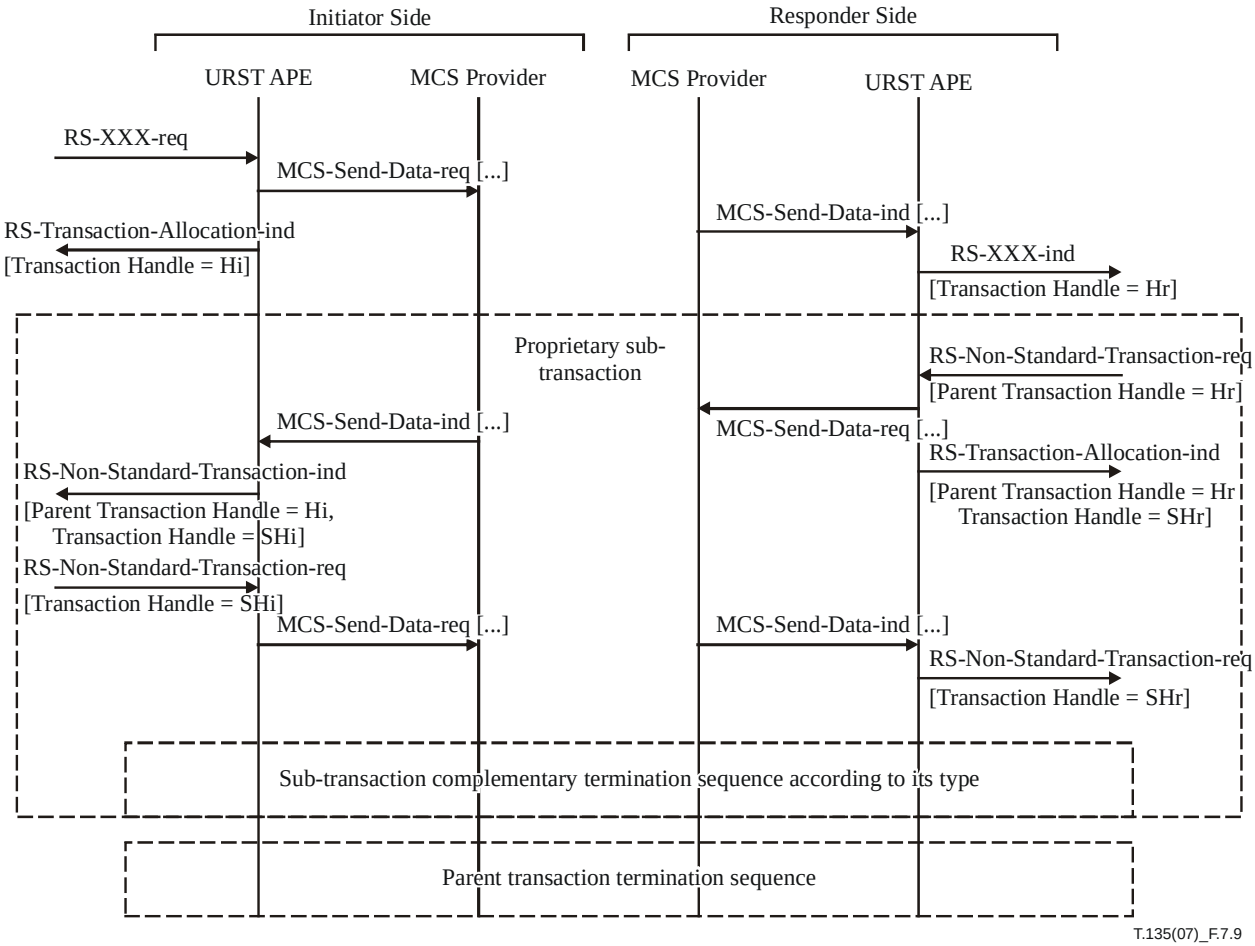

Figure 7-9 – Proprietary sub-transaction typical sequence

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		√	

Table 8-1 – URST service primitives

Service primitive	Subclause	Description	Transaction type		
			U	A	EA
RS-Site-Directory-Inquire	8.2.14	Retrieving a list of sites and associated information, based on a valued set of criteria such as alphabetical interval, geographical location ...	√		
RS-Site-Modify	8.2.12	Modifying a site record in a reservation system database		√	
RS-Site-Record	8.2.11	Creating a site record in a reservation system database		√	
RS-Transaction-Acknowledge	8.2.18	Acknowledging a reservation transaction	Not applicable[a)]		
RS-Transaction-Allocation	8.2.17	Allocating a transaction identifier	Not applicable		
RS-Transaction-Cancel	8.2.19	Cancelling a transaction	Not applicable		
RS-Transaction-Error	8.2.20	Error signalling service	Not applicable		
a) Used for terminating A and EA transactions.					

8.2 Service description

8.2.1 Applicable rules

8.2.1.1 General identifiers

Many URST protocol/service parameters needs identifiers for being referred to within URST transactions. For instance, a conference successfully reserved is referred to through a conference/reservation reference in further modifications of the reservation (see parameter *Conference/Reservation Reference* in 8.2.5). Unless otherwise specified, any URST transaction parameter serving as an identifier reflects a protocol-level parameter of the ASN.1 defined type *GenericID* (see clause 9) which gives the choice between an unsigned integer and an alphanumeric string.

8.2.1.2 Standard result codes

Response forms of most URST service primitives include a parameter *Result Code* that reports a local error or an error notified by the peer side.

The following list gives a set of result codes that may be used for response form of any URST service primitive that contains parameter *Result Code*: "*Success*", "*Bad Connection Handle*", "*Bad Transaction Handle*", "*Bad Parameters*", "*Bad Parameters Values*", "*Local Failure*", "*No Response From Peer*", "*Transaction Refused By Peer*", "*Default Failure Code*".

For each URST service primitive, the set of supplementary standard result codes are listed in the subclause concerned.

8.2.2 RS-Check-Reservation-Systems

This primitive is used by a UT-RA to retrieve the list of RSN-RAs enrolled in the URST registration session. To process an *RS-Check-Reservation-Systems-request*, a URST APE shall issue a *GCC-Application-Roster-Inquire-request* to the local GCC provider, specifying the URST registration session key. It shall then filter the information included in the corresponding *GCC-Application-Roster-Inquire-confirm* to construct the appropriate *RS-Check-Reservation-Systems-confirm* for the UT-RA.

If the local GCC provider does not support the *GCC-Application-Roster-Inquire* service, the URST APE is responsible for maintaining the necessary information through each *GCC-Roster-Report-indication* received from the local GCC provider.

The parameters of this primitive are listed in Table 8-1 *bis*.

Table 8-1 *bis* – RS-Check-Reservation-Systems primitive

Parameter	Req	Cnf
Reservation Systems List		O

Reservation Systems List consists of a list containing as many entries as there are URST APEs of the RSN type enrolled in the URST registration session. Absence of this parameter in the confirm primitive indicates an empty list. Each entry contains the parameters listed in Table 8-2.

Table 8-2 – Reservation Systems List parameter – Structure of an element

Parameter	Cnf
RSN ID	M
RA ID	M
Reservation System Name	M
Reservation System ID	C

RSN ID is the node ID of the RSN, according to the definition of a node ID given in [ITU-T T.124].

RA ID is the MCS User ID associated with the RSN-RA enrolled at that node, according to the definition of an MCS User ID given in [ITU-T T.124].

Reservation System Name is the string contained in the data associated with the *Reservation System Name* non-collapsing capability defined in 7.2.1.

Reservation System ID identifier is contained in the data associated with the *Reservation System ID* non-collapsing capability defined in 7.2.1. It is mandatory if the capability was supplied by the RSN-RA.

8.2.3 RS-Connect

This primitive is used to open a reservation connection and enables the user to log into the reservation system. Its parameters are listed in Table 8-3.

Table 8-3 – RS-Connect primitive

Parameter	Req	Ind	Rsp	Cnf
RA ID	C			C(=RQ)
Subscriber ID	O	C(=)	C(=)	C(=)
Authentication Data	O	O(=)	O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
Connection Handle	C(=CF)	M	M(=IN)	M
Higher Protocols Supported	O	O(=)	O	O(=)
User Data	O	O(=)	O	O(=)

RA ID enables the user to target a particular reservation system. This parameter must match one of the RA IDs obtained by the *RS-Check-Reservation-Systems* service. In the response and confirm primitives, this parameter enables the requesting UT-RA to correlate the reply with the original request. This parameter is mandatory in the first request and confirm forms of the primitive related to this connection. It is absent in all other *RS-Connect* primitives related to this connection.

Subscriber ID identifies the user to the reservation system. It is of the general identifier type as defined in 8.2.1. It is mandatory in the first indication and confirm forms of the primitive related to this connection if it was supplied in the first *RS-Connect-request* issued by the UT-RA, and otherwise absent in those primitives. It is optional in the first *RS-Connect-request* primitive, so that unregistered users may attempt to connect. When *Subscriber ID* is not provided, it is the responsibility of the reservation system to accept or refuse the connection. In the first response and confirm forms of the primitive, it enables the UT-RA to correlate the reply with the original request. It is absent in all other *RS-Connect* service primitives related to that connection.

Authentication Data may take several forms depending on the step number at which the service primitive is sent or received, on whether the authentication scheme is unilateral or mutual, and on the authentication algorithms chosen. The parameter is divided in two logical parts, each being optional in each form of the primitive. The first part is dedicated to data supplied to the peer side, data required from it. Data supplied to the peer side may consist either of a password directly supplied to the peer side or of a response to a challenge previously required by it, while the second part consists of a challenge request. Multiple pass sequences are allowed as indicated in 7.2.2.3. Refer to the protocol definition in clause 9 for the detail of this parameter.

Result Code can take the standard supplementary values "*Authentication In Process*" and "*Authentication Failed*". This parameter is always mandatory in the response and confirm forms of the primitive, but shall be set to "*Authentication In Process*" for all of them except for the last ones.

Result Message is an optional text string that may be used to complete the result code. It is optionally present in the last response and confirm forms of the primitive related to this connection, and irrelevant for previous ones.

Connection Handle identifies the connection as indicated in 7.2.2. It is absent in the first *RS-Connect-request* issued by the UT-RA, and is mandatory in all other *RS-Connect* primitives related to this connection.

NOTE – Connection handles theoretically allow a particular user to perform simultaneous multiple logins in a single reservation system. However, this feature support is not subject to this Recommendation and is the responsibility of each individual reservation system to allow or refuse.

Higher Protocols Supported is a list of protocol keys enabling each side to indicate the list of protocols or data formats that they support through the *RS-Data service*. Each protocol key is either an ASN.1 OBJECT IDENTIFIER belonging to a Recommendation, standard or non-standard protocol, or, alternatively, it is a non-standard identifier using the encoding conventions of [ITU-T H.221].

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.4 RS-Disconnect

Table 8-4 – RS-Disconnect primitive

Parameter	Req	Ind
Connection Handle	M	M
Reason Code	M	M(=)
Reason Message	O	O(=)
User Data	O	O(=)

This primitive is the last step of a reservation connection. A reservation terminal will not disjoin from a conference before closing all reservation connections opened through it.

Connection Handle identifies the connection. See 7.2.2.

Reason Code optionally indicates the reason of the disconnection. Standard values are "*Peer Initiated Disconnection*", "*Peer Disconnected From Conference*" or "*Default Disconnection Code*".

Reason Message is an optional text string that may be used to complete the reason code.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.5 RS-Conference-Reserve

This primitive defines a conference reservation transaction. Its parameters are listed in Table 8-5.

Table 8-5 – RS-Conference-Reserve primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Conference Name (RC)	M	M(=)	O	O(=)
Conference Description	O	O(=)	O(=)	O(=)
MCU Cascading Mode	O	O(=)	O(=)	O(=)
Audiovisual Chair Control	O	O(=)	O(=)	O(=)
Audiovisual Chairman Password	O	O(=)	O(=)	O(=)
T.120 External Convener	O	O(=)	O(=)	O(=)
T.120 Conducted Mode	O	O(=)	O(=)	O(=)
T.120 Conductor Privileges List	O	O(=)	O(=)	O(=)
T.120 Common Privileges List	O	O(=)	O(=)	O(=)
T.120 Conductor Password	O	O(=)	O(=)	O(=)
Common Password	O	O(=)	O(=)	O(=)
Default Transfer Rate (RC)	M	M(=)	O	O(=)
Video Switching Method (RC)	O	O(=)	O	O(=)
Initial Video Format (RC)	O	O(=)	O	O(=)
Initial Video Algorithm (RC)	O	O(=)	O	O(=)
Initial Audio Algorithm (RC)	O	O(=)	O	O(=)

Table 8-5 – RS-Conference-Reserve primitive

Parameter	Req	Ind	Rsp	Cnf
Waiting List Policy	O	O(=)		
Billing Mode (RC)	O	O(=)	O	O(=)
Organizer Billing Account	O	O(=)	O(=)	O(=)
Conference Owner ID	O	O(=)	O(=)	O(=)
Date	M	M(=)	O(=)	O(=)
Time	M	M(=)	O(=)	O(=)
Duration	M	M(=)	O(=)	O(=)
Booking Sites List (RC)	M	M(=)	O	O(=)
Sites Default Joining Method	M	M(=)	O(=)	O(=)
Conference Modifier Password	O	O(=)	O(=)	O(=)
Conference/Reservation Reference			C	C(=)
Explicit Acknowledgment Required			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

Absence of *reservation system controlled* parameters (marked with a "(RC)", see 7.2.3.3) in the response and confirm forms of the primitive shall be interpreted as an implicit acknowledgment by reservation system. Alternatively the reservation system may supply such parameters in the response form of the primitive even when it does not change their values, for explicitly acknowledging the requested ones.

Connection Handle identifies the reservation connection. See 7.2.2.

Transaction Handle identifies of the transaction. See 7.2.3.

Conference Name and *Conference Description* are text strings that name and summarize the conference. At runtime these will map both *Conference Name* and *Conference Description* parameters defined in [ITU-T T.124]. In that case, this parameter is mandatory in the response and confirm forms of the primitive.

MCU Cascading Mode indicates the type of cascading that will take place at conference runtime between MCUs of the requested reservation system reservation domain and external MCUs (i.e., MCUs included in the list of booking sites – see description of the *Booking Sites List* in 8.2.5.2 – or attached to them), and the role that shall be played by those, according to rules specified in [ITU-T H.243]. Standard values of this parameter are "*Simple Cascading*", "*Master External*", "*Master Internal*" or "*Unspecified*". "*Simple Cascading*" indicates that cascading, if any, shall be of the simple form, i.e., MCUs selected in the requested reservation domain shall behave as simple sites to external MCUs. "*Master External*" indicates that cascading shall base on a master/slave relationship, and that the master MCU is external. "*Master Internal*" indicates that cascading shall base on a master/slave relationship, and that the master MCU shall be one the reservation domain. "*Unspecified*" indicates that the cascading mode shall be decided at connection time. Absence of this parameter is equivalent to selecting "*Unspecified*".

Audiovisual Chair Control is a boolean flag that indicates that the MCUs that will be selected in the reservation domain should support the H.243 chair control. Absence of this parameter is equivalent to not requiring it.

Audiovisual Chairman Password is an optional password that specifies which H.243 password shall be used at connection time by a node that intends to grab the H.243 chairmanship. Absence of this parameter or passing a null string means that the H.243 chairmanship is not password protected.

NOTE 1 – As specified in [ITU-T H.243], H.243 chair control and H.243 cascading are separate functions. This means that, when the list of booking sites include external MCUs, the H.243 chairing site will not certainly chair over the entire H.320 conference. The set of equipment through which the H.243 chairing position of a particular terminal will apply depends on the type of cascading running between each H.320 sub-domain. In general, when a reservation is made piece by piece (i.e., when the list of booking sites includes external MCUs) it is the responsibility of the reservations maker(s) to reach consistency between the distinct parts. This Recommendation does not specify which particular combinations should be rejected by a reservation system. Acceptance or refusal of a reservation request is left to the discretion of each individual reservation system.

T.120 External Convener is a boolean flag which indicates if the T.120 conference will be created by a site included in the *Booking Sites List* parameter (or attached to one of these at conference runtime). Absence of this parameter is equivalent to indicate that there is no external convener.

T.120 Conducted Mode is a boolean flag which indicates if the conference is to be T.120 conducted. This parameter is irrelevant if *T.120 External Convener* is set to TRUE, and absence of this parameter while *T.120 External Convener* indicates that there is no external convener is equivalent to schedule the conference in the non conducted mode.

T.120 Conductor Privileges List optionally gives the privileges of the conductor according to [ITU-T T.124], when the conducted mode is scheduled. This parameter is relevant only when *T.120 Conducted Mode* is relevant and set to TRUE. Absence of this parameter when it is relevant means that the conference shall be created according to the set of rules described in [ITU-T T.124].

T.120 Common Privileges List optionally gives the privileges for non-conductor nodes if the conference is to be created in the conducted mode, or for any node if the conference is to be created in the non-conducted mode. This parameter is relevant only when *T.120 External Convener* indicates that there is no external convener. Absence of this parameter when it is relevant means that the conference shall be created according to the set of rules described in [ITU-T T.124].

T.120 Conductor Password is an optional password that specifies which particular T.120 password shall be used at connection time by a node that intends to grab the T.120 conductorship. This parameter is relevant only when *T.120 Conducted Mode* is relevant and indicates that the conference shall be created in the conducted mode. Passing a null string is equivalent to not providing this parameter, that is not password-protecting the T.120 conductorship of the conference.

NOTE 2 – Interactions between H.243 chairmanship and T.120 conductorship are specified in [ITU-T H.243].

Common Password is the password for common nodes when entering the conference, that is for non-audiovisual chairing and non-T.120 conducting nodes (note that this will also be the password for these nodes in the case of a non-password protected audiovisual chairmanship or/and a non-password protected T.120 conductorship). Passing a null string is equivalent to not providing this parameter, that is not password-protecting the conference for common nodes.

Default Transfer Rate is the requested nominal transmission rate for the conference. This parameter represents the global bandwidth of which each terminal should dispose for the combined needs of H.221 framing, bonding if required, audio, video and data. It may indicate multiple values of 64 kbit/s up to 1920 kbit/s. It also may be overwritten at site level by parameter *Site Transfer Rate* of parameter *Booking Sites List* (see 8.2.5.2).

NOTE 3 – This parameter gives the reservation system a default indication for the allocation of ports resources. However, the actual bandwidth that will be used at conference runtime can be different since terminals connected to a restricted digital network may be included in the *Booking Sites List* parameter. Specific information on one particular site such as $N \times 64/64 \times N$ compatibility (see SM-comp in [ITU-T H.242]), the type of its access network ... are given during the site registration phase (see 8.2.11).

Based on these information, the reservation system will determine the type of connection needed and the applicability of the conference.

Video Switching Method indicates the preferred video switching algorithm. This parameter has a structure which varies according to the selected method. In all cases it contains a first sub-parameter *Algorithm Selector* which identifies the video switching algorithm. Possible algorithms are "*Default Voice Activity Detection*", "*Periodic Rotation*", "*Fixed Broadcaster Site*", "*Last Speakers Mosaic*", "*Fixed List Mosaic*". Absence of *Video Switching Method* in the request/indication forms of the primitive is equivalent to selecting the default voice activity detection mode. Tables 8-6 to 8-10 of 8.2.5.1 describe the structure for each case.

Initial Video Format may optionally be used by the requester to indicate a preferred video format at the beginning of the conference. Standard values of this parameter are "*CIF*", "*QCIF*", "*SIF*", "*SQCIF*" or "*Runtime Choice*". Absence of this parameter is equivalent to selecting the "*Runtime Choice*" option, that is, leaving the choice of a video format to the discretion of the set of MCUs and/or other management devices that will be involved.

Initial Video Algorithm may optionally be used by the requester to indicate a preferred video coding algorithm at the beginning of the conference. Standard values for each element of the list are "*H.261*", "*H.262*", "*H.263*". or "*Runtime Choice*". Absence of this parameter is equivalent to selecting the "*Runtime Choice*" option, that is leaving the choice of a video format to the discretion of the set of MCUs and/or other management devices that will be involved.

Initial Audio Algorithm may optionally be used by the requester to indicate a preferred audio coding algorithm at the beginning of the conference. Standard values of this parameter are "*G.711 – A law*", "*G.711 – μ law*", "*G.722*", "*G.723*", "*G.728*", "*G.729*", "*MPEG Audio*" or "*Runtime Choice*". Absence of this parameter is equivalent to selecting the "*Runtime Choice*" option, that is, leaving the choice of an audio coding algorithm to the discretion of the set of MCUs and/or other management devices that will be involved.

Waiting List Policy is an indicator that enables the requester to define the way the reservation system shall treat the reservation if it cannot be entirely accepted because of a lack of MCU or network resources. This parameter can take the values "*All Waiting List*", "*Best Effort*" or "*No Waiting List*". "*All Waiting List*" is to request that the entire conference be placed in the waiting list, while "*Best Effort*" means that the conference shall be reserved for the parts that find resources while the remaining set of booking sites shall be placed in the waiting list. When "*Best Effort*" is selected, the order of the sites in parameter *Booking Sites List* determine their relative priority in regard to their placement in the waiting list. Absence of this parameter is equivalent to selecting the "*No Waiting List*", that is not requesting any waiting list service. In the response/confirm forms of the primitive, the special value "*Waiting List*" of parameter *Result Code* shall be returned if some of the booking sites or the entire conference were placed in the waiting list. Each site placed in the waiting list shall then be indicated through optional parameter *Waiting List Indicator* of the structure describing an element of *Booking Sites List* (see 8.2.5.2). Parameter *Result Code* set to "*Waiting List*" while parameter *Waiting List Indicator* is present for none or all of the booking sites indicates that the entire conference was placed in the waiting list.

NOTE 4 – Placement of a site in the waiting list concerns MCU and network resources only. Reservation of the corresponding terminal (if it is a terminal that can be reserved) is a complete distinct function.

NOTE 5 – When a reservation or a set of booking sites are placed in the waiting list, the way by which users are further notified of the final decision is out the scope of this Recommendation and left to the discretion of each reservation system (i.e., service provider). The support of any of the waiting list service options by a reservation system is not subject to this Recommendation.

Billing Mode optionally indicates the requested billing method. This parameter can take standard values "*Organizer*", "*Participants*", "*Mixed*" or "*Implicit Billing Mode*", beside non-standard values. "*Organizer*" means that the total cost of the conference is to be billed to the organizer (i.e., the person or entity identified by parameter *Subscriber ID* of the *RS-Connect* primitive), "*Participants*"

means that the total cost of the conference shall be shared over the participating sites, and "*Mixed*" indicates that each of the participating sites and the organizer shall be billed according to a scheme defined by parameter *Participant Billing Ratio* described in 8.2.5.2. "*Implicit Billing Mode*" is a value enabling the requester to reference a predefined billing mode which scheme has been previously agreed between the subscriber and the service provider by means outside the scope of this Recommendation. Absence of this parameter is equivalent to selecting the "*Implicit Billing Mode*" option.

Organizer Billing Account identifies a particular billing account that the service provider shall use for the billing of the conference, in the case where the connected subscriber has access to several accounts. It is of the general identifier type as defined in 8.2.1. Not providing this parameter is equivalent to select an implicit billing account which predetermination has been done by means outside the scope of this Recommendation.

Conference Owner ID is an alternative subscriber identifier enabling the requester to reserve the conference on behalf a third-party subscriber. By default (i.e., when this parameter is not supplied), the owner of the reservation is implicitly identified by parameter *Subscriber ID* supplied at connection time (i.e., in the *RS-Connect* primitive).

Date, *Time* and *Duration* are respectively the date, the start time and the duration of the conference. Time shall be expressed according to the UTC reference.

Booking Sites List is a structured list of participating sites. Each element has the structure described in 8.2.5.2.

Sites Default Joining Method indicates the default method that will be used by any site included in the list of booking sites for entering the conference (calling/called), unless re-specified at the site level in parameter *Booking Sites List* (see 8.2.5.2). Possible values are "*Called*" and "*Calling*".

Conference Modifier Password is an optional password that when defined enables a reservation system subscriber or non-registered user whose access privileges do not normally authorize to do so, to modify or cancel the issued reservation. The user will have to provide this password in conjunction with the conference/reservation reference in order to modify or cancel it (see *Conference/Reservation Reference* parameter below and 8.2.6 and 8.2.7).

Conference/Reservation Reference identifies the conference as stored in the reservation system. It is of the general identifier type as defined in 8.2.1. Any further attempt to modify or cancel the conference should be made through this reference. It is allocated by the reservation system and return of this parameter in the response and confirm forms of the primitives is mandatory unless the reservation is refused.

Explicit Acknowledgment Required is a boolean flag that indicates whether the initiator side of the transaction shall explicitly acknowledge the transaction via the *RS-Transaction-Acknowledge* service. See 7.2.3.

Result Code can take the standard supplementary values "*Waiting List*" and "*No Resources Available*".

Result Message is an optional text string that may be used to complete the result code.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.5.1 Video switching method parameter

Table 8-6 – Video Switching Method parameter – "*Default Voice Activity Detection*" case

Parameter	Req	Ind	Rsp	Cnf
Algorithm Selector (="*Default Voice Activity Detection*")	M	M(=)	O	O(=)

The default voice activity detection method broadcasts the speaking site to all others. The speaking site gets the image of the previously speaking site. When this mode is reserved, sites may change the mode in a real-time manner, as specified in [ITU-T H.243] and [ITU-T H.245].

Table 8-7 – Video Switching Method parameter – "*Periodic Rotation*" case

Parameter	Req	Ind	Rsp	Cnf
Algorithm Selector (="*Periodic Rotation*")	M	M(=)	O	O(=)
Rotation Period	M	M(=)	O	O(=)

The periodic rotation method broadcasts alternatively each site of the conference during a given period. *Rotation Period* expresses in seconds the amount of time a site is broadcast before switching.

Table 8-8 – Video Switching Method parameter – "*Fixed Broadcaster Site*" case

Parameter	Req	Ind	Rsp	Cnf
Algorithm Selector (="*Fixed Broadcaster Site*")	M	M(=)	O	O(=)
Broadcaster Site ID	M	M(=)		

The fixed broadcaster site method sets the site optionally identified by parameter *Broadcaster Site ID* as permanent broadcaster site. If present, *Broadcaster Site ID* must match one of the site IDs given in parameter *Booking Sites List*.

Table 8-9 – Video Switching Method parameter – "*Last Speakers Mosaic*" case

Parameter	Req	Ind	Rsp	Cnf
Algorithm Selector (="*Last Speakers Mosaic*")	M	M(=)	O	O(=)
Number of Sources	M	M(=)	O	O(=)

The last speakers Mosaic methods broadcasts a mixed image composed of the speaking site and the (*Number of Sources* – 1) last speaking sites. The speaking site receives a mixed image composed of the *Number of Sources* last speaking sites. *Number of Sources* should not be greater than the number of sites in the *Booking Sites List* parameter.

Table 8-10 – Video Switching Method parameter – "*Fixed List Mosaic*" case

Parameter	Req	Ind	Rsp	Cnf
Algorithm Selector (="*Fixed List Mosaic*")	M	M(=)	O	O(=)
Site IDs List	M	M(=)	O	O(=)

The fixed list Mosaic method broadcasts a mixed image composed of the sites identified by the sites listed in *Site IDs List*. *Site IDs List* is a list where each member must match one of the site IDs given in parameter *Booking Sites List*. The number of sources of the Mosaic is determined by the size of *Site IDs List*.

NOTE – Since no BAS command exists in [ITU-T H.242] to select one of the methods "*Periodic Rotation*", "*Fixed Broadcaster Site*", "*Last Speakers Mosaic*" and "*Fixed List Mosaic*" in a real-time manner, selection or de-selection of such a method cannot be accomplished by other means than logging into the reservation system while the conference is running. Draft Recommendation T.130 may include real-time commands for these modes in further studies. For a conference with no T.120 participant, or no T.120 participant capable of accessing the reservation system while conferencing, no conferencing terminal will be capable of selecting or deselecting these video switching algorithms using this Recommendation.

8.2.5.2 Booking Sites List parameter

Each element of this list has the structure given by Table 8-11.

Table 8-11 – Booking Sites List parameter – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Site ID	M	M(=)	M(=)	M(=)
Site Entry Delay (RC)	O	O(=)	O	O(=)
Site Duration (RC)	O	O(=)	O	O(=)
Site Transfer Rate (RC)	O	O(=)	O	O(=)
Site Joining Method (RC)	O	O(=)	O	O(=)
Allocated Network Addresses			C	C(=)
Waiting List Indicator			O	O(=)
Participant Billing Ratio (RC)	O	O(=)	O	O(=)
Conferees List	O	O(=)	O(=)	O(=)

Reservation system controlled parameters (marked with a "(RC)", see 7.2.3.3) exist in this parameter and follow the same rules as indicated previously.

Site ID is a reference for the site that was allocated by the reservation system when the client reservation application registered the site via the *RS-Site-Record* service or by any other means outside the scope of this Recommendation. It is of the general identifier type as defined in 8.2.1.

Site Entry Delay and *Site Duration* are optional parameters enabling the site to be scheduled only for a part of the conference. Absence of one of these parameters in the request/indication forms of the primitive is equivalent to use the default values passed through parameters *Time* and *Duration* of Table 8-5. *Site Entry Delay* represents the delay between the beginning of the conference (defined by parameter *Time* of Table 8-5) and the entry of the site in the conference (i.e., the value indicated by *Site Entry Delay* added to the value indicated by *Site Duration* should not exceed the value indicated by parameter *Duration* of Table 8-5).

Site Transfer Rate enables the requester to indicate that the multimedia conferencing device at this site is intended to link at this particular transfer rate instead of the one indicated by *Global Transfer Rate*. Absence of this parameter in the request/indication forms of the primitives is equivalent to use the global transfer rate.

Site Joining Method is used to indicate if the site must be called or will call its attachment MCU at entry time. If supplied the request/indication forms of the primitive, this parameter overwrites (i.e., takes precedence) the default method indicated in parameter *Sites Default Joining Method*. Possible values are "*Calling*" and "*Called*".

Allocated Network Address is mandatory if the response/confirm primitive schedules the site as a calling site. This parameter gives information on the type of connection (switched ISDN or CSDN, transfer rate ...), the network addresses, and the type of profile that the calling site will have use for joining the conference. This parameter is described in 8.2.5.2.1.

Waiting List Indicator is a flag that optionally indicates that the reservation of MCU resources for the site is placed in the waiting list (see parameter *Waiting List Policy* in 8.2.5).

Participant Billing Ratio is an optional integer numeric constrained in the interval [0..100] that indicates the portion of the conference cost of the participating site which shall not be billed to the organizer. Not supplying this parameter while parameter *Billing Mode* is present in the request and set to "*Mixed*" is equivalent to supply and set it to 100.

NOTE – Selecting a mixed billing mode while setting the participant billing ratio to 100 for any booking site shall not be considered as equivalent to selecting the "*Participants*" billing mode. For instance, the reservation system (i.e., the service provider) may consider that a mixed billing mode with all participant billing ratios set to 100 (explicitly or implicitly) means that only the reservation cost will be billed to the organizer, each participant being billed the cost of his/her communication, while also sharing the reservation cost over the participants in the case of a "*Participants*" billing mode. In general, the exact interpretation of each mode is left to the discretion of reservation systems.

Conferees List is an optional parameter used to provide information over the conferees at that booking site (or attached to it). This parameter typically applies to videoconference rooms or external MCUs. Information provided in it may be used as pure informational items (such as participant names in a T.120 conference profile, see [ITU-T T.124]), or, for certain parts such as facsimile numbers or electronic mail addresses, to send notifications of the reservation to the participants. Each element of this list has the structure shown by Table 8-16.

8.2.5.2.1 Allocated Network Address parameter

The allocated network address qualifies the connection that the calling site shall establish for joining the conference. It has the structure given by Table 8-12.

Table 8-12 – Allocated Network Address parameter

Parameter	Rsp	Cnf
Connection Descriptor	M	M(=)
Network Interface To Use	M	M(=)
Profile To Operate	M	M(=)

Connection Descriptor has a structure that depends on the type of network interface selected by the reservation system.

For switched digital connections over ISDN or CSDN networks, this parameter has the structure of Table 8-13.

Table 8-13 – Connection Descriptor for ISDN/CSDN networks

Parameter	Description
Combined Circuits	This is a list of digital circuits, each characterized by a transfer mode and an extended E.164 network address. The transfer mode is either of a digital channel at 56 kbit/s for CSDN networks, a 64 kbit/s for ISDN or CSDN networks, or a 2×64 kbit/s, a 384 kbit/s, a 1536 kbit/s, a 1920 kbit/s or a multirate based 64 kbit/s digital channel for ISDN networks. For the latest case, an integer value within the range [1..30] indicates the effective transfer rate of the digital channel. Except the 56 kbit/s channel, these modes correspond to the codes given for octet "*Information transfer rate*" of Information Element "*Bearer Capability*" specified in [ITU-T Q.931]. For ISDN networks, the E.164 extended address is optionally accompanied by a list indicating the codepoints that are usable for IE HLC at call establishment. Both the extended E.164 network address and the list of HLC codepoints are described in Table 8-14.
Channel Aggregations	This is a list of channel aggregation algorithms that may be used to combine the channels. If the *Combined Circuits* lists only one circuit, this parameter is optional and its absence indicates that no channel aggregation shall be operated. Otherwise, this parameter is mandatory and shall contain at least one element. Standard values of one element of this list are "*H.221*", "*H.244*" and "*ISO/IEC 13871*".

Table 8-14 – Extended E.164 Network Address

Parameter	Description
International Number	This is a string of digits, up to 16 digits in length, which represents the full international number of the MCU port to be called by the booking site.
Sub-Address (optional)	This is an optional parameter, valid only in the case of ISDN connections, which represents the ISDN sub-address of the MCU port to be called by the booking site. This is a string of digits, up to 40 digits in length.
Extra Dialing String (optional)	This is an optional parameter which indicates that additional information is needed to reach the data processing unit once the physical connection has been established with the MCU. In the case of a speech or voice-band data connection, for example, this may represent DTMF tones to be transmitted over the voice channel once it has been established. Alternatively, the extra dialing may represent a virtual private network number. This is a string up to 255 characters which may be either the digits 1 through 9, the "#" character, the "*" character, or the "," (comma) character. The comma character is meant to represent a one second delay the booking site is to insert prior to the characters which follow.

Table 8-14 – Extended E.164 Network Address

Parameter	Description
High Layer Compatibility Information (optional)	This is an optional parameter, valid only in the case of ISDN networks, which indicates the mode of operation this portion of the connection is to use. This information is required for connections made through some ISDN networks. The modes of operation are one or more of "*telephony at 3 kHz bandwidth*", "*telephony at 7 kHz bandwidth*", "*videotelephony*", "*videoconferencing*", "*audiographics*", "*audiovisual*", or "*multimedia*". If more than one of these is selected, this indicates that the booking site may use one of the indicated modes at its discretion. This codes match the ones indicated in [ITU-T Q.931] for Information Element *High Layer Compatibility*.

For other types of connection, the connection descriptor parameter consists only in a *general network address*. This is a choice between an extended E.164 network address as defined by Table 8-14, a transport address, or a non-standard address. A transport address has the structure defined by Table 8-15.

Table 8-15 – Transport Address

Parameter	Description
NSAP Address	This is an octet string of up to 20 octets in length which is the preferred binary encoding [per A.8.3.1/X.213 (*Blue Book*)] of the Network Service Access Point address of the data processing unit to be reached.
Transport Selector	This is an optional parameter which may be used to select the Transport Service Access Point at the data processing unit to be reached.

Network Interface To Use identifies the network interface that the calling site shall use to dial in the conference. This is the identifier that is provided when the site is registered (see 8.2.11.2).

Profile To Operate is a basic configuration of a complex suite of protocols that the calling site shall operate over the connection described by Connection Descriptor. Standard values for this parameter are "*speech*", "*telephony-3kHz*", "*telephony-7kHz*", "*voice-band*", "*frame-relay*", "*t123-pstn-basic*", "*t123-psdn-basic*", "*t123-b-isdn-basic*", "*h310*", "*h320*", "*h321*", "*h322*", "*h323*", "*h324*", "*h324m*", "*v61*", "*v70*", "*dsmcc-download-profile*", or a non-standard profile. For the multimedia profiles (h.3xx, asvd, dsvd) a boolean flag is associated that indicates if the T.120 suites of protocols is to be operated for the data media.

8.2.5.2.2 Structure of a Conferee Description

Each conferee description of parameter *Conferees List* has the structure given in Table 8-16.

Table 8-16 – Conferees List parameter – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Name	O	O(=)	O(=)	O(=)
Name Complement	O	O(=)	O(=)	O(=)
Postal Addresses	O	O(=)	O(=)	O(=)
International Telephone Numbers	O	O(=)	O(=)	O(=)
International Facsimile Numbers	O	O(=)	O(=)	O(=)
E-Mail Addresses	O	O(=)	O(=)	O(=)
X.400 Mnemonic Addresses	O	O(=)	O(=)	O(=)
Optional Strings	O	O(=)	O(=)	O(=)

Name is an optional text string that represents the name of the conferee.

Name Complement is an optional text string that may include complementary nominative information such as the company of the conferee, the conferee's position in this company ...

Postal Addresses is an optional list of text strings, each representing an alternative postal address.

International Telephone Numbers is an optional list of strings of digits, each up to 16 digits in length and representing an alternative international telephone number of the conferee.

International Facsimile Numbers is an optional list of strings of digits, each up to 16 digits in length and representing an alternative international facsimile number of the conferee.

E-Mail Addresses is an optional list of text strings, each representing an alternative e-mail address of the conferee.

X.400 Mnemonic Addresses is an optional list of tables, each table representing an alternative X.400 address of the conferee. Each line of one table represents one field of the X.400 address defined under the mnemonic form, according to the rules specified in X.400-Series Recommendations (see clause 9).

Optional Strings is a list of text strings that may be used to provide information not specified in this Recommendation.

8.2.6 RS-Conference-Modify

This primitive defines a reservation modification transaction. Its parameters are listed in Table 8-17.

Table 8-17 – RS-Conference-Modify primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Conference/Reservation Reference	M	M(=)	O(=)	O(=)
Conference Modifier Password	O	O(=)		
Default Policy	O	O(=)		
Conference Name (RC)	O	O(=)	C(=)	C(=)
Conference Description	O	O(=)	O(=)	O(=)
MCU Cascading Mode	O	O(=)	O(=)	O(=)
Audiovisual Chair Control	O	O(=)	O(=)	O(=)

Table 8-17 – RS-Conference-Modify primitive

Parameter	Req	Ind	Rsp	Cnf
Audiovisual Chairman Password	O	O(=)	O(=)	O(=)
T.120 External Convener	O	O(=)	O(=)	O(=)
T.120 Conducted Mode	O	O(=)	O(=)	O(=)
T.120 Conductor Privileges List	O	O(=)	O(=)	O(=)
T.120 Common Privileges List	O	O(=)	O(=)	O(=)
T.120 Conductor Password	O	O(=)	O(=)	O(=)
Common Password	O	O(=)	O(=)	O(=)
Global Transfer Rate (RC)	O	O(=)	O(=)	O(=)
Video Switching Method (RC)	O	O(=)	O	O(=)
Initial Video Format (RC)	O	O(=)	O	O(=)
Initial Video Algorithm (RC)	O	O(=)	O	O(=)
Initial Audio Algorithm (RC)	O	O(=)	O	O(=)
Billing Mode (RC)	O	O(=)	O	O(=)
Organizer Billing Account	O	O(=)	O(=)	O(=)
Conference Owner ID	O	O(=)	O(=)	O(=)
Date	O	O(=)	O(=)	O(=)
Time	O	O(=)	O(=)	O(=)
Duration	O	O(=)	O(=)	O(=)
Booking Sites List (RC)	O	O(=)	O	O(=)
Sites Default Joining Method	O	O(=)	O(=)	O(=)
New Conference Modifier Password	O	O(=)	O(=)	O(=)
Explicit Acknowledgment Required			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

It should be noted that this primitive can be issued over an ongoing conference (i.e., a reservation corresponding to a conference that is running). In general, parameters of this primitive comply with the following rules:

- A parameter is provided to replace the value passed when the conference was reserved or at the last time it was modified.
- Not providing a parameter is equivalent to not modifying it.

  NOTE – For each type of parameter (strings, numbers ...) implementations of a T.135 APE shall provision special values or extra parameters not defined in this Recommendation in order to indicate the un-setting of a parameter (i.e., to cancel a previous setting of the parameter).
- Parameters inter-relationships (conditional presence or irrelevance depending on other parameters presence or value) are the same as those described in 8.2.5.
- Reservation system controlled parameters in the conference reservation transaction remain reservation system controlled parameters for the modification transaction.

Parameters listed in Table 8-17 and not described hereafter follow these rules.

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Conference/Reservation Reference identifies the reservation within the reservation system. It is the reference obtained when the initial reservation was made through a previous conference reservation transaction (i.e., given by parameter *Conference/Reservation Reference* of primitive *RS-Conference-Reserve-confirm*), or through other means outside the scope of this Recommendation.

Conference Modifier Password is the password defined in 8.2.5. It must be provided if it was defined when the conference was initially reserved and the connected user's privileges do not allow him to modify this reservation.

Default Policy defines the reservation system behavior if the new characteristics of the conference lead to a refusal of the reservation/conference modification. Possible values of this parameter are "*Keep Previous*" to request the original reservation be maintained, "*Cancel Previous*" to request it be dropped, "*All Waiting List*" to request it be dropped and the new conference placed in a waiting list, or "*Best Effort*", to request it be dropped and the new conference reserved according to the "best effort" policy (see parameter *Waiting List Policy* in 8.2.5). Absence of this parameter is equivalent to selecting "*Keep Previous*".

Booking Sites List is similar to parameter *Booking Sites List* described in 8.2.5 and 8.2.5.2 with the differences presented in 8.2.6.1.

New Conference Modifier Password optionally enables the requester to change the conference modifier password.

Explicit Acknowledgment Required is a boolean flag that indicates that the initiator side of the transaction shall explicitly acknowledge the transaction via the *RS-Transaction-Acknowledge* service. See 7.2.3.

Result Code can take the standard supplementary values "*Waiting List*", "*No Rights*", "*No Such Conference*" and "*No Resources Available*" or "*Default Failure Code*".

Result Message is an optional text string that may be used to complete the result code.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.6.1 Booking Sites List parameter

Table 8-18 – Booking Sites List parameter – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Operation Type	M	M(=)		
Site ID	M	M(=)	M	M(=)
Site Entry Delay (RC)	O	O(=)	O	O(=)
Site Duration (RC)	O	O(=)	O	O(=)
Site Transfer Rate (RC)	O	O(=)	O	O(=)
Joining Method (RC)	O	O(=)	O	O(=)
Network Interface To Use (RC)	O	O(=)	O	O(=)
Allocated Network Address			O	O(=)
Waiting List Indicator			O	O(=)

Table 8-18 – Booking Sites List parameter – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Participant Billing Ratio (RC)	O	O(=)	O	O(=)
Conferees List	O	O(=)	O	O(=)

In the request/indication forms of the primitive, the booking sites list contains the sites that are affected by the modification transaction. Parameter *Operation Type* identifies the operation to perform on each booking site. Possible values for this parameter are "*Delete*", "*Add*" or "*Modify*". The value of *Operation Type* determines the constraints on other parameters as follows. If *Operation Type* indicates a deletion, only parameter *Site ID* is relevant and mandatory. If *Operation Type* indicates a modification or an addition, other parameters are constrained according to Table 8-18 and follow the rules given in 8.2.6.

In the response/confirm forms of the primitive, the booking sites list shall contain the set of booking sites resulting from the modification transaction; that is the union of the set of sites that were booked before the modification transaction and the set of sites that were present in the *RS-Conference-Modify-request*/*indication* primitives and which associated *Operation Type* parameter was set to either of "*Modify*" or "*Add*". Parameters are constrained according to Table 8-18 and follow the rules given in 8.2.6.

8.2.7 RS-Conference-Cancel

Table 8-19 – RS-Conference-Cancel primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Conference/Reservation Reference	M	M(=)	O(=)	O(=)
Conference Modifier Password	O	O(=)		
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

This primitive is used to cancel a previous reservation.

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Conference/Reservation Reference identifies the reservation within the reservation system. See 8.2.5.

Conference Modifier Password is the password defined in 8.2.5.

Result Code can take the standard supplementary values "*No Such Conference*" and *"Transaction Refused*".

Result Message is an optional text string that may be used to complete the result code.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.8 RS-Conference-List-Inquire

Table 8-20 – RS-Conference-List-Inquire primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Conference Owner ID Filter	O	O(=)		
Conference Owner Name Filter	O	O(=)		
Conference Name Filter	O	O(=)		
Conference Status Filter	O	O(=)		
Date Filter	O	O(=)		
Conference Summaries List			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)

This primitive allows the subscriber to obtain a list of scheduled or ongoing conferences.

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Conference Owner ID Filter optionally indicates a list of subscriber identifiers as a selection criteria for the retrieval. When provided, only the conferences reserved by (or on behalf of) the subscribers identified by this parameter shall be returned.

Conference Owner Name Filter optionally indicates a set of alphabetical intervals as a selection criteria for the retrieval. When provided, only conferences reserved by (or on behalf of) subscribers whose names are comprised within the given intervals shall be returned. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Conference Name Filter optionally indicates a set of alphabetical intervals as a selection criteria for the retrieval. When provided, only conferences the names of which are comprised within the given intervals shall be returned. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Conference Status Filter optionally indicates a set of conference status as a selection criteria for the retrieval. When provided, only the conferences the status of which match the given ones shall be returned. Possible values for this parameter are "*Reserved*", "*Partially Reserved*", "*In Waiting-List*" and "*Ongoing*".

Date Filter optionally specify a set of days-interval for the search. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Conference Summaries List is a list of conferences presenting the result of the inquire transaction. Each conference is described in a summarized form the structure of which is given in Table 8-21 of 8.2.8.1. Absence of this parameter means that there is no visible reserved or ongoing conference for the connected user.

Result Code can take the standard supplementary value "*None Matching Criteria*".

Result Message is an optional text string that may be used to complete the result code.

8.2.8.1 Conference Summaries List parameter

Each element of *Conference Summaries List* has the structure defined by Table 8-21. Whether the requester obtains the list of the sole conferences he/she booked or a list including conferences booked by other subscribers is a reservation system matter.

Table 8-21 – Conference Summaries List parameter – Structure of an element

Parameter	Rsp	Cnf
Conference/Reservation Reference	M	M(=)
Conference Name	O	O(=)
Conference Description	O	O(=)
Conference Owner ID	O	O(=)
Conference Owner Name	O	O(=)
Date	M	M(=)
Time	M	M(=)
Duration	M	M(=)
Status	M	M(=)
Modify Permission	M	M(=)

Conference Owner ID optionally gives the subscriber identifier of the subscriber who reserved the conference (or on behalf of whom the conference was reserved).

Conference Owner Name optionally gives the name of the subscriber who reserved for the conference (or on behalf of whom the conference was reserved).

Status indicates the status of the conference. Possible values for this parameter are "*Reserved*", "*Partially Reserved*", "*In Waiting-List*" and "*Ongoing*.

Modify Permission is a boolean flag that indicates if the connected user has the right to modify or cancel the conference.

See previous subclauses for other parameters' description.

8.2.9 RS-Conference-Detail-Inquire

This primitive enables the requester to get the detailed form of a particular scheduled or ongoing conference. The requester provides the *Conference/Reservation Reference* and the system returns all the parameters describing a reserved conference. Table 8-22 lists the parameters describing a booked conference.

Table 8-22 – RS-Conference-Detail-Inquire

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Conference/Reservation Reference	M	M(=)	M(=)	M(=)
Conference Name			M	M(=)
Conference Description			O	O(=)
Conference Owner ID			O	O(=)

Table 8-22 – RS-Conference-Detail-Inquire

Parameter	Req	Ind	Rsp	Cnf
Conference Owner Name			O	O(=)
MCU Cascading Mode			O	O(=)
Audiovisual Chair Control			O	O(=)
Audiovisual Chairman Password			O	O(=)
T.120 External Convener			O	O(=)
T.120 Conducted Mode			O	O(=)
T.120 Conductor Privileges List			O	O(=)
T.120 Common Privileges List			O	O(=)
T.120 Conductor Password			O	O(=)
Common Password			O	O(=)
Global Transfer Rate			O	O(=)
Video Switching Method			O	O(=)
Initial Video Format			O	O(=)
Initial Video Algorithm			O	O(=)
Initial Audio Algorithm			O	O(=)
Billing Mode			O	O(=)
Organizer Billing Account			O	O(=)
Date			M	M(=)
Time			M	M(=)
Duration			M	M(=)
Booking Sites List			M	M(=)
Sites Default Joining Method			O	O(=)
Conference Modifier Password			O	O(=)
Modify Permission			M	M(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

Each element of parameter *Booking Sites List* contains the same sub-parameters as listed by Table 8-18 excluding *Operation Type*, all constrained as defined by columns "Rsp" and "Cnf" of this table. See previous subclauses for their description.

Modify Permission is a boolean flag that indicates if the connected user has the right to modify or cancel the conference.

Result Code can take the standard supplementary value "*No Such Conference*".

See previous subclauses for the description of the other parameters of the primitive.

8.2.10 RS-Conference-Check-Availability

Table 8-23 – RS-Conference-Check-Availability primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Date Filter	M	M(=)	O(=)	O(=)
Applying Sites List	M	M(=)		
Minimum Transfer Rate Accepted	M	M(=)		
Result List			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)

This primitive enables the requester to obtain a list of free time slots for a given conference defined in terms of all parameters related to resources allocation.

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Date Filter optionally specifies a set of days-interval for the search. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Applying Sites List indicates a list of sites that shall be considered as participating in the conference. Each site of this list has the structure described in 8.2.10.1.

Minimum Transfer Rate Accepted indicates a minimum transfer rate that shall globally be applied to the conference. This parameter is defined in the same manner as parameter *Global Transfer Rate* described in 8.2.5.

Result List gives a list of time slots for which the conference may be reserved. Each element of this list consists into a list of dates or dates-intervals. To each of the dates or dates-intervals is associated a list of time-intervals representing a time slot within which the conference may be reserved. To each of the time-intervals is associated a maximum transfer rate that could be allowed as the global transfer rate for the conference, and optionally, for each site listed in parameter *Applying Sites List*, the list of network interfaces that the site would, or alternatively would not, be allowed to use. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Result Code can take the standard supplementary value "*No Resources Available*".

Result Message may be used to provide a literal complement to *Result Code*.

8.2.10.1 Applying Sites List parameter

Each element of *Applying Sites List* has the structure defined by Table 8-24.

Table 8-24 – Applying Sites List parameter – Structure of an element

Parameter	Req	Ind
Site ID	M	M(=)
Site Entry Delay	O	O(=)
Site Duration	O	O(=)
Site Transfer Rate	O	O(=)
Possible Network Interfaces To Use	O	O(=)

Possible Network Interfaces To Use gives a list of network interfaces that the site would be able to operate for the conference. Absence of this parameter shall be interpreted as the list of all interfaces recorded with the site.

See previous subclauses for other parameters' description.

8.2.11 RS-Site-Record

This primitive is used to create temporary or permanent site records in the reservation system database. It shall be used to register a site (temporary or permanently) prior to issue of any *RS-Conference-Reserve* primitive for reserving a conference involving this site.

Table 8-25 – RS-Site-Record primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Permanent Indicator (RC)	O	O(=)	O	O(=)
Site Type	M	M(=)	O(=)	O(=)
Site Class	C	C(=)	O(=)	O(=)
Site Owner ID	O	O(=)	O(=)	O(=)
Site Name	O	O(=)	O(=)	O(=)
Site Geographic Information	O	O(=)	O(=)	O(=)
Network Interfaces	M	M(=)	O(=)	O(=)
Equipment Description	O(=)	O(=)	O(=)	O(=)
Contact Person	O	O(=)	O(=)	O(=)
Deletion Date			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
Site ID			C	C(=)
User Data	O	O(=)	O	O(=)

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Permanent Indicator is an optional flag that enables the client reservation application to require the permanent recording of the site within the reservation system database (default registration type is temporary). Acceptance or refusal of this service as well as the exact interpretation of "permanent" is left to the discretion of the reservation system. For instance, a reservation system may accept the

permanent registration of a site, but automatically deletes the corresponding record after a long period of non-utilization. In the response/confirm forms of the primitive, this parameter enables the reservation system to indicate the registration type that was adopted. The reservation system then gets the possibility to turn a permanent registration request into a temporary registration, returning a result code indicating a partial success and optionally indicating a deletion date for the site record (see *Deletion Date* and *Result Code* parameters below).

NOTE – The mechanism or registering sites implies the following rules: when a site is temporary registered, the concerned reservation system shall guarantee to the site record a life duration at least equal to the duration of the parent reservation connection if the site is not referenced in an accepted conference reservation issued through that connection; when a site is temporarily registered and referenced in an accepted conference reservation, the reservation system shall maintain the site record at least until the conference is terminated or cancelled.

Site Type identifies the type of device. Standard values are "*MCU*", "*Combined Terminal/MCU*", "*Terminal*", "*Telephone*" or "*Special Device*".

Site Class indicates the media types that the site operates. This parameter is mandatory when parameter *Site Type* is set to one of values "*MCU*", "*Combined Terminal/MCU*", "*Terminal*" or "*Special Device*", and is irrelevant otherwise. Standard values are "*Multimedia*", "*Audiovisual*", "*Audiographic*", "*Visiographic*", "*Data Only*".

Site Owner ID is an alternative subscriber identifier enabling the requester to record the site on behalf of a third-party subscriber. By default (i.e., when this parameter is not supplied), the owner of the site record is implicitly identified by parameter *Subscriber ID* supplied at connection time (i.e., in the *RS-Connect* primitive). It is of the general identifier type as defined in 8.2.1.

Site Name is a text string optionally used to name the site.

Site Geographic Information is a table of sub-parameters optionally providing information on the location of the site. The structure of this parameter is described in 8.2.11.1.

Network Interfaces gives the list of available network interfaces of network interface arrangements that the site is equipped with. The structure of an element is described in 8.2.11.2.

Equipment Description is a table of information about the conferencing equipment at this site. This parameter is described by Table 8-28 in 8.2.11.3.

Contact Person is an optional table providing information over a user associated with the site. Its structure is similar to the one described by Table 8-16.

Deletion Date is an optional parameter that the reservation system may return to indicate the date at which the site record will be deleted.

Result Code can take the standard supplementary values "*Temporary Mode Forced*" and "*User's Space Full*".

Result Message is an optional text string that may be used to provide literal complement to the result code.

Site ID is an identifier that will enable to reference the site in further transactions (conference reservations for instance). Its presence is mandatory if the recording succeeds, and its allocation is under the responsibility of the reservation system. It is of the general identifier type as defined in 8.2.1. This parameter is irrelevant if the recording fails.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.11.1 Site Geographic Information parameter

Table 8-26 – Site Geographic Information parameter

Parameter	Req	Ind	Rsp	Cnf
Country	O	O(=)	O(=)	O(=)
Region	O	O(=)	O(=)	O(=)
City	O	O(=)	O(=)	O(=)
Postal Address	O	O(=)	O(=)	O(=)

Country, *Region* and *City* are an alphabetical strings respectively naming the country, the region or state ..., and the city of the site.

Postal Address is a text string that may be used to provide the literal postal address of the site.

NOTE – Extension to use of X.500 directory services is for further study.

8.2.11.2 Network Interfaces parameter

This parameter has the structure of Table 8-27.

Table 8-27 – Network Interfaces parameter – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Interface Reference	M	M(=)	M(=)	M(=)
Interface Description	M	M(=)	M(=)	M(=)
Supported Profiles	M	M(=)	M(=)	M(=)

Interface Reference is supplied by the connected user and will identify the site in further transactions. It is of the general identifier type as defined in 8.2.1.

Interface Description has a structure that depends on the type of the network associated. Possible network types are ISDN, CSDN, GSTN, PSDN, ATM, Mobile and LAN networks.

For ISDN and CSDN networks, the interface description consists of a network interfaces arrangement description, that is a list of one or more network interfaces that can be combined together (for instance, a set of three ISDN BRI interfaces which B-channels may be aggregated). Each element of the set is list that gives the transfer rates that are supported by the compound interface and the network to which the interface is connected. The possible transfer rates are the ones enumerated for parameter *Connection Descriptor* described in 8.2.5.2.1. Associated with each transfer rate, parameter *Network Address* gives an extended E.164 network addresses that shall be used to dial the compound interface (see 8.2.5.2.1 for the description of an extended E.164 network address). Associated with the list of component interfaces, parameter *Channel Aggregations* specifies the list of channel aggregation algorithms that the site may operate to combine individual channels. Standard values of one element of this list are "*H.221*", "*H.244*" and "*ISO/IEC 13871*".

For GSTN networks, the interface description consists of an extended E.164 network addresses that shall be used to dial the interface, in which optional parameters *Sub-Address* and *High Layers Compatibility* Information are not applicable.

For PSDN networks, the interface description consists in a general network address (see 8.2.5.2.1 for the description of a general network address).

For ATM networks, the interface description consists in a general network address (see 8.2.5.2.1), and optionally an integer value indicating the maximum transfer rate allowed in cells per seconds.

For mobile networks, the interface description consists of an extended E.164 network address that may be used to dial the interface.

For LAN networks, the interface description consists in a general network address (see 8.2.5.2.1), and optionally an integer value indicating the maximum bandwidth that will be available to the interface at conference runtime in kbit/s.

Supported Profiles lists the basic configurations of more complex profiles that the site is capable of operating on the interface. Standard values are the ones listed for parameter *Profile To Operate* described in 8.2.5.2.1. For multimedia profiles (H.3xx, asvd, dsvd), the associated boolean flag indicates if the site is capable of operating the T.120 suite of protocols for the data portion of the multiplex.

8.2.11.3 Equipment Description parameter

Table 8-28 – Equipment Description parameter

Parameter	Req	Ind	Rsp	Cnf
Equipment Reference	O	O(=)	O(=)	O(=)
IMUX Reference	O	O(=)	O(=)	O(=)
Audio Codec Reference	O	O(=)	O(=)	O(=)
Audio Algorithms Supported	C	C(=)	O(=)	O(=)
Video Codec Reference	O	O(=)	O(=)	O(=)
Video Algorithms Supported	C	C(=)	O(=)	O(=)
Image Formats Supported	C	C(=)	O(=)	O(=)
Data Processor Reference	O	O(=)	O(=)	O(=)

Equipment Reference optionally gives general information on the terminal, MCU ... at the site. This parameter maps a protocol parameter of the ASN.1 defined type *VendorModelVersion* which is either of an H.221 non-standard identifier (see the non-collapsing capability *Reservation System ID* in Table 8-1 *bis* of 8.2.2) or a set of three text strings, one for the vendor, one for the model and one for the version of the device.

IMUX Vendor, *IMUX Model* and *IMUX Version* are alphanumeric strings that identify the IMUX processor if present. All are optional but it should be noted that *IMUX Vendor* and *IMUX Model* are useless one without the other.

Audio Codec Reference optionally gives information on the audio codec at the site. The structure is the same as for parameter *Equipment Reference*.

Audio Algorithms Supported consists of a list of one or more algorithms identifiers that indicates the audio capacity of the site. This parameter is irrelevant if the site is a telephone or does not operate any audio, and is mandatory otherwise. Standard values are "*G.711 – A law*", "*G.711 – μ law*", "*G.722*", "*G.723*", "*G.728*", "*G.729*" and "*MPEG Audio*".

Video Codec Reference optionally gives information on the video codec at the site. The structure is the same as for parameter *Equipment Reference*.

Video Algorithms Supported consists of a list of one or more algorithms identifiers that indicates the video capacity of the site. This parameter is irrelevant if the site is a telephone or does not operate any video, and is mandatory otherwise. Standard values for each element of the list are "*H.261*", "*H.262*" or "*H.263*".

Image Formats Supported consists in a list of one or more image format identifiers that indicates the video capacity of the site. This parameter is irrelevant if the site is a telephone or does not operate any video, and is mandatory otherwise. Standard values for each element of the list are "*CIF*", "*QCIF*", "*SIF*", "*SQCIF*".

Data Processor Reference optionally gives information on the audio codec at the site. The structure is the same as for parameter *Equipment Reference*.

8.2.12 RS-Site-Modify

This primitive is used to change a site record in the reservation system database. Its structure is given by Table 8-29.

Table 8-29 – RS-Site-Modify primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Site ID	M	M(=)	O(=)	O(=)
Permanent Indicator (RC)	O	O(=)	O	O(=)
Site Type	O	O(=)	O(=)	O(=)
Site Class	O	O(=)	O(=)	O(=)
Site Owner ID	O	O(=)	O(=)	O(=)
Site Name	O	O(=)	O(=)	O(=)
Site Geographic Information	O	O(=)	O(=)	O(=)
Network Interfaces	O	O(=)	O(=)	O(=)
Equipment Description	O	O(=)	O(=)	O(=)
Contact Person	O	O(=)	O(=)	O(=)
Deletion Date			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

In general, parameters of this primitive comply with the following rules:

- A parameter is provided in this primitive to replace the value passed when the site was recorded or at the last time it was modified.
- Not providing a parameter is equivalent to not modifying it.

NOTE – For each type of parameter (strings, numbers ...) implementations of a T.135 APE may provision special values or extra parameters not defined in this Recommendation in order to indicate the un-setting of a parameter (i.e., to cancel a previous setting of the parameter).

Parameters listed in Table 8-29 and not described hereafter follow these rules.

Site ID identifies the site record to be modified. It is mandatory in the request and indication primitives and optional in the response and confirm ones. Its value is the value returned by the reservation system when the site was recorded through *RS-Record-Site* or other means outside the scope of this Recommendation.

Network Interfaces is similar to parameter *Network Interfaces* of 8.2.11.2 with the differences presented in 8.2.12.1.

Result Code can take the standard supplementary values "*Temporary Mode Forced*", "*No Such Site*" and "*User's Space Full*" or "*Default Failure Code*".

All other parameters and sub-parameters describing the site become optional.

8.2.12.1 Network Interfaces parameter

In the *RS-Site-Modify* primitive, this parameter has the structure of Table 8-30.

Table 8-30 – Network Interfaces parameters – Structure of an element

Parameter	Req	Ind	Rsp	Cnf
Operation Type	M	M(=)		
Interface Reference	M	M(=)	M	M(=)
Interface Description	O	O(=)	M	M(=)
Supported Profiles	O	O(=)	M	M(=)

In the request/indication forms of the primitive, the network interfaces list contains the interfaces that are affected by the modification transaction. Parameter *Operation Type* identifies the operation to perform on each interface. Possible values for this parameter are "*Delete*", "*Add*" or "*Modify*". The value of *Operation Type* determines the constraints on other parameters as follows. If *Operation Type* indicates a deletion, only parameter *Interface Reference* is relevant and mandatory. If *Operation Type* indicates a modification or an addition, other parameters are constrained according to Table 8-27 and follow the rules given in 8.2.11.2.

In the response/confirm forms of the primitive, the network interfaces list shall contain the set of interfaces resulting from the modification transaction; that is, the union of the set of interfaces that were defined before the modification transaction and the set of interfaces that were present in the *RS-Site-Modify-request*/*indication* primitives and which associated *Operation Type* parameter was set to either of "*Modify*" or "*Add*". Parameters are constrained according to Table 8-27 and follow the rules given in 8.2.11.2.

8.2.13 RS-Site-Delete

This primitive is used to delete a site record in the reservation system database. Its parameters are listed in Table 8-31.

Table 8-31 – RS-Site-Delete primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=)	M
Site ID	M	M(=)	O(=)	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

Result Code can take the standard supplementary value "*No Such Site*".

See previous subclauses for other parameters description.

8.2.14 RS-Site-Directory-Inquire

This primitive is used to retrieve a list of sites recorded in the reservation system database. The sites accessible by this service may include other sites than the ones recorded by the user, such as videoconference rooms rented by the service provider, sites recorded by other users, etc. The structure of this primitive is given by Table 8-32.

Table 8-32 – RS-Site-Directory-Inquire primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=IN)	M(=RQ)
Transaction Handle		M	M(=IN)	M
Site Owner ID Filter	O	O(=)		
Site Owner Name Filter	O	O(=)		
Site Name Filter	O	O(=)		
Geographic Location Filter	O	O(=)		
Result Sites List			O	O(=)
Result Code			M	M(=)
Result Message			O	O(=)
User Data	O	O(=)	O	O(=)

Connection Handle is the local handle of the reservation connection. See 7.2.2.

Transaction Handle identifies the transaction. See 7.2.3.

Site Owner ID Filter optionally indicates a list of subscriber identifiers as a selection criteria for the retrieval. When provided, only the sites recorded by (or on behalf of) the subscribers identified by this parameter shall be returned.

Site Owner Name Filter optionally indicates a set of alphabetical intervals as a selection criteria for the retrieval. When provided, only sites recorded by (or on behalf of) subscribers whose names are comprised within the given intervals shall be returned. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Site Name Filters optionally indicates a set of alphabetical intervals as a selection criteria for the retrieval. When provided, only sites which names are comprised within the given intervals shall be returned. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Geographic Location Filter optionally indicates a set of triples of alphabetical intervals as a selection criteria over the country, region and city names. This parameter can take several forms. See clause 9 for the possibilities allowed by the protocol.

Result Sites List is the list of sites delivering the result of the inquiry transaction. This parameter is described in 8.2.14.1. Absence of this parameter in the response/confirm primitives while *Result Code* indicates a success of the transaction shall be interpreted as no visible sites for the connected user.

Result Code can take the standard supplementary value "*No Matching Criteria*".

Result Message is an optional text string that may be used to provide literal complement to *Result Code*.

User Data are optional data that may be used for functions outside the scope of this Recommendation.

8.2.14.1 Result Sites List parameter

This parameter is a list of sites descriptions returned by the reservation system. Each element of the list has the structure defined by Table 8-33.

Table 8-33 – Result Sites List – Structure of an element

Parameter	Rsp	Cnf
Site ID	M	M(=)
Site Type	M	M(=)
Site Class	C	C(=)
Site Owner ID	O	O(=)
Site Owner Name	O	O(=)
Site Name	M	M(=)
Site Geographic Information	O	O(=)
Network Interfaces	M	M(=)
Equipment Description	O	O(=)
Site Extra Information	O	O(=)
Deletion Date	O	O(=)
Modify Permission	M	M(=)

Site ID is the site identifier within the reservation system database.

Site Type gives the type of the site. See 8.2.11.

Site Class gives the class of the site. See 8.2.11.

Site Owner ID identifies subscriber who created the site record (or on behalf of whom the site record was created).

Site Owner Name gives the name of the subscriber who created the site record (or on behalf of whom the site record was created).

Site Name is text string naming the site.

Site Geographic Information optionally provides information about the geographic location of the site. This parameter has the structure defined by Table 8-26 of 8.2.11.1.

Network Interfaces lists all the network interfaces of the site. See 8.2.11.2.

Equipment Description gives information on the equipment at this site as defined in 8.2.11.

Site Extra Information may be used to describe other characteristics of the site. This parameter is a text string of an undetermined length. For instance, it may be used to give the hosting capacity of a rented videoconference room, the presence of an electronic white board, etc.

Deletion Date optionally indicates when the record of the site will be deleted from the reservation database (see 8.2.11).

Modify Permission is a boolean flag that indicates if the connected user has the right to modify or delete the site record.

8.2.15 RS-Non-Standard-Request

This primitive allows the requester to initiate a proprietary transaction or sub-transaction. As described in 7.2.3, a proprietary transaction or sub-transaction may be of the unacknowledged, acknowledged or explicitly acknowledged type. Parameters of this primitive are listed in Table 8-34.

Table 8-34 – RS-Non-Standard-Transaction primitive

Parameter	Req	Ind	Rsp	Cnf
Connection Handle	M	M	M(=)	M(=RQ)
Transaction Handle		M	M(=)	M
Parent Transaction Handle	O	O	O(=)	O(=RQ)
Acknowledgment Type			O	
Explicit Acknowledgment Required				O
Protocol Key	O	O(=)	O	O(=)
Data	M	M(=)	M	M(=)

Connection Handle identifies the reservation connection. See 7.2.2.1.

Transaction Handle identifies the proprietary transaction.

Parent Transaction Handle optionally identifies a parent transaction. See 7.2.3.4.

Acknowledgment Type can take either of value "*Explicit Acknowledgment*", "*Acknowledgment*" or "*No Acknowledgment*", depending on the type of transaction or sub-transaction for which the service is used. This parameter is relevant only in the response form of the primitive, and its absence is equivalent to "*No Acknowledgment*" (See 7.2.3.4).

Explicit Acknowledgment Required indicates that the reservation application shall explicitly acknowledge the transaction using the *RS-Transaction-Acknowledge-request* service, as indicated in 7.2.3.4. This parameter is relevant only for the confirm form of the primitive.

Protocol Key optionally indicates the protocol or data format used for the proprietary data. It is either an ASN.1 OBJECT IDENTIFIER belonging to a Recommendation, standard or non-standard protocol, or, alternatively, a non-standard identifier using the encoding conventions of [ITU-T H.221].

Data contains the information sent or received.

8.2.16 RS-Non-Standard-Data

This primitive allows the requester to send proprietary data to the peer side. As described in 7.2.3, the data sent may be correlated to a particular transaction or sub-transaction. Parameters of this primitive are listed in Table 8-35.

Table 8-35 – RS-Non-Standard-Data primitive

Parameter	Req	Ind
Connection Handle	M	M
Related Transaction Handle	O	O
Protocol Key	O	O(=)
Data	M	M(=)

Connection Handle identifies the reservation connection. See 7.2.2.1.

Related Transaction Handle optionally identifies a (sub-)transaction to which the data sent are related.

Protocol Key optionally indicates the protocol or data format used for the proprietary data (see 8.2.15).

Data contains the information sent or received.

8.2.17 RS-Transaction-Allocation

This primitive exists only under the indication form. It is used by a URST APE to indicate to the reservation application the identifier allocated for the transaction initiated. The URST APE shall raise this primitive as soon as the *rsXXXRequest* or *rsNonStandardRequest* PDU is emitted toward the peer. In order to avoid discrepancies, when a reservation application initiates several transactions simultaneously (i.e., does not wait for the corresponding *RS-Transaction-Allocation-indication* prior to initiate the next transaction), a URST APE shall sequence back *RS-Transaction-Allocation-indication* primitives in the same order as the received transaction initiation requests. However, it is recommended that, if possible, a reservation application that has initiated a transaction wait for the *RS-Transaction-Allocation-indication* primitive before initiating another transaction.

Table 8-36 – RS-Transaction-Allocation primitive

Parameter	Ind
Connection Handle	M
Transaction Handle	M
Parent Transaction Handle	O

Connection Handle identifies the reservation connection.

Transaction Handle identifies the transaction.

Parent Transaction Handle identifies the parent transaction.

8.2.18 RS-Transaction-Acknowledge

The request form of this primitive is used by a reservation application at the initiator side of a transaction to explicitly acknowledge the transaction, when the response is received from the responder side with parameter *Explicit Acknowledgment Required* set in the corresponding service primitive. On receiving an *RS-Transaction-Acknowledge-request* from the reservation application, the URST APE shall send an *rsTransactionAcknowledgeIndication* to the peer side and raise back an *RS-Transaction-Acknowledge-confirm* to the application. At the peer side the URST APE raises an indication form of the primitive to indicate that the transaction was explicitly acknowledged.

Table 8-37 – RS-Transaction-Acknowledge primitive

Parameter	Req	Ind
Connection Handle	M	M
Transaction Handle	M	M(=)

Connection Handle identifies the reservation connection.

Transaction Handle identifies the transaction acknowledged.

8.2.19 RS-Transaction-Cancel

The request form of this primitive is used by a reservation application to cancel a transaction. It shall be used exclusively by the initiator side of the transaction. On receiving an *RS-Transaction-Cancel-request* from the reservation application, the URST APE shall check the corresponding transaction number to determine whether the application is allowed to cancel the transaction, and shall immediately raise back an *RS-Transaction-Cancel-confirm* with *Result Code* set to "*Not Initiator*" if not. At the peer side, the URST APE raises an indication form of the primitive to indicate the cancel.

Table 8-38 – RS-Transaction-Cancel primitive

Parameter	Req	Ind
Connection Handle	M	M
Transaction Handle	M	M(=)

Connection Handle identifies the reservation connection.

Transaction Handle identifies the transaction to be cancelled.

8.2.20 RS-Transaction-Error

This primitive is used to indicate the peer reservation application that an error has been detected in a previous incoming transaction related message. On receiving an *RS-Transaction-Error-request* from the reservation application, the URST APE shall send an *rsTransactionErrorIndication* PDU to the peer side, setting parameter *errorCode* of this PDU to the value passed through parameter *Error Code* of the request form. On receiving an *rsTransactionErrorIndication* PDU, a URST APE shall raise an indication form of the primitive toward the local reservation application.

Table 8-39 – RS-Transaction-Error primitive

Parameter	Req	Ind
Transaction Handle	M	M(=)
Error Code	M	M(=)

Connection Handle identifies the reservation connection.

Transaction Handle identifies the transaction for which an error was detected.

Error Code gives the code of the error detected. Standard values are "*Unrecognized Non-Standard Protocol*" and "*Default Error Code*".

9 Protocol specification

9.1 Encoding of URST PDUs

The structure of URST PDUs is specified in 9.2 using the notation ASN.1 of ITU-T Rec. X.680. All URST PDUs shall be encoded for transmission by applying the Packed Encoding Rules of ITU-T Rec. X.691 using the Basic Aligned variant.

NOTE – The use of Automatic Tags in the URST protocol definition implies that the order of SEQUENCE and CHOICE structures contained within this definition effects to the actual encoded values.

9.2 URST ASN.1 Module

```
URST-PROTOCOL {itu-t(0) recommendation(0) t(20) t135(135) version(0) 2
asn1Modules(2) uRST-PROTOCOL(1)} DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

-- Export all symbols

-- Imports definitions from T.124
IMPORTS

     TextString, SimpleTextString, DiallingString, ExtraDiallingString,
SubAddressString, H221NonStandardIdentifier, Key, NonStandardParameter,
ChallengeResponse, ChallengeRequest, Privilege

FROM GCC-PROTOCOL;


--=========================
-- Types definition
--=========================

UnsignedInteger ::= INTEGER (0..MAX)

LatinCapitalLettersString ::= BMPString (FROM(latinCapitalA..latinCapitalZ))

LatinSmallLettersString ::= BMPString (FROM(latinSmallA..latinSmallZ))

LatinLettersString ::= BMPString (FROM(LatinCapitalLettersString |
LatinSmallLettersString))

DigitalString ::= BMPString (FROM (digit0..digit9))

LatinAlphanumericString ::= BMPString (FROM (LatinLettersString |
DigitalString))

GenericID ::= CHOICE {
     numberID  UnsignedInteger,
     stringID  LatinAlphanumericString
}

ConnectionID ::= SEQUENCE {
     owningFlag         BOOLEAN,
     connectionNumber   INTEGER (0..65535)
}

TransactionID ::= SEQUENCE {
     owningFlag              BOOLEAN,
     transactionNumber       INTEGER (0..65535)
}

SimpleTextStringInterval ::= CHOICE {
     thisString              SimpleTextString,
     anyFromThisString       SimpleTextString,
     anyToThisString         SimpleTextString,
     inBetween               SEQUENCE {
          inferiorString     SimpleTextString,
          superiorString     SimpleTextString
     }
}
```

```
Time ::= SEQUENCE {
     hour      INTEGER (0..23),
     minute    INTEGER (0..59)
}

TimeInterval ::= CHOICE {
     whenEver                    NULL,
     never                       NULL,
     fromThisTimeUntilWhenever   Time,
     fromNowUntilThisTime        Time,
     inBetween                   SEQUENCE {
          startTime Time,
          endTime   Time
     }
}

Day ::= SEQUENCE {
     day       INTEGER (0..31),
     month          INTEGER (1..12),
     year      UnsignedInteger OPTIONAL     -- defaults implicitly to the current
                                            -- year
}

DaysInterval ::= CHOICE {
     whenEver                    NULL,
     never                       NULL,
     todayOnly                   NULL,
     thisDayOnly                 Day,
     fromThisDayUntilWhenever    Day,
     fromNowUntilThisDay         Day,
     inBetween                   SEQUENCE {
          firstDay  Day,
          lastDay   Day
     }
}

Duration ::= SEQUENCE {
     hours          UnsignedInteger,
     minutes        INTEGER (0..59)
}

ConferenceDuration ::= CHOICE {
     permanent           NULL,
     untilTermination    NULL,
     duration            Duration
}

SiteParticipationDuration ::= CHOICE {
     allConference NULL,
     untilExit     NULL,
     duration      Duration
}

AuthenticationData ::= SEQUENCE {
     suppliedElement    CHOICE {
          passwordInTheClear     LatinAlphanumericString, -- password directly
                                                          -- supplied
          challengeResponse      ChallengeResponse
     } OPTIONAL,
     requiredElement             ChallengeRequest OPTIONAL,
     ...
}
```

```
AcknowledgmentType ::= CHOICE {
     noAcknowledgment        NULL,
     simple                  NULL,
     explicit                NULL,
     ...
}

MCUCascadingMode ::= CHOICE {
     simple                  NULL,
     masterExternal          NULL,
     masterInternal          NULL,
     unspecified             NULL,
     ...
}

VideoSwitchingMethod ::= CHOICE {
     voiceActivityDetection  NULL,
     periodicRotation        UnsignedInteger,                -- period in seconds
     fixedBroadcaster        GenericID,                      -- ID obtained at
                                                             -- site registration
     lastSpeakersMosaic      UnsignedInteger,                -- number of sources
     fixedListMosaic         SET (SIZE(1..MAX)) OF GenericID,-- sites
                                                             -- registration IDs
     ...
}

VideoFormat ::= CHOICE {
     cif           NULL,
     qcif          NULL,
     sif           NULL,
     sqcif         NULL,
     runtimeChoice NULL,
     ...
}

VideoCoding ::= CHOICE {
     h261          NULL,
     h262          NULL,
     h263          NULL,
     runtimeChoice NULL,
     ...
}

AudioCoding ::= CHOICE {
     g711-A        NULL,
     g711-mu       NULL,
     g722          NULL,
     g723          NULL,
     g728          NULL,
     g729          NULL,
     mpegAudio     NULL,
     runtimeChoice NULL,
...
}

WaitingListPolicy ::= CHOICE {
     allWaitingListNULL,
     bestEffort    NULL,
     noWaitingList NULL,
     ...
}
```

```
BillingMode ::= CHOICE {
     organizer               NULL,
     participants            NULL,
     mixed                   NULL,
     implicitBillingMode     NULL,
     ...
}

X400Address ::= SEQUENCE {
     administrationDomain    SimpleTextString,               -- A field
     country            SimpleTextString,                    -- C field
     privateDomain      SimpleTextString OPTIONAL,           -- P field
     organization            SimpleTextString OPTIONAL,      -- O field
     organizationalUnits          SET (SIZE(1..4)) OF SimpleTextString OPTIONAL,
                                                             -- OUx fields
     givenName          SimpleTextString OPTIONAL,           -- G field
     initials           SimpleTextString OPTIONAL,           -- I field
     surname            SimpleTextString OPTIONAL,           -- S field
     generationQualifier         SimpleTextString OPTIONAL, -- Q field
     commonName         SimpleTextString OPTIONAL            -- C field
}

ElectronicAddress ::= CHOICE {
     x400      X400Address,
     e-mail    SimpleTextString,
     ...
}

PersonDescription ::= SEQUENCE {
     name                    SimpleTextString,
     nameComplement          SimpleTextString OPTIONAL,
     postalAddress           SimpleTextString OPTIONAL,
     telephoneNumbers        SET (SIZE(1..MAX)) OF DiallingString OPTIONAL,
     facsimileNumbers        SET (SIZE(1..MAX)) OF DiallingString OPTIONAL,
     electronicAddresses     SET (SIZE(1..MAX)) OF ElectronicAddress OPTIONAL,
     optionalStrings         SET (SIZE(1..MAX)) OF SimpleTextString OPTIONAL,
     ...
}

JoiningMethod ::= CHOICE {
     called             NULL,
     calling            NULL,
     ...
}

ExtendedE164NetworkAddress ::= SEQUENCE {
     internationalNumber     DiallingString,
     subAddress              SubAddressString OPTIONAL,
     extraDialling           ExtraDiallingString OPTIONAL
}

TransportAddress ::= SEQUENCE {
     nsapAddress             OCTET STRING (SIZE (1..20)),
     transportSelector       OCTET STRING OPTIONAL
}

NetworkAddress ::= CHOICE {
     extendedE164       ExtendedE164NetworkAddress,
     transportAddress   TransportAddress,
     nonStandard        NonStandardParameter,
     ...
}
```

```
ChannelAggregationMethod ::= CHOICE {
     h221                NULL,
     h244                NULL,
     iso-iec-13871       NULL,     -- The actual mode of bonding is dynamically
                                   -- selected according to the procedures
                                   -- described in ISO/IEC 13871.
     nonStandard         NonStandardParameter,
     ...
}

Profile ::= CHOICE {
     simpleProfile            CHOICE {
     -- Basic transfer modes :
          speech                   NULL,     -- Simple telephony
          telephony-3kHz           NULL,     -- G.711 on ISDN or CSDN
          telephony-7kHz           NULL,     -- G.722 on ISDN or CSDN
          voice-band               NULL,     -- Modems
          frameRelay               NULL,
     -- T.120-only data profiles (T.123):
          t123-pstn-basic          NULL,
          t123-psdn-basic          NULL,
          t123-b-isdn-basic        NULL
     },
     multimediaProfile        SEQUENCE {
          profile                  CHOICE {
               h310                     NULL,
               h320                     NULL,
               h321                     NULL,
               h322                     NULL,
               h323                     NULL,
               h324                     NULL,
               h324m                    NULL,
               v61                      NULL,     -- ASVD
               v70                      NULL      -- DSVD
          },
          t120Indicator            BOOLEAN        -- indicates if T.120 protocols are
                                                  -- operated for data
     },
     dsmccDownloadProfile          NULL,
     nonStandard              NonStandardParameter,
     ...
}

AllocatedNetworkAddress ::= SEQUENCE {
     connectionDescriptor     CHOICE {
          combinedCircuits         SEQUENCE {
               circuitsList             SET (SIZE(1..MAX)) OF SEQUENCE {
                    circuitMode              CHOICE {
                         -- nominal circuits for ISDN and CSDN networks
                         digital-56k              NULL,
                         digital-64k              NULL,
                         digital-2x64k            NULL,
                         digital-384k             NULL,
                         digital-1472k            NULL,
                         digital-1536k            NULL,
                         digital-1920k            NULL,
                         multirate-base-64k       INTEGER (1..30)    -- number of B
                                                                     -- channels
                    },
                    networkAddress SEQUENCE {
                         extendedE164             ExtendedE164NetworkAddress,
                         highLayerCompatibility   HLCSupportedCodePoints OPTIONAL
               },
```

```
                channelAggregations SET OF ChannelAggregationMethod OPTIONAL
                     -- indicates the aggregation methods that may be used
                     -- to combine the circuits. Absence of this parameter
                     -- indicates that no channel aggregation shall be used.
           },
           singleAddress      SEQUENCE {
                address                 NetworkAddress,
                highLayerCompatibility  HLCSupportedCodePoints OPTIONAL
           }
      }
      },
      networkInterfaceToUse       GenericID OPTIONAL,
      profileToOperate            Profile OPTIONAL,
      ...
}

HLCSupportedCodePoints  ::= SEQUENCE {
-- Those are supported code points for IE HLC of the D
-- protocol (Rec. Q.931).
      telephony3kHz      BOOLEAN,
      telephony7kHz      BOOLEAN,
      videotelephony     BOOLEAN,
      videoconference    BOOLEAN,
      audiographic       BOOLEAN,
      audiovisual        BOOLEAN,
      multimedia         BOOLEAN,
      ...
  }

BookingSite ::= SEQUENCE {
      siteID                      GenericID,
      entryDelay                  Duration DEFAULT { hours 0, minutes 0 },
      duration                    SiteParticipationDuration DEFAULT allConference
: NULL,
      transferRate                INTEGER (1..30) OPTIONAL,  -- N× atomic channel
      joiningMethod               JoiningMethod OPTIONAL,
      waitingListIndicator        BOOLEAN DEFAULT FALSE,
      allocatedNetworkAddress     AllocatedNetworkAddress OPTIONAL,
      participantBillingRatio     INTEGER (0..100) OPTIONAL,
      conferees                   SET (SIZE(1..MAX)) OF PersonDescription
OPTIONAL,
      ...
}

ApplyingSite ::= SEQUENCE {
      siteID                      GenericID,
      entryDelay                  Duration DEFAULT { hours 0, minutes 0 },
      duration                    SiteParticipationDuration DEFAULT allConference
: NULL,
      transferRate                INTEGER (1..30) OPTIONAL,  -- N × atomic
channel
      possibleNetworkInterfaces   SET (SIZE(1..MAX)) OF GenericID OPTIONAL,
      ...
}

ConferenceModificationPolicy ::= CHOICE {
      keepPrevious       NULL,
      cancelPrevious     NULL,
      allWaitingList     NULL,
      bestEffort         NULL,
      ...
}
```

```
ConferenceStatus ::= CHOICE {
     reserved                NULL,
     partiallyReserved       NULL,
     inWaitingList           NULL,
     ongoing                 NULL,
     ...
}

ConferenceSummary ::= SEQUENCE {
     conferenceReference     GenericID,
     conferenceName          SimpleTextString,
     conferenceDescription   SimpleTextString DEFAULT "",
     conferenceOwnerID       GenericID OPTIONAL,
     conferenceOwnerName     SimpleTextString OPTIONAL,
     date                    Day,
     time                    Time,
     duration                ConferenceDuration,
     status                  ConferenceStatus,
     modifyRight             BOOLEAN DEFAULT TRUE,
     ...
}


ISDNInterface ::= SEQUENCE {
     combinedInterfaces SET (SIZE(1..MAX)) OF SEQUENCE {
          transferRatesSupported  SET (SIZE(1..MAX)) OF CHOICE {
               digital-64k             NULL,
               digital-2x64k           NULL,
               digital-384k            NULL,
               digital-1536k           NULL,
               digital-1920k           NULL,
               multirate-base-64k      CHOICE {
                    specifiedMultiples SET OF INTEGER (1..30),
                    allMultiples       NULL
               }
          },
          networkAddress     SEQUENCE {
               extendedE164            ExtendedE164NetworkAddress,
               highLayerCompatibility  HLCSupportedCodePoints OPTIONAL
          },
          ...
     },
     channelAggregations     SET (SIZE(1..MAX)) OF ChannelAggregationMethod
OPTIONAL,
     ...
}

CSDNInterface ::= SEQUENCE {
     combinedInterfaces SET (SIZE(1..MAX)) OF SEQUENCE {
          baseRate      CHOICE {
               digital-56k        NULL,
               digital-64k        NULL,
               ...
          },
          networkAddressExtendedE164NetworkAddress,
          ...
     },
     channelAggregations     SET (SIZE(1..MAX)) OF ChannelAggregationMethod
OPTIONAL,
     ...
}
```

```
ATMInterface ::= SEQUENCE {
     networkAddress     NetworkAddress,
     -- The use of several addresses for multicast calls establishment is for
further study.
     maxTransferRate         INTEGER (0..MAX) OPTIONAL,      -- expressed in
cells per seconds
     ...
}

GSTNInterface ::= SEQUENCE {
     networkAddress ExtendedE164NetworkAddress,
     ...
}

MobileInterface ::= SEQUENCE {
     networkAddress ExtendedE164NetworkAddress,
     ...
}

PSDNInterface ::= SEQUENCE {
     networkAddress NetworkAddress,
     ...
}

LANInterface ::= SEQUENCE {
     networkAddress     NetworkAddress,
     maxTransferRate         INTEGER (0..MAX) OPTIONAL,      -- expressed in
kbit/s
     ...
}

NetworkInterface ::= SEQUENCE {
     interfaceID             GenericID,
     interfaceDescription    CHOICE {
          isdnInterface      ISDNInterface,
          csdnInterface      CSDNInterface,
          atmInterface       ATMInterface,
          gstnInterface      GSTNInterface,
          mobileInterface    MobileInterface,
          psdnInterface      PSDNInterface,
          lanInterface       LANInterface,
          nonStandard        NonStandardParameter,
          ...
     },
     profilesSupported       Profile,
     ...
}

SiteType ::= CHOICE {
     mcu                     NULL,
     combinedTerminalMCU     NULL,
     terminal                NULL,
     telephone               NULL,
     specialDevice           NULL,
     nonStandard             NonStandardParameter,
     ...
}

SiteClass ::= CHOICE {
     multimedia         NULL,
     audiovisual        NULL,
     audiographic       NULL,
     visiographic       NULL,
     dataOnly           NULL,
```

```
     nonStandard        NonStandardParameter,
     ...
}


VendorModelVersion ::= CHOICE {
     h221NonStandard    H221NonStandardIdentifier,
     textForm  SEQUENCE {
          vendor    SimpleTextString,
          model     SimpleTextString,
          version   SimpleTextString
     }
}

EquipmentDescription ::= SEQUENCE {
     generalInformation      VendorModelVersion OPTIONAL,
     imuxInformation         VendorModelVersion OPTIONAL,
     audioCodec              VendorModelVersion OPTIONAL,
     audioAlgorithms         SET (SIZE(1..MAX)) OF AudioCoding OPTIONAL,
     videoCodec              VendorModelVersion OPTIONAL,
     videoAlgorithms         SET (SIZE(1..MAX)) OF VideoCoding OPTIONAL,
     t120StackInformation    VendorModelVersion OPTIONAL,
     ...
}

UserData ::= CHOICE {
     rawData       OCTET STRING,
     identifiedDataSET (SIZE(1..MAX)) OF NonStandardParameter
}

ResultCode ::= CHOICE {
     success                     NULL,
     badParameters               NULL,
     badValues                   NULL,
     transactionRefused          NULL,
     defaultFailureCode          NULL,
     authenticationInProgress    NULL,
     authenticationFailed        NULL,
     waitingList                 NULL,
     noResourcesAvailable        NULL,
     noSuchConference            NULL,
     temporaryModeForced         NULL,
     usersSpaceFull              NULL,
     noSuchSite                  NULL,
     noneMatchingCriteria        NULL,
     nonStandard                 NonStandardParameter,
     ...
}


--================================
-- Constants definition
--================================

latinCapitalA           BMPString ::= {0, 0, 0, 65}
latinCapitalZ           BMPString ::= {0, 0, 0, 90}
latinSmallA             BMPString ::= {0, 0, 0, 97}
latinSmallZ             BMPString ::= {0, 0, 0, 122}
digit0                  BMPString ::= {0, 0, 0, 48}
digit9                  BMPString ::= {0, 0, 0, 57}
implicitAccount         GenericID ::= stringID : ""
connectedSubscriber     GenericID ::= stringID : ""
noPassword              LatinAlphanumericString ::= ""
```

```
--==================
-- PDUs definition
--==================


RSConnectRequest ::= SEQUENCE {
     connectionID                ConnectionID,
     subscriberID                GenericID OPTIONAL,  -- enables anonymous
access
     authenticationData          AuthenticationData OPTIONAL,
     higherProtocolsSupported    SET (SIZE(1..MAX)) OF Key OPTIONAL,
     userData                    UserData OPTIONAL,
     ...
}

RSConnectResponse ::= SEQUENCE {
     connectionID           ConnectionID,
     subscriberID                GenericID OPTIONAL,
     authenticationData          AuthenticationData OPTIONAL,
     higherProtocolsSupported         SET (SIZE(1..MAX)) OF Key OPTIONAL,
     userData               UserData OPTIONAL,
     resultCode                  ResultCode,
     resultMessage          SimpleTextString OPTIONAL,
     ...
}


RSDisconnectIndication ::= SEQUENCE {
     connectionID  ConnectionID,
     reasonCode         CHOICE {
          normalTermination               NULL,
          disconnectingFromConference     NULL,
          defaultDisconnectionReason      NULL,
          nonStandard                     NonStandardParameter,
           ...
     } OPTIONAL,
     reasonMessage SimpleTextString OPTIONAL,
     userData      UserData OPTIONAL,
     ...
}

RSConferenceReserveRequest ::= SEQUENCE {
     connectionID           ConnectionID,
     transactionID          TransactionID,
     conferenceName         SimpleTextString,
     conferenceDescription       TextString DEFAULT "",     -- no description
     mcuCascadingMode       MCUCascadingMode DEFAULT unspecified:NULL,
     audiovisualChairControl     BOOLEAN DEFAULT FALSE,
     audiovisualChairmanPassword LatinAlphanumericString DEFAULT noPassword,
     t120ExternalConvener        BOOLEAN DEFAULT FALSE,
     t120ConductedMode      BOOLEAN DEFAULT FALSE,
     t120ConductorPrivileges     SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120CommonPrivileges        SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120ConductorPassword       LatinAlphanumericString DEFAULT noPassword,
     commonPassword         LatinAlphanumericString DEFAULT noPassword,
     globalTransferRate          INTEGER (1..30) ,    -- N x atomic channel
     videoSwitchingMethod        VideoSwitchingMethod DEFAULT
voiceActivityDetection:NULL,
     initialVideoFormat          VideoFormat DEFAULT runtimeChoice:NULL,
     initialVideoAlgorithm       VideoCoding DEFAULT runtimeChoice:NULL,
     initialAudioAlgorithm       AudioCoding DEFAULT runtimeChoice:NULL,
     waitingListPolicy           WaitingListPolicy DEFAULT noWaitingList:NULL,
     billingMode                 BillingMode DEFAULT implicitBillingMode:NULL,
```

```
    organizerBillingAccount     GenericID DEFAULT implicitAccount,
    conferenceOwnerID       GenericID DEFAULT connectedSubscriber,
    date                    Day,
    time                    Time,
    duration                ConferenceDuration DEFAULT untilTermination : NULL,
    bookingSites                SET (SIZE(2..MAX)) OF BookingSite,
    sitesDefaultJoiningMethod   JoiningMethod,
    conferenceModifierPassword  LatinAlphanumericString DEFAULT noPassword,
    userData                UserData OPTIONAL,
    ...
}


RSConferenceReserveResponse ::= SEQUENCE {
    connectionID            ConnectionID,
    transactionID           TransactionID,
    conferenceName          SimpleTextString OPTIONAL,
    conferenceDescription       TextString OPTIONAL,
    mcuCascadingMode        MCUCascadingMode OPTIONAL,
    audiovisualChairControl     BOOLEAN OPTIONAL,
    audiovisualChairmanPassword LatinAlphanumericString OPTIONAL,
    t120ExternalConvener        BOOLEAN OPTIONAL,
    t120ConductedMode       BOOLEAN OPTIONAL,
    t120ConductorPrivileges     SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
    t120CommonPrivileges        SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
    t120ConductorPassword       LatinAlphanumericString OPTIONAL,
    commonPassword          LatinAlphanumericString OPTIONAL,
    globalTransferRate          INTEGER (1..30) OPTIONAL,  -- N x atomic
channel
    videoSwitchingMethod        VideoSwitchingMethod OPTIONAL,
    initialVideoFormat          VideoFormat OPTIONAL,
    initialVideoAlgorithm       VideoCoding OPTIONAL,
    initialAudioAlgorithm       AudioCoding OPTIONAL,
    billingMode                 BillingMode OPTIONAL,
    organizerBillingAccount     GenericID OPTIONAL,
    conferenceOwnerID       GenericID OPTIONAL,
    date                    Day OPTIONAL,
    time                    Time OPTIONAL,
    duration                ConferenceDuration OPTIONAL,
    bookingSites                SET (SIZE(2..MAX)) OF BookingSite OPTIONAL,
    siteDefaultJoiningMethod        JoiningMethod OPTIONAL,
    conferenceModifierPassword  LatinAlphanumericString OPTIONAL,
    conferenceReference         GenericID OPTIONAL,
    acknowledgmentType      AcknowledgmentType (WITH COMPONENTS {simple,
explicit}) DEFAULT explicit:NULL,
    resultCode                  ResultCode,
    resultMessage           SimpleTextString OPTIONAL,
    userData                UserData OPTIONAL,
    ...
}


RSConferenceModifyRequest ::= SEQUENCE {
    connectionID            ConnectionID,
    transactionID           TransactionID,
    conferenceReference         GenericID,
    conferenceModifierPassword  LatinAlphanumericString OPTIONAL,
    defaultPolicy               ConferenceModificationPolicy DEFAULT
keepPrevious:NULL,
    conferenceName          SimpleTextString OPTIONAL,
    conferenceDescription       TextString OPTIONAL,
    mcuCascadingMode        MCUCascadingMode OPTIONAL,
    audiovisualChairControl     BOOLEAN OPTIONAL,
    audiovisualChairmanPassword LatinAlphanumericString OPTIONAL,
```

```
     t120ExternalConvener          BOOLEAN OPTIONAL,
     t120ConductedMode        BOOLEAN OPTIONAL,
     t120ConductorPrivileges       SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120CommonPrivileges          SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120ConductorPassword         LatinAlphanumericString OPTIONAL,
     commonPassword           LatinAlphanumericString OPTIONAL,
     globalTransferRate            INTEGER (1..30)  OPTIONAL, -- N x atomic
channel
     videoSwitchingMethod          VideoSwitchingMethod OPTIONAL,
     initialVideoFormat            VideoFormat OPTIONAL,
     initialVideoAlgorithm         VideoCoding OPTIONAL,
     initialAudioAlgorithm         AudioCoding OPTIONAL,
     billingMode                   BillingMode OPTIONAL,
     organizerBillingAccount       GenericID OPTIONAL,
     conferenceOwnerID        GenericID OPTIONAL,
     date                     Day OPTIONAL,
     time                     Time OPTIONAL,
     duration                 ConferenceDuration OPTIONAL,
     bookingSites                  CHOICE {
          fullRefresh         SET (SIZE(2..MAX)) OF BookingSite,
          partialRefreshSET (SIZE(1..MAX)) OF CHOICE {
               deleteSite          GenericID,
               modifySite          BookingSite,
               addSite        BookingSite
          }
     } OPTIONAL,
     siteDefaultJoiningMethod           JoiningMethod OPTIONAL,
     newModifierPassword           LatinAlphanumericString OPTIONAL,
     userData                 UserData OPTIONAL,
     ...
}

RSConferenceModifyResponse ::= SEQUENCE {
     connectionID             ConnectionID,
     transactionID            TransactionID,
     conferenceReference           GenericID,
     conferenceName           SimpleTextString OPTIONAL,
     conferenceDescription         TextString OPTIONAL,
     mcuCascadingMode         MCUCascadingMode OPTIONAL,
     audiovisualChairControl       BOOLEAN OPTIONAL,
     audiovisualChairmanPassword LatinAlphanumericString OPTIONAL,
     t120ExternalConvener          BOOLEAN OPTIONAL,
     t120ConductedMode        BOOLEAN OPTIONAL,
     t120ConductorPrivileges       SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120CommonPrivileges          SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120ConductorPassword         LatinAlphanumericString OPTIONAL,
     commonPassword           LatinAlphanumericString OPTIONAL,
     globalTransferRate            INTEGER (1..30)  OPTIONAL, -- N x atomic
channel
     videoSwitchingMethod          VideoSwitchingMethod OPTIONAL,
     initialVideoFormat            VideoFormat OPTIONAL,
     initialVideoAlgorithm         VideoCoding OPTIONAL,
     initialAudioAlgorithm         AudioCoding OPTIONAL,
     billingMode                   BillingMode OPTIONAL,
     organizerBillingAccount       GenericID OPTIONAL,
     conferenceOwnerID        GenericID OPTIONAL,
     date                     Day OPTIONAL,
     time                     Time OPTIONAL,
     duration                 ConferenceDuration OPTIONAL,
     bookingSites                  SET (SIZE(2..MAX)) OF BookingSite OPTIONAL,
     siteDefaultJoiningMethod           JoiningMethod OPTIONAL,
     newModifierPassword           LatinAlphanumericString OPTIONAL,
```

```
     acknowledgmentType      AcknowledgmentType (WITH COMPONENTS {simple,
explicit}) DEFAULT explicit:NULL,
     resultCode                   ResultCode,
     resultMessage           SimpleTextString OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}

RSConferenceCancelRequest ::= SEQUENCE {
     connectionID            ConnectionID,
     transactionID           TransactionID,
     conferenceReference          GenericID,
     conferenceModifierPassword  LatinAlphanumericString OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}

RSConferenceCancelResponse ::= SEQUENCE {
     connectionID            ConnectionID,
     transactionID           TransactionID,
     conferenceReference          GenericID,
     acknowledgmentType      AcknowledgmentType (WITH COMPONENTS{simple})
OPTIONAL,
     resultCode                   ResultCode,
     resultMessage           SimpleTextString OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}

RSConferenceListInquireRequest ::= SEQUENCE {
     connectionID            ConnectionID,
     transactionID           TransactionID,
     conferenceOwnerIDFilter      SET (SIZE(1..MAX)) OF GenericID OPTIONAL,
     conferenceOwnerNameFilter    SET (SIZE(1..MAX)) OF SimpleTextStringInterval
OPTIONAL,
     conferenceNameFilter         SET (SIZE(1..MAX)) OF SimpleTextStringInterval
OPTIONAL,
     conferenceStatusFilter       SET (SIZE(1..4)) OF ConferenceStatus,
     dateFilter                   SET (SIZE(1..MAX)) OF DaysInterval OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}


RSConferenceListInquireResponse ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     conferenceSummaries     SET (SIZE(1..MAX)) OF ConferenceSummary OPTIONAL,
     acknowledgmentType AcknowledgmentType (WITH COMPONENTS{noAcknowledgment})
OPTIONAL,
     resultCode              ResultCode,
     resultMessage      SimpleTextString OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSConferenceDetailInquireRequest ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     conferenceReference    GenericID,
     userData           UserData OPTIONAL,
     ...
}
```

```
RSConferenceDetailInquireResponse ::= SEQUENCE {
     connectionID            ConnectionID,
     transactionID           TransactionID,
     conferenceReference          GenericID,
     conferenceName          SimpleTextString,
     conferenceDescription        TextString DEFAULT "",
     conferenceOwnerID       GenericID DEFAULT connectedSubscriber,
     conferenceOwnerName          SimpleTextString OPTIONAL,
     mcuCascadingMode        MCUCascadingMode DEFAULT unspecified:NULL,
     audiovisualChairControl      BOOLEAN DEFAULT FALSE,
     audiovisualChairmanPassword LatinAlphanumericString DEFAULT noPassword,
     t120ExternalConvener         BOOLEAN DEFAULT FALSE,
     t120ConductedMode       BOOLEAN DEFAULT FALSE,
     t120ConductorPrivileges      SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120CommonPrivileges         SET (SIZE(1..MAX)) OF Privilege OPTIONAL,
     t120ConductorPassword        LatinAlphanumericString DEFAULT noPassword,
     commonPassword          LatinAlphanumericString DEFAULT noPassword,
     globalTransferRate           INTEGER (1..30),      -- N x atomic channel
     videoSwitchingMethod         VideoSwitchingMethod DEFAULT
voiceActivityDetection : NULL,
     initialVideoFormat           VideoFormat DEFAULT runtimeChoice:NULL,
     initialVideoAlgorithm        VideoCoding DEFAULT runtimeChoice:NULL,
     initialAudioAlgorithm        AudioCoding DEFAULT runtimeChoice:NULL,
     billingMode                  BillingMode DEFAULT implicitBillingMode:NULL,
     organizerBillingAccount      GenericID DEFAULT implicitAccount,
     date                    Day,
     time                    Time,
     duration                ConferenceDuration,
     bookingSites                 SET (SIZE(2..MAX)) OF BookingSite,
     siteDefaultJoiningMethod          JoiningMethod,
     conferenceModifierPassword  LatinAlphanumericString DEFAULT noPassword,
     modifyRight                  BOOLEAN DEFAULT TRUE,
     acknowledgmentType      AcknowledgmentType (WITH
COMPONENTS{noAcknowledgment}) OPTIONAL,
     resultCode                   ResultCode,
     resultMessage           SimpleTextString OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}


RSConferenceCheckAvailabilityRequest ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     dateIntervals      SET (SIZE(1..MAX)) OF DaysInterval,
     applyingSites      SET (SIZE(2..MAX)) OF ApplyingSite,
     minimumTransferRate     INTEGER (1..30) ,   -- N x atomic channel
     userData           UserData OPTIONAL,
     ...
}


RSConferenceCheckAvailabilityResponse ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     querryResult  SET (SIZE(1..MAX)) OF SEQUENCE {
          daysInterval  DaysInterval,
          possibilities SET (SIZE(1..MAX)) OF SEQUENCE {
               timeInterval       SET (SIZE(1..MAX)) OF TimeInterval,
               maximumTransferRate     INTEGER (1..30) ,      -- N x atomic
channel
               interfaceRestrictions   SET (SIZE(1..MAX)) OF SEQUENCE {
                    siteID             GenericID,
```

```
                    interfaces          CHOICE {
                         allowedInterfaces       SET (SIZE(1..MAX)) OF GenericID,
                         forbiddenInterfaces     SET (SIZE(1..MAX)) OF GenericID
                    }
               } OPTIONAL     -- all interfaces are allowed by default
          }
     } OPTIONAL,
     resultCode          ResultCode,
     resultMessage  SimpleTextString OPTIONAL,
     userData       UserData OPTIONAL,
     ...
}

RSSiteRecordRequest ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     permanentRecord         BOOLEAN DEFAULT FALSE,
     siteType           SiteType,
     siteClass          SiteClass OPTIONAL,
     siteOwnerID             GenericID DEFAULT connectedSubscriber,
     siteName           SimpleTextString OPTIONAL,
     geographicInformation   SEQUENCE {
          country       SimpleTextString OPTIONAL,
          region             SimpleTextString OPTIONAL,
          city          SimpleTextString OPTIONAL,
          postalAddress SimpleTextString OPTIONAL
     } OPTIONAL,
     networkInterfaces       SET (SIZE(1..MAX)) OF NetworkInterface,
     equipmentDescription    EquipmentDescription OPTIONAL,
     contactPerson      PersonDescription OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSSiteRecordResponse ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     permanentRecord         BOOLEAN OPTIONAL,
     siteType           SiteType OPTIONAL,
     siteClass          SiteClass OPTIONAL,
     siteOwnerID             GenericID OPTIONAL,
     siteName           SimpleTextString OPTIONAL,
     geographicInformation   SEQUENCE {
          country       SimpleTextString OPTIONAL,
          region             SimpleTextString OPTIONAL,
          city          SimpleTextString OPTIONAL,
          postalAddress SimpleTextString OPTIONAL
     } OPTIONAL,
     networkInterfaces       SET (SIZE(1..MAX)) OF NetworkInterface,
     equipmentDescription    EquipmentDescription OPTIONAL,
     contactPerson      PersonDescription OPTIONAL,
     siteID                  GenericID OPTIONAL,
     deletionDate            Day OPTIONAL,
     acknowledgmentType AcknowledgmentType (WITH COMPONENTS{simple}) OPTIONAL,
     resultCode              ResultCode,
     resultMessage      SimpleTextString OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSSiteModifyRequest ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     siteID                  GenericID,
```

```
     permanentRecord         BOOLEAN OPTIONAL,
     siteType           SiteType OPTIONAL,
     siteClass          SiteClass OPTIONAL,
     siteOwnerID             GenericID OPTIONAL,
     siteName           SimpleTextString OPTIONAL,
     geographicInformation   SEQUENCE {
          country       SimpleTextString OPTIONAL,
          region             SimpleTextString OPTIONAL,
          city          SimpleTextString OPTIONAL,
          postalAddress SimpleTextString OPTIONAL
     } OPTIONAL,
     networkInterfaces       CHOICE {
          fullRefresh        SET (SIZE(1..MAX)) OF NetworkInterface,
          partialRefreshSET (SIZE(1..MAX)) OF CHOICE {
               deleteInterface    GenericID,
               modifyInterface    SEQUENCE {
                    interfaceID        GenericID,
                    interface     NetworkInterface
               },
               addInterface       NetworkInterface
          }
     } OPTIONAL,
     equipmentDescription    EquipmentDescription OPTIONAL,
     contactPerson      PersonDescription OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSSiteModifyResponse ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     siteID                  GenericID OPTIONAL,
     permanentRecord         BOOLEAN OPTIONAL,
     siteType           SiteType OPTIONAL,
     siteClass          SiteClass OPTIONAL,
     siteOwnerID             GenericID OPTIONAL,
     siteName           SimpleTextString,
     geographicInformation   SEQUENCE {
          country       SimpleTextString OPTIONAL,
          region             SimpleTextString OPTIONAL,
          city          SimpleTextString OPTIONAL,
          postalAddress SimpleTextString OPTIONAL
     } OPTIONAL,
     networkInterfaces       SET (SIZE(1..MAX)) OF NetworkInterface OPTIONAL,
     equipmentDescription    EquipmentDescription OPTIONAL,
     contactPerson      PersonDescription OPTIONAL,
     deletionDate            Day OPTIONAL,
     acknowledgmentType AcknowledgmentType (WITH COMPONENTS{simple}) OPTIONAL,
     resultCode              ResultCode,
     resultMessage      SimpleTextString OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSSiteDeleteRequest ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     siteID             GenericID,
     userData      UserData OPTIONAL,
     ...
}
```

```
RSSiteDeleteResponse ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     siteID                  GenericID OPTIONAL,
     acknowledgmentType AcknowledgmentType (WITH COMPONENTS{simple}) OPTIONAL,
     resultCode              ResultCode,
     resultMessage      SimpleTextString OPTIONAL,
     userData           UserData OPTIONAL,
     ...
}

RSSiteDirectoryInquireRequest ::= SEQUENCE {
     connectionID            ConnectionID,
     transactionID           TransactionID,
     siteOwnerIDFilter            SET (SIZE(1..MAX)) OF GenericID OPTIONAL,
     siteOwnerNameFilter          SET (SIZE(1..MAX)) OF SimpleTextStringInterval
OPTIONAL,
     siteNameFilter          SET (SIZE(1..MAX)) OF SimpleTextStringInterval
OPTIONAL,
     geographicLocationFilter         SET (SIZE(1..MAX)) OF SEQUENCE {
          country       SimpleTextString OPTIONAL,
          region             SimpleTextString OPTIONAL,
          city          SimpleTextString OPTIONAL,
          postalAddress SimpleTextString OPTIONAL
     } OPTIONAL,
     userData                UserData OPTIONAL,
     ...
}

RSSiteDirectoryInquireResponse ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     querryResult  SET (SIZE(1..MAX)) OF SEQUENCE {
          siteType      SiteType,
          siteClass     SiteClass OPTIONAL,
          siteOwnerID        GenericID OPTIONAL,
          siteOwnerName SimpleTextString OPTIONAL,
          siteName      SimpleTextString,
          geographicInformation   SEQUENCE {
               country       SimpleTextString OPTIONAL,
               region             SimpleTextString OPTIONAL,
               city          SimpleTextString OPTIONAL,
               postalAddress SimpleTextString OPTIONAL
          } OPTIONAL,
          networkInterfaces       SET (SIZE(1..MAX)) OF NetworkInterface,
          equipmentDescription    EquipmentDescription OPTIONAL,
          contactPerson      PersonDescription OPTIONAL,
          siteExtraInformation    TextString OPTIONAL,
          deletionDate            Day OPTIONAL,
          modifyRight             BOOLEAN DEFAULT TRUE,
          ...
     } OPTIONAL,
     resultCode         ResultCode,
     resultMessage SimpleTextString OPTIONAL,
     userData      UserData OPTIONAL,
     ...
}

RSNonStandardRequest ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     parentTransactionID     TransactionID OPTIONAL,
     protocolKey             Key OPTIONAL,
     data               OCTET STRING,
```

```
     ...
}

RSNonStandardResponse ::= SEQUENCE {
     connectionID       ConnectionID,
     transactionID      TransactionID,
     acknowledgmentType AcknowledgmentType,
     protocolKey             Key OPTIONAL,
     data               OCTET STRING,
     ...
}


RSNonStandardIndication ::= SEQUENCE {
     connectionID       ConnectionID,
     relatedTransactionID    TransactionID OPTIONAL,
     protocolKey             Key OPTIONAL,
     data               OCTET STRING,
     ...
}

RSTransactionAcknowledgeIndication ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     ...
}

RSTransactionCancelIndication ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     ...
}


RSTransactionErrorIndication ::= SEQUENCE {
     connectionID  ConnectionID,
     transactionID TransactionID,
     errorCode     CHOICE {
          unrecognizedNonStandardProtocol  NULL,
          defaultErrorCode                 NULL,
          nonStandard                      NonStandardParameter,
          ...
     },
     ...
}


----------------
-- All URST PDUs
----------------

URSTPDU ::= CHOICE {
     rsConnectRequest                    RSConnectRequest,
     rsConnectResponse                   RSConnectResponse,
     rsDisconnectIndication              RSDisconnectIndication,
     rsConferenceReserveRequest          RSConferenceReserveRequest,
     rsConferenceReserveResponse         RSConferenceReserveResponse,
     rsConferenceModifyRequest           RSConferenceModifyRequest,
     rsConferenceModifyResponse          RSConferenceModifyResponse,
     rsConferenceCancelRequest           RSConferenceCancelRequest,
     rsConferenceCancelResponse          RSConferenceCancelResponse,
     rsConferenceListInquireRequest      RSConferenceListInquireRequest,
     rsConferenceListInquireResponse     RSConferenceListInquireResponse,
     rsConferenceDetailInquireRequest    RSConferenceDetailInquireRequest,
```

```
    rsConferenceDetailInquireResponse     RSConferenceDetailInquireResponse,
    rsConferenceCheckAvailabilityRequest
    RSConferenceCheckAvailabilityRequest,
    rsConferenceCheckAvailabilityResponse
    RSConferenceCheckAvailabilityResponse,
    rsSiteRecordRequest                   RSSiteRecordRequest,
    rsSiteRecordResponse                  RSSiteRecordResponse,
    rsSiteModifyRequest                   RSSiteModifyRequest,
    rsSiteModifyResponse                  RSSiteModifyResponse,
    rsSiteDeleteRequest                   RSSiteDeleteRequest,
    rsSiteDeleteResponse                  RSSiteDeleteResponse,
    rsSiteDisrectoryInquireRequest        RSSiteDirectoryInquireRequest,
    rsSiteDisrectoryInquireResponse       RSSiteDirectoryInquireResponse,
    rsNonStandardRequest                  RSNonStandardRequest,
    rsNonStandardResponse                 RSNonStandardResponse,
    rsNonStandardIndication               RSNonStandardIndication,
    rsTransactionAcknowledgeIndication
    RSTransactionAcknowledgeIndication,
    rsTransactionCancelIndication         RSTransactionCancelIndication,
    rsTransactionErrorIndication          RSTransactionErrorIndication,
    ...
}

END
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	**Terminals for telematic services**
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2008