International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.120
(01/2007)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Data protocols for multimedia conferencing

ITU-T Recommendation T.120

ITU-T

International Telecommunication Union

ITU-T Recommendation T.120

Data protocols for multimedia conferencing

Summary

The T.120-series Recommendations collectively define a multipoint data communication service for use in multimedia conferencing environments. The purpose of ITU-T Recommendation T.120 is to provide an introduction and guide to the T.120 series. ITU-T Recommendation T.120 defines the T.120 architectural model and shows the interrelationships between the constituent Recommendations. Each Recommendation in the series is outlined and the requirements for T.120 compliance are specified. Annex A specifies T.120 channel and token allocations. Annex B specifies MCS domain parameters. Annex C specifies lightweight profiles for the T.120 architecture.

This revised version of ITU-T Recommendation T.120 introduces a number of clarifications to the previous version.

Source

ITU-T Recommendation T.120 was approved on 13 January 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.120

Data protocols for multimedia conferencing

1 Scope

This Recommendation introduces a suite of standards, collectively referred to as the T.120 series.

This Recommendation describes the T.120 system model that provides an architecture for multipoint data communication in a multimedia conferencing environment. It provides an introduction and functional description of the Recommendations that go to make up the T.120 infrastructure. In addition, it provides an overview to other Recommendations in the series that provide standardized application protocol functionality.

This Recommendation defines the criteria for compliance when the T.120 data protocols are used in a conferencing or group-working environment.

This Recommendation only covers completed work contained in approved Recommendations. When new Recommendations are approved, supporting text will be generated for inclusion in this Recommendation at the next publication date.

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. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T T.121] ITU-T Recommendation T.121 (1996), *Generic application template*.

[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*.

[ITU-T T.126] ITU-T Recommendation T.126 (1997), *Multipoint still image and annotation protocol*.

[ITU-T T.127] ITU-T Recommendation T.127 (1995), *Multipoint binary file transfer protocol*.

3 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

APE Application Protocol Entity

ARM Application Resource Manager

ASE Application Service Element

B-ISDN Broadband Integrated Services Digital Network

CSDN	Circuit Switched Data Network
GAT	Generic Application Template
GCC	Generic Conference Control
ISDN	Integrated Services Digital Network
LAN	Local Area Network
MBFT	Multipoint Binary File Transfer
MCS	Multipoint Communication Service
MCU	Multipoint Control Unit
MSIA	Multipoint Still Image and Annotation
PDU	Protocol Data Unit
PSDN	Packet Switched Data Network
PSTN	Public Switched Telephone Network
QoS	Quality of Service
SI	Still Image (SI is a commonly used abbreviation for MSIA)

4 Overview

The T.120-series Recommendations collectively define a multipoint communication service for use in multimedia conferencing environments. The purpose of this Recommendation is to provide an introduction and guide for the T.120 series, showing the interrelationships between the constituent Recommendations and to define the requirements for compliance to this Recommendation for conferencing.

This Recommendation provides facilities to establish and manage interactive communications (conferences) involving two or more participants on and between a variety of different networks. It provides a comprehensive data communication service for those participants, which is independent of the underlying network. Within a conference it allows communications to be established between any combination of conference participants. This Recommendation also provides support for applications and their associated protocols, defining start-up mechanisms and capability exchange procedures, etc.

This Recommendation makes provisions to ensure interoperability of commonly required functionality such as file transfer, still image exchange and shared whiteboards through the definition of standardized application protocols.

The T.120 protocols provide a means of telecommunicating many forms of Data/Telematic information between two or more multimedia terminals and of managing such communication. They provide a multipoint data communication service that has a particular application in multimedia conferencing.

The T.120 protocols are suitable for use on many types of network: PSTN, ISDN, CSDN, PSDN, B-ISDN, LANs. They provide the capability for seamless interworking of applications between terminals connected to different networks.

The T.120 protocols provide:

- support for conference establishment among a group of network nodes (such as conference terminals and MCUs);
- mechanisms to identify the participating nodes and a comprehensive roster and capability exchange mechanism;

- flexible management of communication between any combination of these elements.

The T.120 protocols can handle one or more simultaneous conferences. A terminal may participate in more than one of these if authorized to do so. The *convenor* of a conference may control the participation in that conference and the information which flows in that conference.

In a conference that allows conductorship, the convenor may delegate part or all authority to the conductor. If a conference enters conducted mode, application protocols that are "conductor aware" modify their behaviour, as specified by their protocol for this mode of operation.

This Recommendation imposes few inherent constraints on the configuration of connections between conference nodes (terminals and MCUs): but they must be arranged in a hierarchy, with a single node at the top of a tree. Nodes may be all connected to one star-point, or connected one to two others in a chain, or a chain of star-points, and so on, as long as it is clear for each connection which direction is upward and there are no loops. The top node should be present from the beginning of a conference, since any change in the top can be disruptive.

No constraint is placed on the rate or volume of information transmitted within the various media; the T.120 protocols have the capability to organize different rates of information flow, within the constraints imposed by the type of network and connections established thereon. They allow relative priorities to be set by the applications using the T.120 protocols.

The structure of the T.120 protocols is described in clause 6. Not all of the T.120 protocol provisions are mandatory: T.123, T.122/125, and T.124 are mandatory for conferencing and group working environments. The remainder are conditional: where functionality covered by the standards is provided, the standard protocols of the T.120 series must be implemented (see clause 9 for T.120 compliance requirements). This ensures that it is always possible to achieve a basic level of interworking, and does not prohibit customized enhancements and negotiation of proprietary modes if (and only if) all participating elements are able to support such modes.

5 Introduction to multipoint multimedia communication

Traditionally, telephony services have been constrained to point-to-point operation. In order to support group activities such as meetings, conferences etc., involving physically separated participants, there is a requirement to join together more than two locations. The term "multipoint communication" simply describes the interconnection of multiple terminals. Normally, a special network element, known as a Multipoint Control Unit (MCU), or more simply a bridge, is required in order to provide this function.

A conference typically refers to a group of geographically dispersed nodes that are electronically joined together and that are capable of exchanging audiographic and audiovisual information across various communication networks.

Conference participants may have access to various types of media handling capabilities such as audio only (telephony), audio and data, audio and video, or audio, video and data.

The T.120 series of Recommendations define the component which is used to provide both a data communications service, and a management service for any other media services present.

T.120 protocols provide the infrastructure required to provide data services for many types of conferencing and group working, making it suitable for a diverse range of application areas. It is expected to find use in videotelephony and audiographic conferencing as well as other forms of multipoint multimedia communication.

This Recommendation regards point-to-point connections as the simplest form (a degenerate case) of a multipoint connection. Both forms of connection are supported by the T.120 protocols. Terminals with multiple communication ports (each with an appropriate T.120 transport stack) can act as T.120 data bridges and allow multipoint connections to be established involving three or more nodes. Figure 1 b) shows a four site conference with multiport terminals acting as data bridges.

MCUs are nodes that do not normally support terminal functionality. They act as bridging nodes, bridging data and other media streams present in the connections. Figure 1 c) shows an example of how three MCUs may be connected to bridge a group of terminals.

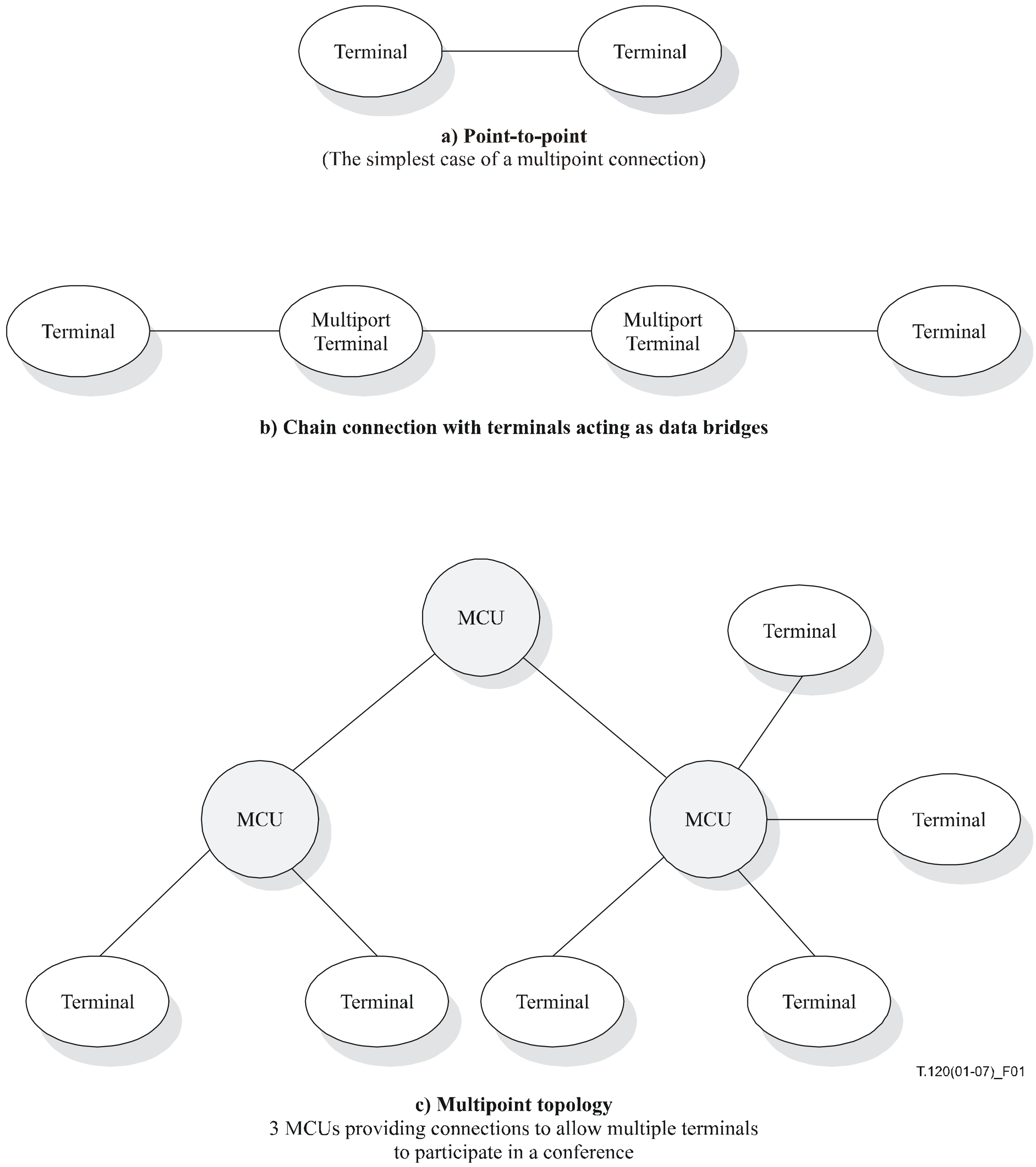


Figure 1 – Examples of multipoint conference configurations showing various connection topologies and node types

Figure 2 is an example of a conference that involves terminals of diverse capability, on multiple different networks and across administrations.

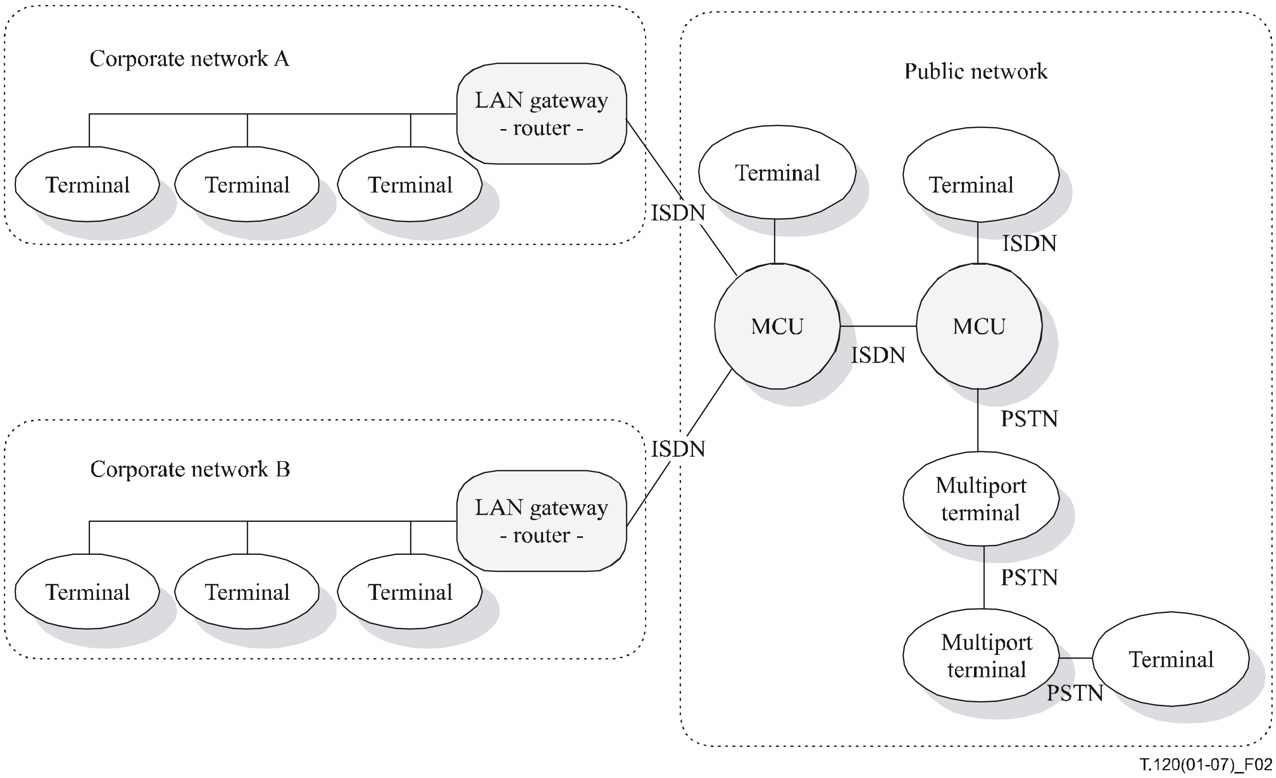


Figure 2 – Example of a mixed-network conference topology

6 The T.120 system model

The T.120 model is comprised of a communications infrastructure and the application protocols that make use of it. Figure 3 shows the full model with both standardized and non-standardized applications. The model serves to show both the scope of the T.120 series of Recommendations (indicated by the shaded background) and the relationship between each of the Recommendations and other components in the system.

Generally, each layer provides services to the layer above and communicates to its peer(s) by sending Protocol Data Units (PDUs) via services provided by the layer below.

This clause will address each of the major functional levels in Figure 3: User applications, Application protocols, Node controller, Communications infrastructure and Networks.

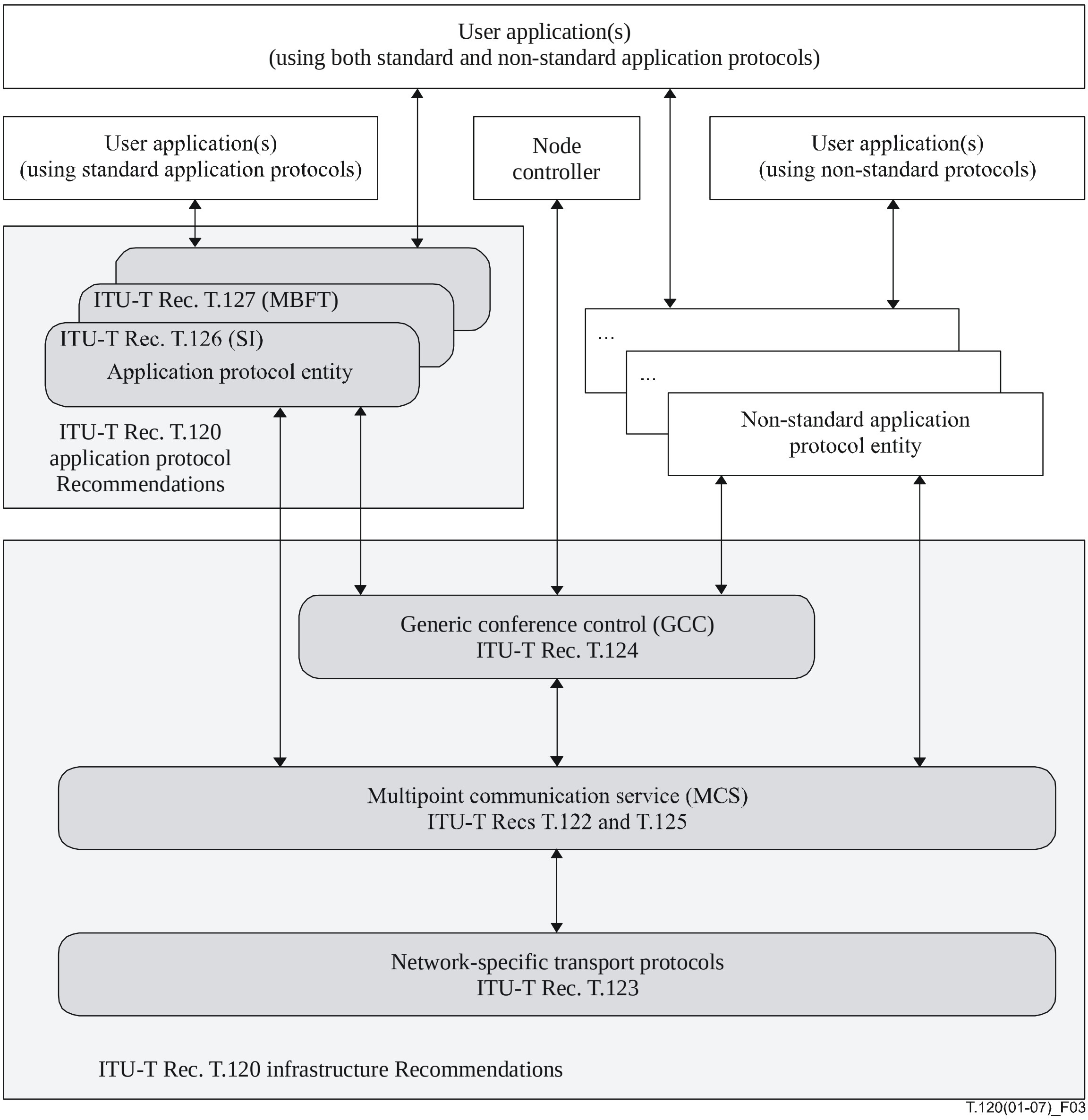


Figure 3 – T.120 system model

6.1 User applications

Applications *per se* are not the subject of standardization in the T.120-series Recommendations. Applications that use the services offered by the T.120 series Recommendations will generally be multipoint aware and designed to use the T.120 services provided by GCC and MCS. These applications are termed "User Applications" and they may use any combination of standardized and non-standard protocols to communicate with peer user applications. The T.120 environment supports multiple user applications concurrently operating in the same conference by providing mechanisms for the applications to coordinate the use of communications resources. The Generic Application Template [ITU-T T.121] provides guidance to user application developers on how to utilize the T.120 infrastructure in a coherent and consistent way. A user application addresses those tasks which have no direct effect on interworking (e.g., user interface) and which may thus be product and platform specific. The influence of the user application is felt at other sites through the application protocols it employs.

6.2 Application protocols

Application protocols comprise a set of protocol data units (PDUs) and associated actions for application peer-to-peer(s) communication. These may be proprietary protocols or they may be standardized by the ITU-T or other international or national standards bodies. The T.120 series Recommendations includes a set of application protocols designed to meet the needs of multipoint conferencing. These protocols define minimum requirements in order to ensure interworking between different implementations. [ITU-T T.127] provides simultaneous multipoint file transfer. [ITU-T T.126] provides still image viewing and annotation, shared whiteboard, and facsimile. A given application may use any combination of standardized and non-standard application protocols.

An Application Protocol Entity is an instance of an application protocol. It may be considered as comprising of two functional components: the application resource manager (ARM), that provides the generic functionality relevant to all protocols; and the application service element (ASE), that provides the application-specific functionality. Both of these are described further in the T.121 generic application template recommendation. [ITU-T T.121] presents templates and guidelines which may assist in the definition of new application protocols.

6.3 Node controller

The node controller is the element that provides the T.120 management role at a terminal or MCU. It issues primitives to the GCC provider which starts and controls the communication session. The node controller itself is outside the scope of the T.120 series of Recommendations, and only where it communicates to GCC are its interfaces defined. However, correct interaction with GCC imposes some normative requirements on the node controller.

6.4 Communications infrastructure

The communications infrastructure provides multipoint connectivity with reliable data delivery. It can accommodate multiple independent applications concurrently using the same multipoint environment. Connections between nodes can be any combination of circuit-switched telecommunications networks and packet-based LANs and data networks. The T.120 infrastructure is composed of three standardized components: Generic conference control (GCC), the multipoint communication service (MCS) and transport protocol profiles for each of the supported networks.

Generic conference control

GCC provides a set of services for setting up and managing the multipoint conference. It provides access control and arbitration of capabilities. GCC facilities are used by applications to coordinate the use of MCS channels and tokens. GCC facilities can be used to query an MCU or multiport terminal node to find a desired conference. Multiple applications can be running on any given node and can be dynamically launched, used and shut down during a conference. As part of the management role, peer GCC providers exchange information about the applications present and their capabilities. GCC also makes a centralized registry facility available to applications in order to identify dynamically assigned resources such as channels and tokens.

Multipoint communication service

MCS provides a general purpose multipoint connection-oriented data service. It collects point-to-point transport connections and combines them to form a multipoint domain. Within that domain, a large number of logical channels are provided that can provide one-to-one, one-to-many and many-to-one data delivery. Nodes within an MCS domain are hierarchically organized in a tree structure. Data delivery normally follows the most efficient path to the nodes that are to receive the data, but a mechanism is provided to guarantee that data originating from different nodes is received in the same sequence at all nodes. MCS acts as a resource provider to the layers above, independent of the underlying network, providing channels and token resources on demand. Tokens are provided for applications to use for coordinating events and processes.

Network-specific transport protocols

[ITU-T T.123] provides reliable point-to-point sequenced data delivery of MCS PDUs and segmentation of that data if necessary. [ITU-T T.123] specifies a protocol stack for each particular network supported. [ITU-T T.123] presents a uniform OSI Transport Service interface to the MCS layer above.

6.5 Networks

The T.120-series Recommendations suite provide for operation over the following networks:

- ISDN – Integrated Services Digital Network as defined in the I-series Recommendations.
- CSDN – Other (switched or permanent) digital circuits.
- PSDN – Packet Switched Data Network using ITU-T Rec. X.25.
- PSTN – Public Switched Telephone Network (or compatible service).
- Use of this Recommendation on other networks such as B-ISDN and LANs is currently under study. Alternative protocol stack profiles may be defined in the future.

The approach taken with the T.120 architecture leads to the multipoint routing information being located in MCS above the transport stacks; this is the key to the network independence that can be achieved with this Recommendation.

The various Recommendations that comprise the T.120-series Recommendations suite are described in greater detail in clauses 7 and 8.

7 T.120 infrastructure Recommendations

The T.120 protocols are designed to operate over a wide range of networks and indeed to facilitate communication between endpoints on a mixture of networks. The differences in T.120 operation for the various networks are confined to the lowest layers as detailed in [ITU-T T.123].

7.1 Protocol stacks for audiographic and audiovisual conferencing – ITU-T Rec. T.123

[ITU-T T.123] defines the network-specific transport stacks for each supported network. Generally, existing link layer protocols appropriate to each network are selected and then mapped into a common interface layer, thus defining a transport profile for a given network. At Transport level the conference is seen as a group of point-to-point connected pairs (different pairs may be on different networks). The multipoint communication service (MCS), takes the transport pairs from the layer below it and maps them into a multipoint domain. See Figure 4.

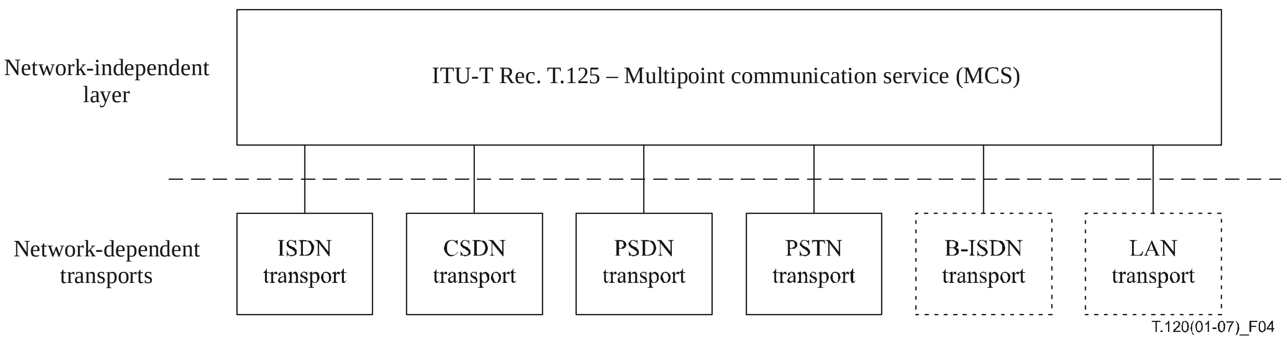


Figure 4 – Protocol stacks – defined in ITU-T Rec. T.123

7.2 Multipoint communication service (MCS) – ITU-T Recs T.122 and T.125

The multipoint communication service (MCS) is contained in two Recommendations: [ITU-T T.122] defines the MCS service and [ITU-T T.125] the protocol which is mandatory.

MCS is a key element in the infrastructure of this Recommendation. It takes the point-to-point transport connections provided by the layers below it and combines them to form a multipoint domain. The T.120 protocol can handle a multiplicity of domains – each effectively an independent conference or group activity. Nodes may participate in more than one domain.

A domain with only two terminal nodes participating mirrors the traditional point-to-point communications model and is fully supported by MCS but with the potential of introducing more nodes as required.

Each connected pair in an MCS domain is ordered, so that one node is higher than the other. The domain must rise hierarchically (without loops) to a top node. The MCS provider at the top node is assigned the Top Provider role, acting as the resource server for the domain.

The MCS provides transport of control and data streams from one terminal to any or all others in the conference. It does not need to know anything about the content of application data streams.

MCS introduces the concept of channels to provide data distribution within a domain; an MCS channel connects all the nodes which have expressly joined it. It is not required that a channel be joined in order to send to it. Data sent to a channel will be conveyed to all other nodes which have joined that channel. MCS supports four types of channels.

A "static" channel exists when a domain is established. MCS reserves a block of channel identifiers for static use. Static channels may be given preassigned roles by the protocols making use of MCS services (such roles are not of concern to MCS). Annex A defines the mapping of preassigned roles to channel identifiers. Any node may join a static channel, and there is no "owner".

A "dynamic" channel is created on request by any node or application. Dynamic channels come in three forms: "Multicast", "Private" and "Single Member". Multicast (Assigned) channels are similar to static channels in that they are open access – any terminal may join them, and there is no owner. Private channels however are owned by their creator and are joined by invitation only, thus forming a private or closed-user-group. A "Single Member" channel is normally used to provide a User Identifier, giving the owner a unique address within the domain.

There are two specified types of data: ordinary data is sent by the shortest route to its destinations – nothing governs the order in which information arrives from different source terminals, so it can happen that presentations to users may not be identical. Uniform sequenced data, on the other hand, is routed to a common point (the top MCU in the connection hierarchy, called the "top provider") and distributed thence to all relevant terminals in the same order; clearly this may take longer than for ordinary data.

MCS supports four data priorities; according to the priority requested in the header of data primitives coming down into MCS, they are routed into one of up to four corresponding transport connections.

MCS provides a token management service and is able to support the use of a range of tokens. In order to provide exclusivity, and thus consistency across a domain only the Top Provider is able to execute actions on tokens. MCS supports the following token actions; grab, inhibit, give, please and release. The role allocated to tokens is assigned by the layers above MCS (and is not the concern of MCS).

Much of the power and flexibility of MCS derives from its provision of services in a manner that is independent of the underlying network connections. This allows portability across networks and an inherent capability to interwork between terminals on different networks.

7.3 Generic conference control (GCC) – ITU-T Rec. T.124

The generic conference control (GCC) service and protocol is defined in [ITU-T T.124]; it resides above MCS in the T.120 stack infrastructure. GCC is a mandatory component for peer-to-peer conferencing and group working environments, providing a high-level framework for management and control in support of a diverse range of terminals and MCUs.

A GCC conference has a direct correspondence to an MCS domain. GCC provides mechanisms for the creation, control and the termination of conferences. It also makes provision for building and distributing the conference and application databases.

MCS supports four data priorities, one of which (top priority) is reserved for exclusive use by GCC for control and management. The remaining three priorities are available for application use.

GCC reserves a block of MCS tokens and designates them to be static and thus assigned functionality by this Recommendation (Annex A). The remaining tokens are designated dynamic and their functionality is assigned in the registry and is valid only for the duration of a conference session.

The conference roster contains a record of the conference configuration, that includes such things as the name of the conference, the types of participating nodes (terminal, MCU or multiport terminal), and site and participant information for each node.

When a node joins a conference it announces its presence to that conference. This causes the conference roster to be updated and distributed.

GCC supports the enrolment in a conference of application protocols. Each GCC provider maintains a local application roster containing information and capabilities for its enrolled application protocols. The local rosters are sent to the top node of a conference where a conference application roster is compiled and then distributed. In this way all nodes learn of the additional capabilities of their peers.

Once enrolled, application protocols entities are free to use MCS resources as described in the generic application template [ITU-T T.121]. However, if the conference enters conducted mode, permission may be required before an application protocol entity can act.

8 Application protocol Recommendations

The T.120 series includes application protocols that provide commonly required functionality to user applications, in a way that ensures a guaranteed level of interworking across a diverse range of terminals with differing capabilities.

8.1 The generic application template (GAT) – ITU-T Rec. T.121

[ITU-T T.121] defines a conceptual model of a T.120 application protocol. It serves as a guide to application protocol developers, ensuring a consistent approach to the development of application protocols. It also provides guidance to user application developers on how to utilize the T.120 infrastructure to best effect.

The model consists of two functionally distinct parts: the application resource manager (ARM) and the application service element (ASE). The ARM is responsible for managing GCC and MCS resources on behalf of the ASE.

The ARM provides generic functionality, available to all application protocols. It effectively defines a template to which specific application functionality can be added. The functionality provided by the template is required by all application protocols, both standard and non-standard. Compliance to the template is mandatory for standardized application protocols. Although not mandatory for non-standard application protocols, it is recommended, to ensure consistency and reduce the potential for unforeseen interaction between different protocols.

The ASE provides application-protocol-specific functionality, independent of the type of channel and token resources provided to it by the ARM. See Figure 5.

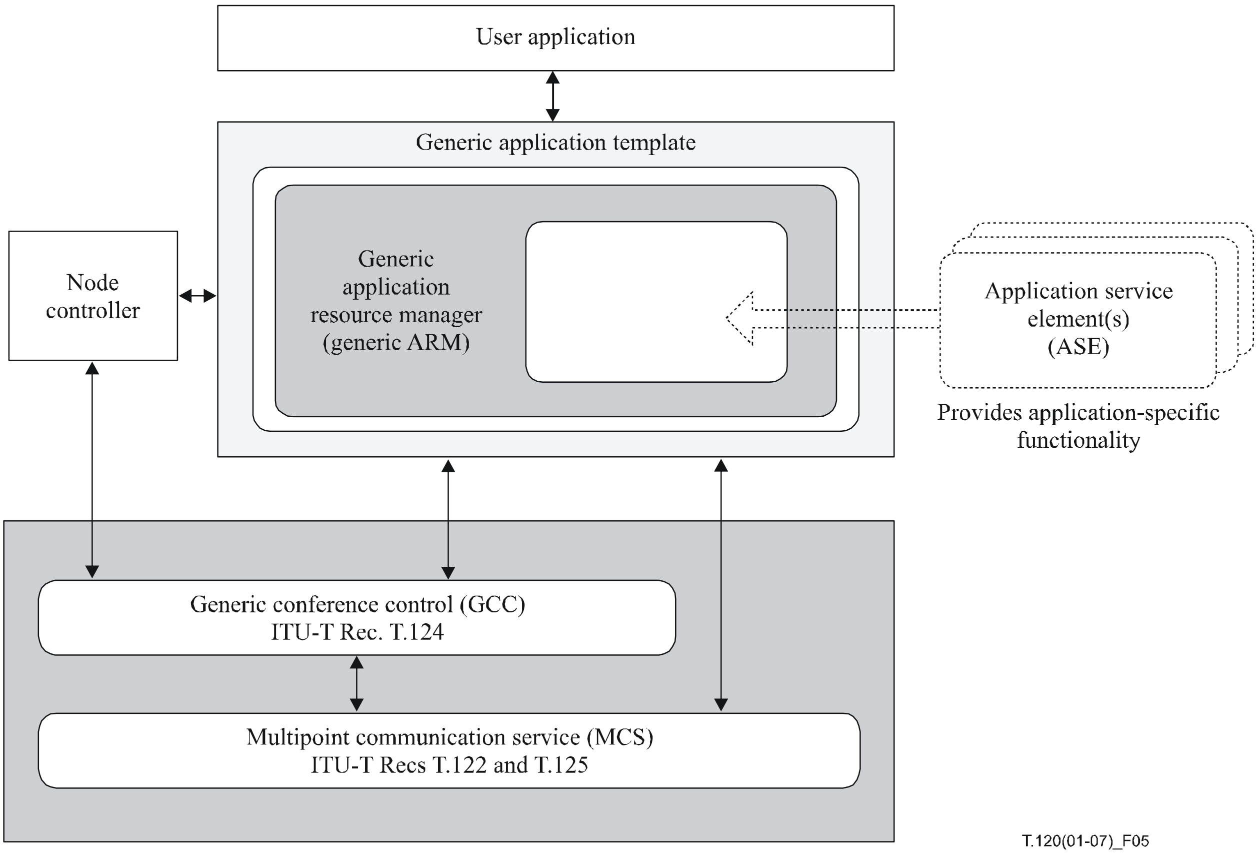


Figure 5 – Generic application template

8.2 Multipoint still image and annotation protocol (MSIA) – ITU-T Rec. T.126

[ITU-T T.126] defines the protocol to be used by a broad set of user applications that require interoperable graphical information exchange in a multi-vendor environment. It can be employed by user applications requiring simple whiteboarding, annotated image exchange and hard copy image exchange as well as for more advanced functions such as remote computer application piloting and screen sharing. The protocol manages the conference-wide synchronization of multi-plane/multi-view graphical workspaces. An extensible set of bitmap, pointer and parametric drawing primitives can be directed to these workspaces. Advanced options such as keyboard and pointing device signalling to support computer application remote piloting and screen sharing are also defined. All aspects of the protocol have provisions for in-band extensibility to allow any new or extended primitives that are not defined to be added and detected within a conference.

8.3 Multipoint binary file transfer (MBFT) – ITU-T Rec. T.127

[ITU-T T.127] defines a protocol to support interchange of binary files within an interactive conferencing or group working environment. It does not impose any restrictions on the content of the files to be transferred. It is a versatile, lightweight protocol which provides the core functionality to allow interworking between applications requiring a basic general-purpose file transfer capability. It also has the flexibility to meet the demands of more sophisticated applications. Mechanisms are provided which facilitate both distribution and retrieval of files.

A basic file transfer application conforming to [ITU-T T.127] may simply offer the ability to broadcast one file at a time to all applications which support the MBFT protocol. Optional advanced features defined in [ITU-T T.127] include:

- broadcast of multiple files simultaneously;
- private distribution of files to a selected subset of the conference;
- conductor control of file distribution.

8.4 T.120 extensions for audiovisual control

For further study

8.5 Proprietary extensions to standardized protocols

Extensions to standardized protocol may be made by making use of the non-standard PDU or other elements that have been reserved in the relevant specifications for proprietary use.

8.6 Non-standard application protocols

Provision of additional functionality may be made by creation of additional proprietary application protocols which use the resources of MCS and GCC. Where separate proprietary protocols are defined, it is recommended that the Generic Application Template [ITU-T T.121] be used as the model for the generic component of Application Protocol, the Application Resource Manager. Guidance on extension, modification and reuse of existing application protocols is given in [ITU-T T.121]. A non-standard application protocol entity must use GCC in a way that does not risk conflict with other application protocol entities. In particular, it shall not use static channels or static tokens and must enroll using a non-standard application protocol key.

9 T.120 compliance

For use within a multimedia conferencing environment, this Recommendation requires:

- compliance with the Transport Protocol stack profile [ITU-T T.123] for the selected networks;
- compliance with the Multipoint Communication Service protocol [ITU-T T.125];
- compliance with the mandatory parts of Generic Conference Control [ITU-T T.124];
- compliance with the mandatory parts of any standardized application protocols that have scope covering functionality supported by the user applications.

Compliance to this Recommendation is specified in Table 1.

Options within each item are clearly set out within the requisite Recommendation; thus, except where otherwise stated, in this table compliance means to the minimum requirement of the Recommendation.

Table 1 – T.120 compliance table

Item (Recommendation)	Status	Conditions
T.121	Conditional	Mandatory when specified by an application protocol
T.123	Mandatory	Basic mode profile, according to network type
T.124 (GCC)	Mandatory	Mandatory protocol elements – As specified in Table 6-1 of [ITU-T T.124], according to node type Conditional protocol elements – Requirement dependent on the needs of application protocols to be supported at the node In addition, a node attempting to enter an existing conference via GCC-Conference-Join request shall specify exactly the minima and maxima of the MCS domain parameter values defined in Annex B as its allowed range of negotiation
T.125 (MCS)	Mandatory	The defined protocol is mandatory. The domain parameter values selected for an MCS domain shall lie within the ranges defined in Annex B.
T.126 (MSIA)	Conditional	Mandatory when a node makes use of one or more of the following features: • exchange of soft copy still images; • exchange of hard copy still images (including FAX); • shared whiteboard functionality; • exchange of annotated soft copy still images. In the above cases it must be possible to activate the standard base session for [ITU-T T.126] for the same purpose. Annex A of [ITU-T T.126] defines those parts of [ITU-T T.126] that are mandatory for each of the above functions
T.127 (MBFT)	Conditional	Mandatory for general purpose file transfer applications It must be possible to activate the standard base session for [ITU-T T.127]

Annex A

T.120 channel and token allocations

(This annex forms an integral part of this Recommendation)

A.1 Static channels

Symbolic name	MCS channel ID	Description	Recommendation
GCC-CHANNEL-0	1	GCC broadcast channel	T.124
GCC-CHANNEL-1	2	Convenor channel	T.124
SI-CHANNEL-0	8	MSIA communications channel	T.126
MBFT-CHANNEL-0	9	Control channel	T.127
MBFT-CHANNEL-1	10	Data channel	T.127
AS-CHANNEL-0	11	Application sharing channel	T.128
CHAT-CHANNEL-0	12	Chat data channel	T.134

A.2 Static tokens

Symbolic name	MCS token ID	Description	Recommendation
GCC-TOKEN-0	1	Conductors token	T.124
SI-TOKEN-0	8	Bit map create token	T.126
SI-TOKEN-1	9	Workspace create token	T.126
MBFT-TOKEN-0	10	File transmit token	T.127
MBFT-TOKEN-1	11	File receive token	T.127

A.3 Standard application protocol session identifiers

Application protocol Recommendation	MCS channel identifier used as session identifier
T.126	SI-CHANNEL-0
T.127	MBFT-CHANNEL-0

Annex B

MCS domain parameters

(This annex forms an integral part of this Recommendation)

The MCS protocol [ITU-T T.125] states that MCS providers allocate resources and execute procedures according to the domain parameters established for any given domain. The establishment of the first MCS connection between two nodes fixes the domain parameters for all subsequent joiners.

In order to avoid the possibility of these values being set at values inconsistent with the requirements of later nodes, this Recommendation requires that MCS domain parameter settings adhere to the specified ranges in all situations.

In addition, a node attempting to enter an existing conference via GCC-Conference-Join request shall specify exactly the minima and maxima of the MCS domain parameter values defined in this Annex as its allowed range of negotiation.

NOTE – It is possible for MCS domain parameters to be negotiated out-of-band prior to conference establishment. However, only in cases where it could be guaranteed that all participating nodes can conform to the agreed values, is it permissible to specify values other than those defined in this annex. This method is not recommended, and could cause subsequent connections to fail.

a) Maximum number of MCS channels that may be in use simultaneously. This includes channels that are joined by any user, channel IDs that have been assigned, and private channels that have been created.

Minimum value	10
Maximum value	65 535

b) Maximum number of user ids that may be assigned simultaneously. This is a sub-limit within the constraint of the preceding parameter.

Minimum value	10
Maximum value	64 535

c) Maximum number of token ids that may be grabbed or inhibited simultaneously.

Minimum value	0
Maximum value	65 535

d) Number of data transfer priorities implemented. This equals the number of transport connections in an MCS connection. An MCS user may still send and receive data with priorities outside the limit. However, such priorities may be treated the same as the lowest priority that is implemented.

Minimum value	1
Maximum value	4

e) Enforced throughput. Although global flow control limits data transfer within a domain to the rate of the slowest receiver, receivers must not be allowed to run arbitrarily slowly. Otherwise, one party in a conference may obstruct all others. This parameter instructs MCS providers to enforce a minimum receiving rate at each MCS attachment and over each downward MCS connection. Violators run the risk of being involuntarily detached or disconnected, respectively.

Minimum value	0 (no minimum rate enforced)
Maximum value	0 (no minimum rate enforced)

f) Maximum height. This constrains the height of all MCS providers, in particular the top provider.

Minimum value	2
Maximum value	100

g) Maximum size of domain MCSPDUs. Global flow control is based on buffering domain MCSPDUs within an MCS provider (but not connect MCSPDUs). For simplicity, fixed-size buffers are assumed. An MCS provider shall not generate larger MCSPDUs. This constrains the amount of information that can be packed into a single control MCSPDU and suggests where unlimited user data should be segmented in data MCSPDUs.

Minimum value	128 octets
Maximum value	4096 octets

h) Protocol version. This takes one of two values defining different encodings for domain MCSPDUs.

Minimum value	Version 2 (Packed Encoding Rules for all but connect PDUs)
Maximum value	Version 2 (Packed Encoding Rules for all but connect PDUs)

Annex C

Lightweight profiles for the T.120 architecture

(This annex forms an integral part of this Recommendation)

Overview

This annex provides a lightweight T.120 infrastructure (referred to throughout this annex as "T-*Lite*") that suffices as a platform to run applications such as audio-video control or text telephony. The lightweight profiles for the following T.120-related protocols are defined here:

- T.123;
- T.122/T.125;
- T.124;
- T.121.

These profiles preserve their T.120 compliance "on the wire" when interacting with other T.120 (T-*Lite* as well as conventional) nodes. They also can be implemented with less effort than traditional T.120 enhanced multimedia terminals. An important point to consider is that T.120 services that are not included in the profiles defined here can be added back, on an as-needed basis. For example, if an implementer wishes to support conductorship within their audio-video-data product, they can simply add conductorship to the T-*Lite* profile defined here. Another example is the addition of other APEs. If an application is limited to using a single T.120 APE (say T.128), the thin T.124 implementation could be limited to support a single roster for that APE. If support for additional APEs becomes a requirement, support for additional rosters can be added to T.124. The important consideration is: what services are required by the APEs sitting on top of the T.120 infrastructure. If a service is not required it need not be implemented (unless specified in this profile).

It is important for a T-*Lite* developer to know, in advance, which specific APEs are to be supported by the stack. This is important because the Application Protocol Key (which is a unique identifier for the APE) is used to filter unsupported Application Rosters received by the lightweight node (see C.3.5).

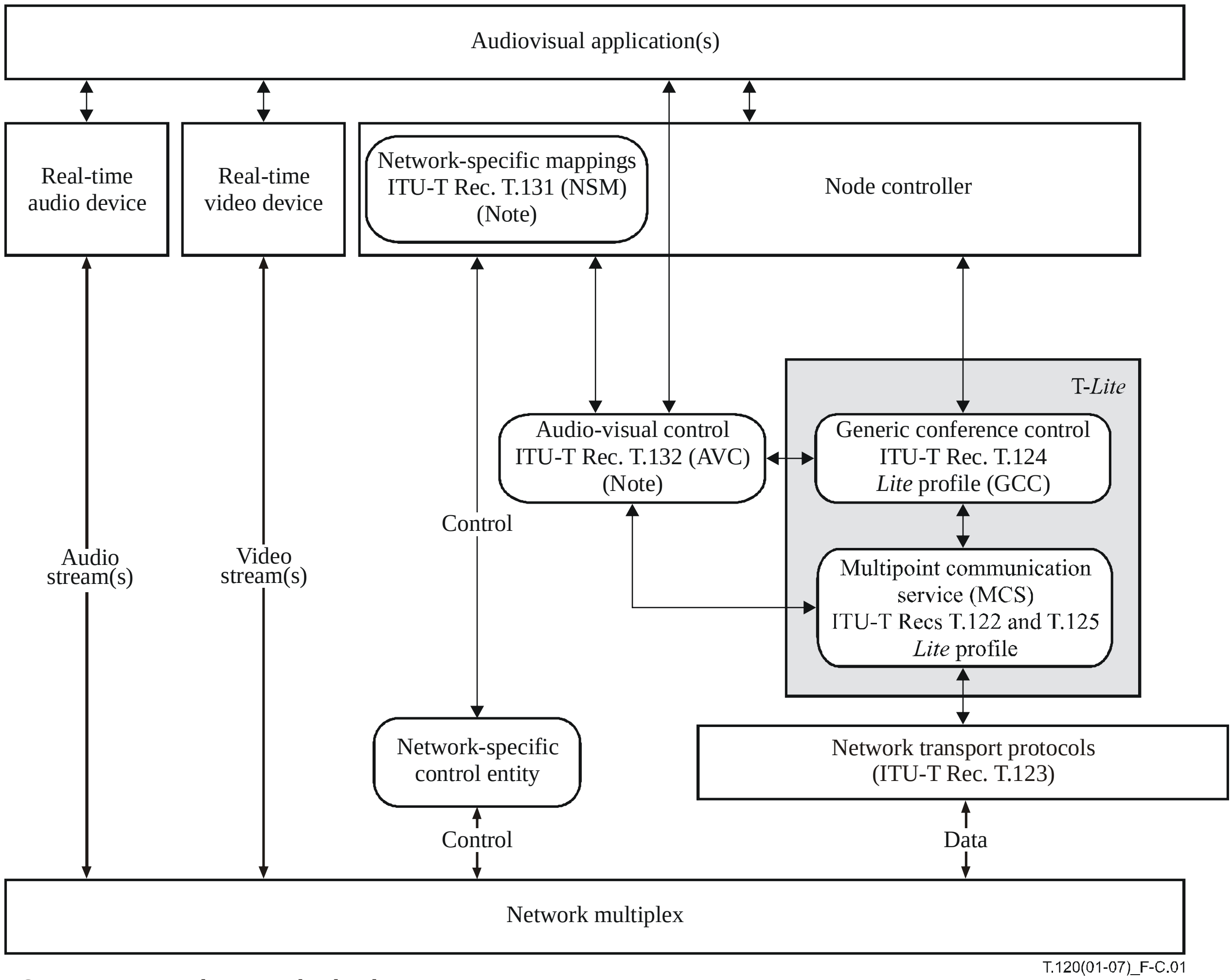
This annex also addresses the issues that may arise when two (or more) T.120 implementations with different or identical feature sets attempt to communicate. This includes discussions on how to treat incoming T.120 PDUs not included in this thin profile. It also addresses backward compatibility issues. The services included in the profiles defined below are the minimal set required to maintain T.120 compliance, while also providing a reasonable subset of functionality. These profiles can be thought of as the least-common-denominator feature set that is necessary to maintain interoperability with all standard compliant flavours of T.120 nodes.

Figure C.1 gives an overview of the components of the T.120 architecture that are retained for the T-*Lite* profile (within the framework of a typical AVD product). Implementing the supported functionality in each of the components of the T-*Lite* profile is considerably simplified to that of a traditional T.120 node because a terminal using this thin infrastructure may be restricted in its operation in the following ways:

1) supports only a single connection to another node;
2) participation in a single conference at a time only (i.e., one MCS connection, one MCS domain, a single transport protocol hierarchy active at a time, one GCC conference, etc.);
3) use of only a single application protocol entity; and

4) Top Provider operation restricted to point-to-point conferences between two nodes which then requires reduced Top Provider functionality.

In some cases, it may be known in advance that a T-*Lite* implementation will only be used as a leaf-node and will never be responsible for providing Top Provider services. In these cases, developers may decide to not include Top Provider functionality in their implementation. This is an acceptable approach.



NOTE – Recommendations under development.

Figure C.1 – T-*Lite* within a typical AVD framework

C.1 *Lite* T.123 profile

This clause only addresses the fact that an implementer is free to adjust the Q.922 protocol parameters suggested by clause 10 of [ITU-T 123] to meet the resource requirements of a particular implementation. For example, when the terminal implementer knows *a priori* that the application(s) to be used are control oriented (e.g., T.123), as compared to data transfer intensive application(s) (e.g., T.128), the Q.922 parameters can be negotiated to reduced values.

In particular, traditional narrow-band audiovisual terminals that only implement T.120 services to host T.123 can take advantage of this knowledge and reduce the Q.922 protocol parameters.

C.2 *Lite* T.122/T.125 profile

This clause outlines the service primitives required from [ITU-T T.122] to realize the above T-*Lite* scenario. In essence, MCS functionality can be reduced to domain management services, services to handle static and assigned MCS channels, and data transmission.

C.2.1 Required primitives and PDUs

In the following table, taken from [ITU-T T.125], all service primitives to be provided by a T-*Lite* node are shaded in gray. The legend below gives a description of each possible action that can be associated with all the possible MCS PDUs. These actions are listed in the right-most column of Table C.1:

- R (Receive) – Implies that the PDU should be received and processed as defined by [ITU-T T.125].
- T (Transmit) – Implies that the PDU should be transmitted as defined by [ITU-T T.125].
- R-NA (Receive – Not Applicable) – Implies that the PDU should never be received. This is for one of three reasons: domain merge is not supported; no request was sent (so the confirm is invalid); or the neighbouring provider cannot own the resource to which the PDU refers. Reception of a PDU of this type is the equivalent of an assertion failure.
- T-NA (Transmit – Not Applicable) – Implies that since the primitive is not supported, this PDU will never be sent.
- R-D (Receive – Default) – Implies that when this PDU is received, the node should follow some default behaviour to insure interoperability.
- T-D (Transmit – Default) – Implies that this PDU should be transmitted based on some default behaviour.
- R-I (Receive – Ignore) – Implies that this PDU can be ignored on reception. Reception of a PDU of this type is NOT an assertion failure.

Table C.1 – MCS primitives and PDUs

Functional unit	Primitives	Associated MCSPDUs	Required actions
Domain management	MCS-CONNECT-PROVIDER request	ConnectInitial	T
	MCS-CONNECT-PROVIDER indication	ConnectInitial	R
	MCS-CONNECT-PROVIDER response	ConnectResponse	T
	MCS-CONNECT-PROVIDER confirm	ConnectResponse	R
		ConnectAdditional	T/R
		ConnectResult	T/R
		PlumbDomainIndication	T/R
		ErectDomainRequest	T/R
		MergeChannelsRequest	T-NA/R-NA
		MergeChannelsConfirm	T-NA/R-NA
		PurgeChannelsIndication	T-NA/R-NA
		MergeTokensRequest	T-NA/R-NA
		MergeTokensConfirm	T-NA/R-NA
		PurgeTokensIndication	T-NA/R-NA
	MCS-DISCONNECT-PROVIDER request	DisconnectProviderUltimatum	T
	MCS-DISCONNECT-PROVIDER indication	DisconnectProviderUltimatum	R
		RejectUltimatum	T/R
	MCS-ATTACH-USER request	AttachUserRequest	T/R
	MCS-ATTACH-USER confirm	AttachUserConfirm	T/R
	MCS-DETACH-USER request	DetachUserRequest	T/R
	MCS-DETACH-USER indication	DetachUserIndication	T/R
		MergeChannelsConfirm	T-NA/R-NA
		PurgeChannelsIndication	T-NA/R-NA
		MergeTokensConfirm	T-NA/R-NA
		PurgeTokensIndication	T-NA/R-NA

Functional unit	Primitives	Associated MCSPDUs	Required actions
Channel management	MCS-CHANNEL-JOIN request MCS-CHANNEL-JOIN confirm	ChannelJoinRequest ChannelJoinConfirm	T/R T/R
	MCS-CHANNEL-LEAVE request MCS-CHANNEL-LEAVE indication	ChannelLeaveRequest MergeChannelsConfirm PurgeChannelsIndication	T/R T-NA/R-NA T-NA/R-NA
	MCS-CHANNEL-CONVENE request MCS-CHANNEL-CONVENE confirm	ChannelConveneRequest ChannelConveneConfirm	T-NA/R-D T-D/R-NA
	MCS-CHANNEL-DISBAND request MCS-CHANNEL-DISBAND indication	ChannelDisbandRequest MergeChannelsConfirm PurgeChannelsIndication	T-NA/R-NA T-NA/R-NA T-NA/R-NA
	MCS-CHANNEL-ADMIT request MCS-CHANNEL-ADMIT indication	ChannelAdmitRequest ChannelAdmitIndication	T-NA/R-NA T-NA/R-I
	MCS-CHANNEL-EXPEL request MCS-CHANNEL-EXPEL indication	ChannelExpelRequest ChannelExpelIndication ChannelDisbandIndication MergeChannelsConfirm PurgeChannelsIndication	T-NA/R-NA T-NA/R-I T-NA/R-I T-NA/R-NA T-NA/R-NA
Data transfer	MCS-SEND-DATA request MCS-SEND-DATA indication	SendDataRequest SendDataIndication	T/R T/R
	MCS-UNIFORM-SEND-DATA request MCS-UNIFORM-SEND-DATA indication	UniformSendDataRequest UniformSendDataIndication	T/R T/R
Token management	MCS-TOKEN-GRAB request MCS-TOKEN-GRAB confirm	TokenGrabRequest TokenGrabConfirm	T-NA/R-D T-D/R-NA
	MCS-TOKEN-INHIBIT request MCS-TOKEN-INHIBIT confirm	TokenInhibitRequest TokenInhibitConfirm	T-NA/R-D T-D/R-NA
	MCS-TOKEN-GIVE request MCS-TOKEN-GIVE indication	TokenGiveRequest TokenGiveIndication	T-NA/R-NA T-NA/R-D

Functional unit	Primitives	Associated MCSPDUs	Required actions
	MCS-TOKEN-GIVE response MCS-TOKEN-GIVE confirm	TokenGiveResponse TokenGiveConfirm	T-D/R-NA T-NA/R-NA
	MCS-TOKEN-PLEASE request MCS-TOKEN-PLEASE indication	TokenPleaseRequest TokenPleaseIndication	T-NA/R-I T-NA/R-NA
	MCS-TOKEN-RELEASE request MCS-TOKEN-RELEASE confirm	TokenReleaseRequest TokenReleaseConfirm	T-NA/R-NA T-NA/R-NA
	MCS-TOKEN-TEST request MCS-TOKEN-TEST confirm	TokenTestRequest TokenTestConfirm	T-NA/R-D T-D/R-NA

C.2.2 Handling unsupported PDUs and primitives

This clause discusses what to do with PDUs that are not simply handled normally (T/R). These PDUs fall into one of the following three categories:

- The PDU is simply ignored on reception.
- The PDU is not applicable under this profile.
- The PDU has a default behaviour associated with it.

PDUs that are ignored require no default behaviour when received. A PDU of this type may be received at any time and must at least be decoded by the receiving node before it is discarded (to ensure the PDU type).

If a PDU is listed as NA or not applicable, it means that the PDU should never be sent or received (depending on the PDU type) from a node supporting this profile. The reception of a PDU that is NA should probably be flagged as an error or assertion because PDUs of this type should never be received. This is an indication that the remote node is exhibiting non-standard behaviour.

The third class of PDUs requires some default behaviour when received or sent. Typically, it is a rejection to a message or service that is not supported.

Table C.2 below is a definitive list of all the PDUs that require default behaviour (along with a description of that behaviour). Note that reception of any PDU that initiates some predefined default behaviour should require no action at the service layer interface.

Table C.2 – Table describing PDUs with default behaviour

PDUs requiring default behaviour	Associated default behaviour
ChannelConveneRequest – Rx	If the receiving node is the Top Provider, the reception of a ChannelConveneRequest should result in the transmission of the ChannelConveneConfirm described next.
ChannelConveneConfirm – Tx	On reception of a ChannelConveneRequest, a ChannelConveneConfirm should be generated which includes the following: the initiator ID from the request, and a result of "too many channels".
TokenGrabRequest – Rx	If the receiving node is the Top Provider, the reception of a TokenGrabRequest should result in the transmission of the TokenGrabConfirm described next.
TokenGrabConfirm – Tx	On reception of a TokenGrabRequest, a TokenGrabConfirm should be generated which includes the following: the initiator ID and token ID from the request, a result of "too many tokens", and a token status of "not in use".
TokenInhibitRequest – Rx	If the receiving node is the Top Provider, the reception of a TokenInhibitRequest should result in the transmission of the TokenInhibitConfirm described next.
TokenInhibitConfirm – Tx	On reception of a TokenInhibitRequest, a TokenInhibitConfirm should be generated which includes the following: the initiator ID and token ID from the request, a result of "too many tokens", and a token status of "not in use".
TokenGiveIndication – Rx	If the receiving node is a leaf node, then the reception of a TokenGiveIndication should result in the transmission of the TokenGiveResponse described next.

Table C.2 – Table describing PDUs with default behaviour

PDUs requiring default behaviour	Associated default behaviour
TokenGiveResponse – Tx	On reception of a TokenGiveIndication, a TokenGiveResponse should be generated which includes the following: the recipient ID and token ID from the indication, and a result of "user rejected".
TokenTestRequest – Rx	If the receiving node is the Top Provider, the reception of a TokenTestRequest should result in the transmission of the TokenTestConfirm described next.
TokenTestConfirm – Tx	On reception of a TokenTestRequest, a TokenTestConfirm should be generated which includes the following: the initiator ID and token ID from the request, and a token status of "not in use".

C.2.3 Handling supported PDUs and primitives

All T.125 PDUs that are supported by the T-*Lite* profile should be handled as specified by [ITU-T T.125].

When a T-*Lite* node is required to act as Top Provider, it should establish the Domain Parameters to indicate that the maximum number of tokens is zero (0). This reflects the fact that all token-related requests will be denied.

C.2.4 Negotiating Top-Provider responsibilities

A T-*Lite* node may be required to act as the Top MCS Provider. The connection setup for a T-*Lite* node should attempt to hand off Top Provider responsibilities whenever possible (the only exception to this case is when two T-*Lite* nodes are attempting to connect). This ensures the lowest probability that a thin Top MCS Provider would receive an MCSPDU other than those indicated above. The hand-off procedure will be handled by the GCC Conference Query and is discussed below.

C.3 *Lite* T.124 profile

This clause describes the service primitives required from a T.124 provider for a T-*Lite* node. For this portion of the T-*Lite* profile, only a subset of the GCC services are needed. These include conference establishment and termination, support for the conference roster, and support for a single application roster (which does not reduce the number of services but simplifies internal management). Neither the application registry nor the conference conductorship services are needed. Also, no miscellaneous functions are required other than the sending of text messages.

C.3.1 Required primitives and PDUs

In the following table (Table C.3) taken from [ITU-T T.124], all service primitives to be provided by a T-*Lite* node are shaded in gray. For each primitive, it is noted whether or not it applies only to a Top Provider node (TP stands for Top Provider). For nodes that are guaranteed to never be a Top Provider node, it is not necessary to support the primitives specified as TP. T-*Lite* nodes that can act as a Top Provider should support all the gray-shaded primitives.

The legend below gives a description of each possible action that can be associated with all the possible GCC Primitives and PDUs. These actions are listed in the right-most column of Table C.3:

- R (Receive) – Implies that the PDU should be received and processed as defined by [ITU-T T.124].
- T (Transmit) – Implies that the PDU should be transmitted as defined by [ITU-T T.124].
- R-NA (Receive – Not Applicable) – Implies that since the request portion of this exchange will never be sent, a response should never occur (and can be ignored). Reception of a PDU of this type is the equivalent of an assertion failure.
- T-NA (Transmit – Not Applicable) – Implies that since the primitive is not supported, this PDU will never be sent.
- R-D (Receive – Default) – Implies that when this PDU is received, the node should follow some default behaviour to insure interoperability.
- T-D (Transmit – Default) – Implies that this PDU should be transmitted based on some default behaviour.
- R-I (Receive – Ignore) – Implies that this PDU can be ignored on reception. Reception of a PDU of this type is NOT an assertion failure.

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
Conference establishment and termination	GCC-CONFERENCE-CREATE request		ConferenceCreateRequest	T
	GCC-CONFERENCE-CREATE indication	TP	ConferenceCreateRequest	R
	GCC-CONFERENCE-CREATE response	TP	ConferenceCreateResponse	T
	GCC-CONFERENCE-CREATE confirm		ConferenceCreateResponse	R
			UserIDIndication	T, R
	GCC-CONFERENCE-QUERY request		ConferenceQueryRequest	T
	GCC-CONFERENCE-QUERY indication		ConferenceQueryRequest	R
	GCC-CONFERENCE-QUERY response		ConferenceQueryResponse	T
	GCC-CONFERENCE-QUERY confirm		ConferenceQueryResponse	R
	GCC-CONFERENCE-JOIN request		ConferenceJoinRequest	T
	GCC-CONFERENCE-JOIN indication	TP	ConferenceJoinRequest	R
	GCC-CONFERENCE-JOIN response	TP	ConferenceJoinResponse	T
	GCC-CONFERENCE-JOIN confirm		ConferenceJoinResponse	R
			UserIDIndication	T, R
	GCC-CONFERENCE-INVITE request		ConferenceInviteRequest	T-NA
	GCC-CONFERENCE-INVITE indication		ConferenceInviteRequest	R
	GCC-CONFERENCE-INVITE response		ConferenceInviteResponse	T
	GCC-CONFERENCE-INVITE confirm		ConferenceInviteResponse	R-NA
			UserIDIndication	T, R-NA

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-CONFERENCE-ADD request		ConferenceAddRequest	T-NA
	GCC-CONFERENCE-ADD indication		ConferenceAddRequest	R-D
	GCC-CONFERENCE-ADD response		ConferenceAddResponse	T-D
	GCC-CONFERENCE-ADD confirm		ConferenceAddResponse	R-NA
	GCC-CONFERENCE-LOCK request		ConferenceLockRequest	T-NA
	GCC-CONFERENCE-LOCK indication		ConferenceLockRequest	R-D
	GCC-CONFERENCE-LOCK response		ConferenceLockResponse	T-D
	GCC-CONFERENCE-LOCK confirm		ConferenceLockResponse	R-NA
	GCC-CONFERENCE-UNLOCK request		ConferenceUnlockRequest	T-NA
	GCC-CONFERENCE-UNLOCK indication		ConferenceUnlockRequest	R-D
	GCC-CONFERENCE-UNLOCK response		ConferenceUnlockResponse	T-D
	GCC-CONFERENCE-UNLOCK confirm		ConferenceUnlockResponse	R-NA
	GCC-CONFERENCE-LOCK-REPORT indication		ConferenceLockIndication ConferenceUnlockIndication	T-NA, R-I T-NA, R-I
	GCC-CONFERENCE-DISCONNECT request		–	–
	GCC-CONFERENCE-DISCONNECT indication		–	–
	GCC-CONFERENCE-DISCONNECT confirm		–	–
	GCC-CONFERENCE-TERMINATE request		ConferenceTerminateRequest ConferenceTerminateRequest	T-NA R
	GCC-CONFERENCE-TERMINATE indication		ConferenceTerminateIndication ConferenceTerminateIndication	T R
	GCC-CONFERENCE-TERMINATE confirm		ConferenceTerminateResponse ConferenceTerminateResponse	T R-NA
	GCC-CONFERENCE-EJECT-USER request		ConferenceEjectUserRequest ConferenceEjectUserRequest	T-NA R-D

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-CONFERENCE-EJECT-USER indication		ConferenceEjectUserIndication	T-NA
			ConferenceEjectUserIndication	R
	GCC-CONFERENCE-EJECT-USER confirm		ConferenceEjectUserResponse	T-D
			ConferenceEjectUserResponse	R-NA
	GCC-CONFERENCE-TRANSFER request		ConferenceTransferRequest	T-NA
			ConferenceTransferRequest	R-D
	GCC-CONFERENCE-TRANSFER indication		ConferenceTransferIndication	T-NA
			ConferenceTransferIndication	R-I
	GCC-CONFERENCE-TRANSFER confirm		ConferenceTransferResponse	T-D
			ConferenceTransferResponse	R-NA
Conference roster	GCC-CONFERENCE-ANNOUNCE-PRESENCE request		RosterUpdateIndication	T, R
			RosterRefreshRequest	T, R
	GCC-CONFERENCE-ANNOUNCE-PRESENCE confirm		–	–
	GCC-CONFERENCE-ROSTER-REPORT indication		RosterUpdateIndication	T, R
	GCC-CONFERENCE-ROSTER-INQUIRE request		–	–
	GCC-CONFERENCE-ROSTER-INQUIRE confirm		–	–
Application roster	GCC-APPLICATION-PERMISSION-TO-ENROLL indication		–	–
	GCC-APPLICATION-ENROLL request		RosterUpdateIndication	T, R
			RosterRefreshRequest	T, R
	GCC-APPLICATION-ENROLL confirm		–	–
	GCC-APPLICATION-ROSTER-REPORT indication		RosterUpdateIndication	T, R
	GCC-APPLICATION-ROSTER-INQUIRE request		–	–
	GCC-APPLICATION-ROSTER-INQUIRE confirm		–	–
	GCC-APPLICATION-INVOKE request		ApplicationInvokeIndication	T-NA

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-APPLICATION-INVOKE indication		ApplicationInvokeIndication	R-I
	GCC-APPLICATION-INVOKE confirm		–	–
Application registry	GCC-REGISTRY-REGISTER-CHANNEL request		RegistryRegisterChannelRequest	T-NA
			RegistryRegisterChannelRequest	R-D
	GCC-REGISTRY-REGISTER-CHANNEL confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-ASSIGN-TOKEN request		RegistryAssignTokenRequest	T-NA
			RegistryAssignTokenRequest	R-D
	GCC-REGISTRY-ASSIGN-TOKEN confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-SET-PARAMETER request		RegistrySetParameterRequest	T-NA
			RegistrySetParameterRequest	R-D
	GCC-REGISTRY-SET-PARAMETER confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-RETRIEVE-ENTRY request		RegistryRetrieveEntryRequest	T-NA
			RegistryRetrieveEntryRequest	R-D
	GCC-REGISTRY-RETRIEVE-ENTRY confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-DELETE-ENTRY request		RegistryDeleteEntryRequest	T-NA
			RegistryDeleteEntryRequest	R-D
	GCC-REGISTRY-DELETE-ENTRY confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-MONITOR request		RegistryMonitorEntryRequest	T-NA
			RegistryMonitorEntryRequest	R-D

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-REGISTRY-MONITOR indication		RegistryMonitorEntryIndication	T-NA
			RegistryMonitorEntryIndication	R-I
	GCC-REGISTRY-MONITOR confirm		RegistryResponse	T-D
			RegistryResponse	R-NA
	GCC-REGISTRY-ALLOCATE-HANDLE request		RegistryAllocateHandleRequest	T-NA
			RegistryAllocateHandleRequest	R-D
	GCC-REGISTRY-ALLOCATE-HANDLE confirm		RegistryAllocateHandleResponse	T-D
			RegistryAllocateHandleResponse	R-NA
Conference conductorship	GCC-CONDUCTOR-ASSIGN request		–	–
	GCC-CONDUCTOR-ASSIGN indication		ConductorAssignIndication	T-NA
			ConductorAssignIndication	R-I
	GCC-CONDUCTOR-ASSIGN confirm		–	–
	GCC-CONDUCTOR-RELEASE request		ConductorReleaseIndication	T-NA
			ConductorReleaseIndication	R-I
	GCC-CONDUCTOR-RELEASE indication		ConductorReleaseIndication	T-NA
			ConductorReleaseIndication	R-I
	GCC-CONDUCTOR-RELEASE confirm		–	–
	GCC-CONDUCTOR-PLEASE request		–	–
	GCC-CONDUCTOR-PLEASE indication		–	–
	GCC-CONDUCTOR-PLEASE response		–	–
	GCC-CONDUCTOR-PLEASE confirm		–	–
	GCC-CONDUCTOR-GIVE request		–	–
	GCC-CONDUCTOR-GIVE indication		–	–
	GCC-CONDUCTOR-GIVE response		ConductorAssignIndication	T-NA

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-CONDUCTOR-GIVE confirm		–	–
	GCC-CONDUCTOR-INQUIRE request		–	–
	GCC-CONDUCTOR-INQUIRE confirm		–	–
	GCC-CONDUCTOR-PERMISSION-ASK request		ConductorPermissionAskIndication	T-NA
	GCC-CONDUCTOR-PERMISSION-ASK indication		ConductorPermissionAskIndication	R-I
	GCC-CONDUCTOR-PERMISSION-ASK confirm		–	–
	GCC-CONDUCTOR-PERMISSION-GRANT request		ConductorPermissionGrantIndication	T-NA
	GCC-CONDUCTOR-PERMISSION-GRANT indication		ConductorPermissionGrantIndication	R-I
	GCC-CONDUCTOR-PERMISSION-GRANT confirm		–	–
Miscellaneous functions	GCC-CONFERENCE-TIME-REMAINING request		ConferenceTimeRemainingIndication	T-NA
	GCC-CONFERENCE-TIME-REMAINING indication		ConferenceTimeRemainingIndication	R-I
	GCC-CONFERENCE-TIME-REMAINING confirm		–	–
	GCC-CONFERENCE-TIME-INQUIRE request		ConferenceTimeInquireIndication	T-NA
	GCC-CONFERENCE-TIME-INQUIRE indication		ConferenceTimeInquireIndication	R-I
	GCC-CONFERENCE-TIME-INQUIRE confirm		–	–
	GCC-CONFERENCE-EXTEND request		ConferenceTimeExtendIndication	T-NA
	GCC-CONFERENCE-EXTEND indication		ConferenceTimeExtendIndication	R-I
	GCC-CONFERENCE-EXTEND confirm		–	–
	GCC-CONFERENCE-ASSISTANCE request		ConferenceAssistanceIndication	T-NA
	GCC-CONFERENCE-ASSISTANCE indication		ConferenceAssistanceIndication	R-I
	GCC-CONFERENCE-ASSISTANCE confirm		–	–

Table C.3 – GCC primitives and PDUs

Functional unit	Primitives	TP	Associated PDUs	Actions
	GCC-TEXT-MESSAGE request	O	TextMessageIndication	T
	GCC-TEXT-MESSAGE indication	O	TextMessageIndication	R
	GCC-TEXT-MESSAGE confirm	O	–	–
	–	–	FunctionNotSupported	T, R
M Mandatory C Conditionally Mandatory O Optional				

C.3.2 Handling unsupported PDUs and primitives

This clause discusses what to do with PDUs that are not simply handled normally (T/R). These PDUs fall into one of the following three categories:

- The PDU is simply ignored on reception.
- The PDU is not applicable under this profile.
- The PDU has a default behaviour associated with it.

PDUs that are ignored require no default behaviour when received. A PDU of this type may be received at any time and must at least be decoded by the receiving node before it is discarded (to ensure the PDU type).

If a PDU is listed as NA or not applicable, it means that the PDU should never be sent or received (depending on the PDU type) from a node supporting this profile. The reception of a PDU that is NA should probably be flagged as an error or assertion because PDUs of this type are only received due to an associated request that is NOT supported by this profile. Therefore, if a not-applicable PDU is received, it means that either the receiving node initiated a request that it should not have or another node in the conference is exhibiting non-standard behaviour.

The third class of PDU requires some default behaviour when received or sent. Typically, it is a rejection to a message or service that is not supported. In many cases, it will appear to the requesting node that it is not a valid requester. This can often be attributed to lack of a particular privilege.

Table C.4 below is a definitive list of all the PDUs of this type along with their associated default behaviour. Note that reception of any PDU that initiates some predefined default behaviour should require no action at the service layer interface.

Table C.4 – Table describing PDUs with default behaviour

PDUs requiring default behaviour	Associated default behaviour
ConferenceAddRequest – Rx	On reception of an add-request, the add-response described below should be generated.
ConferenceAddResponse – Tx	On reception of an add-request, an add-response should be generated which includes the following: the tag sent in the add request, a result of connectionUnsuccessful and no User Data.
ConferenceLockRequest – Rx	On reception of a lock-request, the lock-response described below should be generated.
ConferenceLockResponse – Tx	On reception of a lock-request, a lock-response should be generated which includes the following: a result of invalidRequester.
ConferenceUnlockRequest – Rx	On reception of an unlock-request, the unlock-response described below should be generated.
ConferenceUnlockResponse – Tx	On reception of an unlock-request, an unlock-response should be generated which includes the following: a result of invalidRequester.
ConferenceEjectUserRequest – Rx	On reception of an eject-user-request, the eject-user-response described below should be generated.
ConferenceEjectUserResponse – Tx	On reception of an eject-user-request, an eject-user-response should be generated which includes the following: the nodeToEject UserID included in the request and a result of invalidRequester.

Table C.4 – Table describing PDUs with default behaviour

PDUs requiring default behaviour	Associated default behaviour
ConferenceTransferRequest – Rx	On reception of a transfer-request, the transfer-response described below should be generated.
ConferenceTransferResponse – Tx	On reception of a transfer-request, a transfer-response should be generated which includes the following: the conference name and modifier included in the request, the transferring node list included in the request and a result of invalidRequester.
RegistryRegisterChannelRequest – Rx	On reception of a register-channel-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of a register-channel-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to registerChannel, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of registryFull.
RegistryAssignTokenRequest – Rx	On reception of an assign-token-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of an assign-token-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to assignToken, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of registryFull.
RegistrySetParameterRequest – Rx	On reception of a set-parameter-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of a set-parameter-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to setParameter, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of registryFull.
RegistryRetrieveEntryRequest – Rx	On reception of a retrieve-entry-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of a retrieve-entry-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to retrieveEntry, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of notFound.
RegistryDeleteEntryRequest – Rx	On reception of a delete-entry-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of a delete-entry-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to deleteEntry, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of notFound.

Table C.4 – Table describing PDUs with default behaviour

PDUs requiring default behaviour	Associated default behaviour
RegistryMonitorEntryRequest – Rx	On reception of a monitor-entry-request, the registry-response described below should be generated.
RegistryResponse – Tx	On reception of a monitor-entry-request, a registry-response should be generated which includes the following: the entityID of the requesting node, the primitiveType set to monitorEntry, the registryKey set to the key included in the request, the registryItem set to vacant, the registry entry owner set to notOwned, no modification-rights and a result of notFound.
RegistryAllocateHandleRequest – Rx	On reception of an allocate-handle-request, the allocate-handle-response described below should be generated.
RegistryAllocateHandleResponse – Tx	On reception of an allocate-handle-request, an allocate-handle-response should be generated which includes the following: the entityID of the requesting node, the numberOfHandle set to the number of handles specified in the request, the firstHandle set to zero, and a result of noHandlesAvailable.

C.3.3 Handling supported PDUs and primitives

All T.124 PDUs that are supported by the T-*Lite* profile should be handled as specified by [ITU-T T.124]. However, there are a number of parameters that shall be set to a "fixed" or static value in these supported PDUs when implementing the thin T.124 portion of the T-*Lite* profile. These are listed in Table C.5.

Table C.5 – Fixed parameter values for transmitted PDUs

Parameter	Value
ConferenceCreateRequest: lockedConference	Always set to FALSE.
ConferenceCreateRequest: conductibleConference	Always set to FALSE.
ConferenceCreateRequest: conductorPrivileges	Should never include ejectUser, add, lockUnlock or transfer.
ConferenceCreateRequest: conductedPrivileges	Should never include ejectUser, add, lockUnlock or transfer.
ConferenceCreateRequest: nonConductedPrivileges	Should never include ejectUser, add, lockUnlock or transfer.
ConferenceCreateResponse:result	Should always be set to userRejected if conference is already running on the node.
ConferenceQueryRequest: nodeType	Always set to terminal.
ConferenceQueryResponse: nodeType	Always set to terminal.
ConferenceJoinResponse: lockedConference	Always set to FALSE.
ConferenceJoinResponse: conductibleConference	Always set to FALSE.
ConferenceJoinResponse: conductorPrivileges	Should never include ejectUser, add, lockUnlock or transfer.
ConferenceJoinResponse: conductedPrivileges	No need to specify since conductorship is not supported.

Table C.5 – Fixed parameter values for transmitted PDUs

Parameter	Value
ConferenceJoinResponse: nonConductedPrivileges	Should never include ejectUser, add, lockUnlock or transfer.
ConferenceJoinResponse: result	If node is already connected to another node this should always be set to userRejected.
NodeRecord: nodeType	Always set to terminal.
NodeRecord: nodeProperties: managementDevice	Always set to FALSE.
NodeRecord: nodeProperties: peripheralDevice	Always set to TRUE.
ApplicationRecord: conductingOperationCapable	Always set to FALSE.

There is also an issue of how a node should handle a roster update for an APE that it does not support. This could be the case if a T-*Lite* node is participating in a point-to-point conference with a T.120 terminal node which can support multiple simultaneous APE sessions (say T.128, T.127 and T.126). If this should occur, it is up to the T-*Lite* node to process the roster update(s) associated with the APE(s) that it supports and ignore the rest. Note that, according to C.3.5, if the T-*Lite* node is the Top Provider, the unsupported APEs running on the terminal node will never see a roster update which includes itself and will therefore never complete the establishment of its session.

C.3.4 Negotiating Top-Provider responsibilities

As stated above, it is desirable for a T-*Lite* node to avoid Top-Provider responsibilities whenever possible. This prevents a node with minimal functionality from limiting the scope and services supported by the conference. Unfortunately, there is no definitive way to guarantee this. However, by following the rules for determining the default action of the called and calling nodes defined in Table C.6 below, along with the fixed parameters stated above, only one possible situation exists where a T-*Lite* node could assume Top Provider responsibilities in a conference. This is the situation where a T-*Lite* node is connected to another T-*Lite* node. If this situation occurs, the conference will be severely limited but it will still be able to proceed. Some of these limitations include:

- no multipoint support;
- no "miscellaneous" services support;
- no "privilege" services such as eject user, add, transfer or lock/unlock;
- support for only a single APE.

By following the rules in Table C.6 and setting the appropriate fixed parameters, the thin T.124 node has the greatest probability of joining the conference as a leaf or subordinate node. MCU/Terminals and MCUs will always default to the parent node in situations where it is connected to a T-*Lite* node. Note that, as mentioned in the overview, a T-*Lite* node can reduce its size even more if it is guaranteed to never be a Top Provider node (even in a point-to-point connection).

Table C.6 – Rules for determining the default action of the Called and Calling Nodes

Calling Node type	Called Node type	Unlocked Conferences in List	Default Conference Flag	Wait for Invite Flag	No Unlisted Conference Flag	Default action of Calling Node	Default action of Called Node
T-*Lite* terminal	Any terminal	*	*	*	*	Caller creates remotely	Wait for caller

Table C.6 shows that whenever a T-*Lite* node initiates a call, it will always dial "upward" from MCS's perspective. This will guarantee that a fully compliant terminal will always end up as the Top Provider in a point-to-point call. Note that all other cases such as a T-*Lite* node calling an MCU should follow the terminal cases outlined in Table 7-17 of [ITU-T T.124]

C.3.5 Support of a single static session

T-*Lite* nodes that wish to support only a single static session may find it necessary to "filter" incoming session or roster information that is not relevant to the node. This could include roster update indications for non-static APE sessions or roster updates for APEs that are not supported by the T-*Lite* node.

A GCC session is strictly defined by its associated Session Key which includes an Application Protocol Key and a Session ID. The Application Protocol Key is a unique identifier which defines the Application Protocol that created the session. The Session ID is simply the channel in use by that session (which in this case is a static channel). Roster updates associated with a particular session can be easily filtered if the T-*Lite* node knows in advance which static session(s) it must support. As stated above, this information would include the Application Protocol Key associated with the APE and the channel ID for the static session of interest. Any application information imbedded in a roster update indication that does not include the specified Session Key would simply be ignored.

C.4 *Lite* T.121 profile

This clause describes the services required from a T.120 provider implementing the Generic Application Template (T.GAT or T.121) for a T-*Lite* node. For this portion of the T-*Lite* profile, only a subset of the GAT services is needed. These only include services associated with setting up a Static mode session. This is because only static channels and tokens are supported within the T-*Lite* MCS profile and no GCC registry-related functions are supported. Therefore, Dynamic Multicast and Dynamic Private session support can be ignored when building a T-*Lite* node. This greatly simplifies the effort required to support T-*Lite* from the perspective of an APE.

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, 2007