INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

T.120

Annex C
(02/98)

SERIES T: TERMINALS FOR TELEMATIC SERVICES

Data protocols for multimedia conferencing

Annex C: Lightweight profiles for the T.120 architecture

ITU-T Recommendation T.120 – Annex C

(Previously CCITT Recommendation)

ITU-T T-SERIES RECOMMENDATIONS

TERMINALS FOR TELEMATIC SERVICES

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

ITU-T RECOMMENDATION T.120

DATA PROTOCOLS FOR MULTIMEDIA CONFERENCING

ANNEX C

Lightweight profiles for the T.120 architecture

Summary

The intent of this Annex is to provide a lightweight T.120 profile that can be implemented in products requiring minimal data services. This includes pure audiovisual environments wishing to incorporate the services of Application Protocol Entities (APEs) such as T.130 (in contrast to multimedia terminals encompassing audio, video, and data services). It also includes products incorporating thin implementations of other T.120 APEs such as text telephony. This set of profiles will be particularly useful in audiovisual environments where resources are extremely limited.

Source

Annex C to ITU-T Recommendation T.120 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 6th of February 1998.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The 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 Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 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.

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The 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, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

Recommendation T.120

DATA PROTOCOLS FOR MULTIMEDIA CONFERENCING

ANNEX C

Lightweight profiles for the T.120 architecture

(Geneva, 1998)

C.0 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.130), 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.4.5).

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:

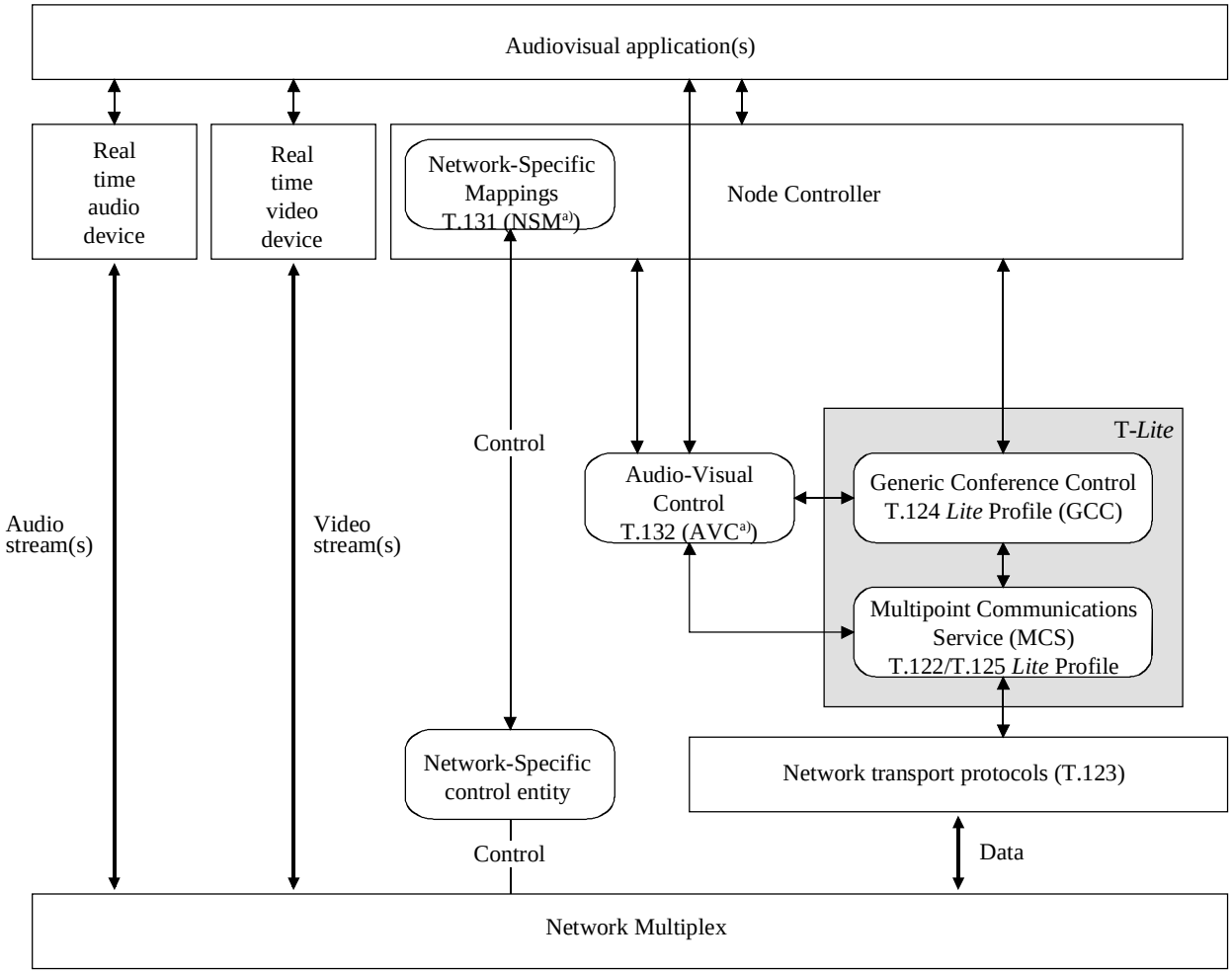
1) supports only a single connection to another node;

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.



[a] Denotes Recommendations under development.

Figure C.1/T.120 – T-*Lite* within a typical AVD framework

C.1 Normative 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; all 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 Recommendation T.120 (1996), *Data protocols for multimedia conferencing*.

– ITU-T Recommendation T.122 (1998), *Multipoint communication service – Service definition*.

– ITU-T Recommendation T.124 (1998), *Generic Conference Control*.

– ITU-T Recommendation T.125 (1998), *Multipoint communication service protocol specification*.

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

C.2 *Lite* T.123 profile

This subclause only addresses the fact that an implementer is free to adjust the Q.922 protocol parameters suggested by clause 10/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.132), 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 audio/visual terminals that only implement T.120 services to host T.132 can take advantage of this knowledge and reduce the Q.922 protocol parameters.

C.3 *Lite* T.122/T.125 profile

This subclause outlines the service primitives required from 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.3.1 Required primitives and PDUs

In the following table, taken from Recommendation 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 Recommendation T.125.
- T (Transmit) – Implies that the PDU should be transmitted as defined by Recommendation 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/T.120 – 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
Channel Management	MCS-CHANNEL-JOIN request MCS-CHANNEL-JOIN confirm	ChannelJoinRequest ChannelJoinConfirm	T/R T/R
	MCS-CHANNEL-LEAVE request	ChannelLeaveRequest	T/R
	MCS-CHANNEL-LEAVE indication	MergeChannelsConfirm	T-NA/R-NA
		PurgeChannelsIndication	T-NA/R-NA

Table C.1/T.120 – MCS primitives and PDUs *(concluded)*

Functional Unit	Primitives	Associated MCSPDUs	Required actions
	MCS-CHANNEL-CONVENE request	ChannelConveneRequest	T-NA/R-D
	MCS-CHANNEL-CONVENE confirm	ChannelConveneConfirm	T-D/R-NA
	MCS-CHANNEL-DISBAND request	ChannelDisbandRequest	T-NA/R-NA
	MCS-CHANNEL-DISBAND indication	MergeChannelsConfirm	T-NA/R-NA
		PurgeChannelsIndication	T-NA/R-NA
	MCS-CHANNEL-ADMIT request	ChannelAdmitRequest	T-NA/R-NA
	MCS-CHANNEL-ADMIT indication	ChannelAdmitIndication	T-NA/R-I
	MCS-CHANNEL-EXPEL request	ChannelExpelRequest	T-NA/R-NA
	MCS-CHANNEL-EXPEL indication	ChannelExpelIndication	T-NA/R-I
		ChannelDisbandIndication	T-NA/R-I
		MergeChannelsConfirm	T-NA/R-NA
		PurgeChannelsIndication	T-NA/R-NA
Data Transfer	MCS-SEND-DATA request	SendDataRequest	T/R
	MCS-SEND-DATA indication	SendDataIndication	T/R
	MCS-UNIFORM-SEND-DATA request	UniformSendDataRequest	T/R
	MCS-UNIFORM-SEND-DATA indication	UniformSendDataIndication	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	TokenInhibitRequest	T-NA/R-D
	MCS-TOKEN-INHIBIT confirm	TokenInhibitConfirm	T-D/R-NA
	MCS-TOKEN-GIVE request	TokenGiveRequest	T-NA/R-NA
	MCS-TOKEN-GIVE indication	TokenGiveIndication	T-NA/R-D
	MCS-TOKEN-GIVE response	TokenGiveResponse	T-D/R-NA
	MCS-TOKEN-GIVE confirm	TokenGiveConfirm	T-NA/R-NA
	MCS-TOKEN-PLEASE request	TokenPleaseRequest	T-NA/R-I
	MCS-TOKEN-PLEASE indication	TokenPleaseIndication	T-NA/R-NA
	MCS-TOKEN-RELEASE request	TokenReleaseRequest	T-NA/R-NA
	MCS-TOKEN-RELEASE confirm	TokenReleaseConfirm	T-NA/R-NA
	MCS-TOKEN-TEST request	TokenTestRequest	T-NA/R-D
	MCS-TOKEN-TEST confirm	TokenTestConfirm	T-D/R-NA

C.3.2 Handling unsupported PDUs and primitives

This subclause 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 PDU 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/T.120 – 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/T.120 – Table describing PDUs with default behaviour *(concluded)*

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.3.3 Handling supported PDUs and primitives

All T.125 PDUs that are supported by the T-*Lite* profile should be handled as specified by Recommendation 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.3.4 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.4 *Lite* T.124 profile

This subclause 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 or the conference conductorship services are needed. Also, no miscellaneous functions are required other than the sending of text messages.

C.4.1 Required primitives and PDUs

In the following table taken from Recommendation 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 Recommendation T.124.

- T (Transmit) – Implies that the PDU should be transmitted as defined by Recommendation 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/T.120 – 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

Table C.3/T.120 – GCC primitives and PDUs *(continued)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
	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
	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	T-NA, R-I
			ConferenceUnlockIndication	T-NA, R-I
	GCC-CONFERENCE-DISCONNECT request		–	–
	GCC-CONFERENCE-DISCONNECT indication		–	–
	GCC-CONFERENCE-DISCONNECT confirm		–	–

Table C.3/T.120 – GCC primitives and PDUs *(continued)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
	GCC-CONFERENCE-TERMINATE request		ConferenceTerminateRequest	T-NA
			ConferenceTerminateRequest	R
	GCC-CONFERENCE-TERMINATE indication		ConferenceTerminateIndication	T
			ConferenceTerminateIndication	R
	GCC-CONFERENCE-TERMINATE confirm		ConferenceTerminateResponse	T
			ConferenceTerminateResponse	R-NA
	GCC-CONFERENCE-EJECT-USER request		ConferenceEjectUserRequest	T-NA
			ConferenceEjectUserRequest	R-D
	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		–	–

Table C.3/T.120 – GCC primitives and PDUs *(continued)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
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
	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

Table C.3/T.120 – GCC primitives and PDUs *(continued)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
	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
	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		–	–

Table C.3/T.120 – GCC primitives and PDUs *(continued)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
	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
	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		–	–

Table C.3/T.120 – GCC primitives and PDUs *(concluded)*

Functional Unit	Primitives	TP	Associated PDUs	Actions
	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		–	–
	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.4.2 Handling unsupported PDUs and primitives

This subclause 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 shouldn't 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/T.120 – 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.
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.

Table C.4/T.120 – Table describing PDUs with default behaviour *(continued)*

PDUs requiring default behaviour	Associated default behaviour
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.
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.

Table C.4/T.120 – Table describing PDUs with default behaviour *(concluded)*

PDUs requiring default behaviour	Associated default behaviour
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.4.3 Handling supported PDUs and primitives

All T.124 PDUs that are supported by the T-*Lite* profile should be handled as specified by Recommendation 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/T.120 – 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.
ConferenceJoinResponse: nonConductedPrivileges	Should never include ejectUser, add, lockUnlock or transfer.

Table C.5/T.120 – Fixed parameter values for transmitted PDUs *(concluded)*

Parameter	Value
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.130, T.128 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.4.5, 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.4.4 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 we 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 below 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/T.120 – 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/T.124.

C.4.5 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.5 *Lite* T.121 profile

This subclause 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 are 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.

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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	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	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
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 and open system communications
Series Y	Global information infrastructure
Series Z	Programming languages