



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.450-2

(02/98)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS
Supplementary services for multimedia

Call transfer supplementary service for H.323

ITU-T Recommendation H.450-2

(Previously CCITT Recommendation)

ITU-T H-SERIES RECOMMENDATIONS
AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Characteristics of transmission channels used for other than telephone purposes	H.10–H.19
Use of telephone-type circuits for voice-frequency telegraphy	H.20–H.29
Telephone circuits or cables used for various types of telegraph transmission or simultaneous transmission	H.30–H.39
Telephone-type circuits used for facsimile telegraphy	H.40–H.49
Characteristics of data signals	H.50–H.99
CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	H.280–H.299
Systems and terminal equipment for audiovisual services	H.300–H.399
Supplementary services for multimedia	H.450–H.499

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

ITU-T RECOMMENDATION H.450.2

CALL TRANSFER SUPPLEMENTARY SERVICE FOR H.323

Summary

This Recommendation describes the procedures and the signalling protocol for the Call Transfer supplementary service (SS-CT) in H.323 (Packet-based multimedia communications systems) networks.

SS-CT is a supplementary service which enables the served user A to transform an existing call (user A-user B) into a new call between user B and a user C selected by user A. User A may or may not have a call established with user C prior to Call Transfer.

This Recommendation makes use of the "Generic functional protocol for the support of supplementary services in H.323" as defined in Recommendation H.450.1

The procedures and the signalling protocol of this Recommendation are derived from the Call Transfer supplementary service specified in ISO/IEC 13865 and 13869.

Source

ITU-T Recommendation H.450.2 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.

© ITU 1998

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	Page
1	Scope 1
2	References 1
3	Terms and definitions 1
4	Abbreviations and acronyms 2
5	Description 2
6	Messages and information elements 3
7	Actions at the transferring endpoint (user A) 3
7.1	Normal procedures if only the primary call exists 3
7.2	Normal procedures if also the secondary call exists 4
7.3	Exceptional procedures 4
8	Actions at the transferred endpoint (user B) 4
8.1	Normal procedures 4
8.2	Exceptional procedures 5
9	Actions at the transferred-to endpoint (user C) 5
9.1	Normal procedures 5
9.2	Exceptional procedures 5
10	Additional procedures 6
10.1	Interworking with non-H.323 terminals 6
10.2	Hold and consultation 6
10.3	Transfer without rerouting (i.e. transfer by join) 6
10.4	Further actions at transferred and transferred-to endpoints 6
10.5	Interactions with H.323 Ad Hoc Conference 7
10.6	Gatekeeper actions 7
10.6.1	Call transfer performed by gatekeeper 7
11	Dynamic description of transfer with rerouting 9
11.1	Transfer with rerouting without secondary call 9
11.1.1	Operational model for Transfer if no gatekeeper is involved in the service. 9
11.1.2	Operational model for transfer using gatekeeper routed model 10
11.2	Transfer with secondary call 12
11.2.1	Operational model for transfer if no gatekeeper is involved in the service .. 12
11.2.2	Operational model for transfer using gatekeeper routed model 13

	Page	
11.3	Communication between transferring signalling entity TRGSE and TRGSE user....	19
11.3.1	Table of primitives	19
11.3.2	Primitive definition	20
11.3.3	Parameter definition	20
11.3.4	States	20
11.4	Communication between transferred signalling entity TRDSE and TRDSE user.....	20
11.4.1	Table of primitives	20
11.4.2	Primitive definition	21
11.4.3	Parameter definition	22
11.4.4	States	22
11.5	Communication between transferred-to signalling entity TRTSE and TRTSE user..	23
11.5.1	Table of primitives	23
11.5.2	Primitive definition	23
11.5.3	Parameter definition	24
11.5.4	States	24
11.6	Peer-to-peer communication for transfer with rerouting.....	24
11.6.1	Messages	24
11.6.2	Timers	25
11.6.3	Counters	25
11.6.4	Message flow examples	25
12	Operations in support of SS-CT	29
13	Specification and Description Language (SDL) diagrams.....	33
13.1	Call Transfer block diagram.....	33
13.2	Call transfer substructure diagram.....	33
13.3	Transferring signalling entity SDLs	35
13.4	Transferred signalling entity SDLs.....	38
13.5	Transferred-to signalling entity SDLs.....	41

Recommendation H.450.2

CALL TRANSFER SUPPLEMENTARY SERVICE FOR H.323

(Geneva, 1998)

1 Scope

This Recommendation describes the Call Transfer supplementary service (SS-CT), which is applicable to various basic services supported by H.323 Multimedia Endpoints. SS-CT is based on the equivalent supplementary service for Private Integrated Services Networks (PISN) specified in ISO/IEC 13865 and 13869.

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; 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 H.225.0 (1998), *Call signalling protocols and media stream packetization for packet based multimedia communications systems.*
- ITU-T Recommendation H.245 (1998), *Control protocol for multimedia communication.*
- ITU-T Recommendation H.323 (1998), *Packet based multimedia communications systems.*
- ITU-T Recommendation H.450.1 (1998), *Generic functional protocol for the support of supplementary services in H.323.*
- ITU-T Recommendation I.112 (1993), *Vocabulary of terms for ISDNs.*
- ITU-T Recommendation I.210 (1993), *Principles of telecommunication services supported by an ISDN and the means to describe them.*
- ISO/IEC 13865:1995, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Specification, functional model and information flows – Call transfer supplementary service.*
- ISO/IEC 13869:1995, *Information technology – Telecommunications and information exchange between systems – Private Integrated Services Network – Inter-exchange signalling protocol – Call transfer supplementary service.*

3 Terms and definitions

This Recommendation defines the following terms:

3.1 endpoint; terminal; user: See Recommendation H.323.

3.2 transferring endpoint; user A: User/Endpoint transferring a call with a terminal B, to a user C.

3.3 transferred endpoint; user B: The remote endpoint/user of the call with user A which is transferred by user A to a user C.

- 3.4 transferred-to endpoint; user C:** The endpoint/user, to which a user B, originally having a call to user A, is transferred.
- 3.5 primary call:** Call between the transferring user A and the transferred user B.
- 3.6 secondary call:** Call between the transferring user A and the transferred-to user C associated with the primary call (this association is an implementation matter within endpoint A).
- 3.7 transferred call:** Call between transferred user B and transferred-to user C which results from successful completion of call transfer.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
CT	Call Transfer
GK/GKp/GKs	Gatekeeper/Gatekeeper on primary call/Gatekeeper on secondary call
MC	Multipoint Controller
PISN	Private Integrated Services Network
SCN	Switched Circuit Network
SDL	Specification and Description Language
SS-CT	Supplementary Service Call Transfer
TRDSE	Transferred Signalling Entity
TRGSE	Transferring Signalling Entity
TRTSE	Transferred-to Signalling Entity

5 Description

Call Transfer (SS-CT) is a supplementary service which enables the served user (user A or transferring user) to transform an existing call with user B (primary call) into a new call between user B and a user C (transferred-to user) selected by user A. User A may or may not have a call established with user C (secondary call) prior to transfer. Each call can either be an incoming call to user A or an outgoing call from user A. On successful completion of SS-CT, user B and user C can communicate with each other and user A will no longer be able to communicate with user B or user C.

The initial call between user A and user B (primary call) must be answered before transfer can be initiated. On invocation of call transfer, if a call between user A and user C exists (secondary call), the transferred-to endpoint is informed of the pending call transfer, and transfer only proceeds if this endpoint is able to participate. In this case, the transferred-to endpoint returns a temporary identifier to be used in the subsequent transfer procedure to identify the secondary call.

The transferring endpoint (user A) requests the transferred endpoint (user B) to call the transferred-to endpoint (user C). The transferred endpoint then establishes a call to the transferred-to endpoint (transfer by rerouting) and includes the temporary identifier for the secondary call if this call exists. User B's gatekeeper may establish the call if it is handling the call transfer.

The primary call is retained until the first acknowledgment has been received from the transferred-to endpoint (user C), and is then released. This means that the primary call remains in place if call transfer fails before that stage. If the secondary call exists, it is retained until the new call request arrives at the transferred-to endpoint and is then released.

Upon answer from user C (if no secondary call exists) or successful association of user C with the new call (if user C was already involved in the secondary call), user B and user C can communicate with each other. The media exchange capabilities of the new call should be equal to those of the primary call as far as possible.

If call transfer fails after the primary call was released, the action taken is for further study.

NOTE – Use of Call Transfer Supplementary service for transferring only a subset of H.323 media streams or for transferring T.120 calls is for further study.

6 Messages and information elements

The APDUs of call transfer operations are transported within User-to-User Information elements in call control and FACILITY messages as defined in Recommendation H.450.1. The call transfer operations are described in clause 12.

When conveying the invoke APDU of the operations defined in clause 12, the destinationEntity data element of the NFE shall contain the value "endpoint".

When conveying the invoke APDU of operations callTransferAbandon, callTransferUpdate, callTransferComplete, callTransferActive and subaddressTransfer, the Interpretation APDU shall contain value discardAnyUnrecognizedInvokePdu.

When conveying the invoke APDU of operation callTransferSetup, the Interpretation APDU shall contain value clearCallIfAnyInvokePduNotRecognized.

When conveying the invoke APDU of operation callTransferIdentify or callTransferInitiate, the Interpretation APDU shall be omitted or shall contain value rejectAnyUnrecognizedInvokePdu.

7 Actions at the transferring endpoint (user A)

In the procedures below, "clearing" or "release" of a call means the release of all resources associated with the call, as specified in Recommendation H.323 (i.e. all logical channels, H.245 control channel, call signalling channel, call reference, etc.).

7.1 Normal procedures if only the primary call exists

In order to initiate call transfer, the transferring endpoint shall send a callTransferInitiate invoke APDU in a FACILITY message to the transferred endpoint, using the call reference of the primary call, start timer CT-T3 and enter state CT-Await-Initiate-Response. The invoke APDU shall contain element reroutingNumber with the address of user C as selected by user A. Element callIdentity shall be empty.

On receipt of a callTransferInitiate return result APDU on the primary call, the transferring endpoint shall stop timer CT-T3, clear the primary call and enter state CT-Idle.

If the primary call is released while in state CT-Await-Initiate-Response, the transferring endpoint shall stop timer CT-T3 and enter state CT-Idle.

7.2 Normal procedures if also the secondary call exists

In order to initiate call transfer, the transferring endpoint shall send a callTransferIdentify invoke APDU in a FACILITY message to the transferred-to endpoint, using the call reference of the secondary call, start timer CT-T1 and enter state CT-Await-Identify-Response.

On receipt of a callTransferIdentify return result APDU on the secondary call, the transferring endpoint shall stop timer CT-T1, send a callTransferInitiate invoke APDU in a FACILITY message to the transferred endpoint, using the call reference of the primary call, start timer CT-T3 and enter state CT-Await-Initiate-Response. The invoke APDU shall contain elements reroutingNumber and callIdentity with the values received in callTransferIdentify return result.

On receipt of a callTransferInitiate return result APDU on the primary call, the transferring endpoint shall stop timer CT-T3, clear the primary and secondary call if not already cleared, and enter state CT-Idle.

If the primary and/or secondary call is released while in state CT-Await-Identify-Response or CT-Await-Initiate-Response, the transferring endpoint shall stop the relevant timer, as an option clear the other call if not already cleared, and enter state CT-Idle.

7.3 Exceptional procedures

On receipt of a callTransferIdentify reject or return error APDU while in state CT-Await-Identify-Response, the transferring endpoint shall stop timer CT-T1 and enter state CT-Idle.

On receipt of a callTransferInitiate reject or return error APDU while in state CT-Await-Initiate-Response, the transferring endpoint shall send a callTransferAbandon invoke APDU in a FACILITY message on the secondary call if this call exists, stop timer CT-T3 and enter state CT-Idle.

On expiry of timer CT-T1 or CT-T3, the transferring endpoint shall send a callTransferAbandon invoke APDU in a FACILITY message on the secondary call if this call exists, and enter state CT-Idle.

8 Actions at the transferred endpoint (user B)

8.1 Normal procedures

On receipt of a callTransferInitiate invoke APDU on the primary call, the transferred endpoint shall determine whether it can participate in the call transfer. If it can it shall use the address contained in element reroutingNumber to initiate call establishment towards the transferred-to endpoint, include in the SETUP message a callTransferSetup invoke APDU with the same value of callIdentity as received in the callTransferInitiate invoke APDU, and enter state CT-Await-Setup-Response. The callTransferSetup invoke APDU may also include the address of user A in element transferringNumber. Optionally, timer CT-T4 may be started.

On receipt of an ALERTING or CONNECT message with a callTransferSetup return result APDU from the transferred-to endpoint, the transferred endpoint shall stop timer CT-T4 if running, send a RELEASE COMPLETE message with a callTransferInitiate return result APDU on the primary call to the transferring endpoint and enter state CT-Idle. Further connection establishment between user B and user C shall follow standard H.323 procedures, whereby media properties may be inherited from the primary call or automatically negotiated between endpoints B and C.

8.2 Exceptional procedures

If, on receipt of a callTransferInitiate invoke APDU, the transferred endpoint cannot participate in call transfer, a callTransferInitiate return error APDU shall be sent in a FACILITY message on the primary call.

On expiry of timer CT-T4, or on receipt of a clearing indication on the new call while in state CT-Await-Setup-Response, the transferred endpoint shall send a callTransferInitiate return error APDU in a FACILITY message on the primary call, stop timer CT-T4 if running, and enter state CT-Idle. If a callTransferSetup return error APDU was received from the transferred-to endpoint, the same error value shall be used in the callTransferInitiate return error APDU.

If the primary call is cleared while in state CT-Await-Setup-Response, the transferred endpoint shall stop timer CT-T4 if running, clear the new call and enter state CT-Idle.

9 Actions at the transferred-to endpoint (user C)

9.1 Normal procedures

On receipt of a callTransferIdentify invoke APDU on the secondary call, the transferred-to endpoint shall determine whether it can participate in the call transfer. If it can, it shall send a callTransferIdentify return result APDU in a FACILITY message on the secondary call, start timer CT-T2 and enter state CT-Await-Setup. The return result APDU shall contain user C's address in element reroutingNumber and a local identifier for the secondary call in element callIdentity.

On receipt of a SETUP message with a callTransferSetup invoke APDU from the transferred endpoint, the transferred-to endpoint shall, if in state CT-Await-Setup, stop timer CT-T2, check the destination address information of the SETUP message and element callIdentity received in the callTransferSetup invoke APDU, and continue with call establishment if they correctly identify the secondary call. The secondary call shall be cleared using normal H.323 procedures.

If the callTransferSetup invoke APDU is received in state CT-Idle, call setup shall continue if element callIdentity is empty and if call transfer to user C is acceptable.

Either the ALERTING or the CONNECT message on the new call shall include a callTransferSetup return result APDU.

9.2 Exceptional procedures

If, on receipt of a callTransferIdentify invoke APDU, the transferred-to endpoint cannot participate in call transfer, a callTransferIdentify return error APDU shall be sent in a FACILITY message on the secondary call.

If, on receipt of a callTransferSetup invoke APDU with an empty element callIdentity, the transferred-to endpoint cannot accept call transfer to user C, the endpoint shall return a RELEASE COMPLETE message with a callTransferSetup return error APDU.

If, on receipt of a callTransferSetup invoke APDU with a non-empty element callIdentity, the transferred-to endpoint cannot match the identifier with a secondary call, the endpoint shall return a RELEASE COMPLETE message with a callTransferSetup return error APDU.

If, in state CT-Await-Setup, a callTransferAbandon invoke APDU is received on the secondary call, or if the secondary call is cleared, the transferred-to endpoint shall stop timer CT-T2 and enter state CT-Idle.

On expiry of timer CT-T2, the transferred-to endpoint shall enter state CT-Idle.

10 Additional procedures

10.1 Interworking with non-H.323 terminals

In the specification of procedures above, the term "endpoint" means any entity that processes H.225 call signalling. Examples are "terminal" or "gateway".

If user B or user C are non-H.323 terminals, the gateway on the route to user B or user C will act as transferred endpoint or transferred-to endpoint, respectively.

10.2 Hold and consultation

Call transfer procedures do not demand a particular hold state for the primary or secondary call; i.e. user B or user C may or may not be put on hold prior to call transfer, depending on the capabilities of terminal A and the specific implementation.

With existing (e.g. PISN) implementations, call transfer is frequently performed by user A putting user B on hold, setting up the secondary call to user C ("consultation call") and then invoking transfer. This method is also shown in an example in 11.6.

10.3 Transfer without rerouting (i.e. transfer by join)

ISO/IEC 13869 defines two procedures for call transfer:

- 1) transfer by rerouting; and
- 2) transfer by join.

In this Recommendation, where the transferring endpoint typically is the terminal rather than a private integrated network exchange, the transfer by rerouting procedure was chosen as the primary procedure (it is not expected from a H.323 terminal to join two calls in which it no longer participate).

However, the transfer by join procedure may be used in this Recommendation if a gatekeeper performs the transfer (e.g. if the terminals do not support H.450.2 call transfer by rerouting) or in case of gateway interworking with the corresponding SCN transfer by join procedure (see also 10.6.1 and 11.2.2.2).

10.4 Further actions at transferred and transferred-to endpoints

Together with the callTransferSetup invoke APDU, a callTransferUpdate invoke APDU may be sent in the SETUP message from the transferred endpoint to the transferred-to endpoint. Within the argument, optional elements redirectionNumber and basicCallInfoElements may be conveyed. If a callTransferUpdate invoke APDU is received by the transferred-to endpoint, this information may be conveyed to the transferred-to user.

Together with the callTransferSetup return result APDU, a callTransferUpdate invoke APDU may be sent in the ALERT or CONNECT message from the transferred-to endpoint to the transferred endpoint. Within the argument, optional elements redirectionNumber and basicCallInfoElements may be conveyed. If a callTransferUpdate invoke APDU is received by the transferred endpoint, this information may be conveyed to the transferred user.

If during CT-Idle, a FACILITY message containing a subaddressTransfer invoke APDU is received, the endpoint may relay the subaddress on to the local user.

10.5 Interactions with H.323 Ad Hoc Conference

Call transfer may be used in the context of conference.

An H.323 endpoint A (that does not have the active MC) may transfer its connection to the conference to another endpoint D. For this purpose, the endpoint A shall send a FACILITY message containing a callTransferInitiate invoke APDU towards the conference entity B containing the Active MC (may be the GK or another endpoint). The entity B may then issue a new SETUP message to endpoint D, containing a callTransferSetup invoke APDU and conferenceGoal = "invite". After successful establishment of the H.225 channel between entity B (active MC) and the new endpoint D, the call between entity B and endpoint A (transferring endpoint) is released following normal call transfer procedures. Establishment of the H.245 channel and of logical channels between entity B and endpoint D follows 8.4.3/H.323.

An H.323 endpoint A that contains the active MC may transfer an endpoint B to an endpoint C, if the call between the endpoints A and B is a two-party conference call at the time of call transfer request.

An H.323 endpoint that has the active MC shall reject a call transfer request, if the active MC serves a multipoint conference at the time of call transfer request.

10.6 Gatekeeper actions

In the case of the gatekeeper routed model, the gatekeeper shall either transparently transport (see also Figure 4) or act on the operations defined in clause 12 (see 10.6.1).

10.6.1 Call transfer performed by gatekeeper

A gatekeeper may decide to become the destination for a callTransferInitiate invoke APDU. In this case, the gatekeeper shall perform the actions applicable for the transferred endpoint with the additions described in 10.6.1.1.

A gatekeeper may decide to become the destination for a callTransferIdentify invoke APDU. In this case, the gatekeeper shall perform the actions applicable for the transferred-to endpoint with the additions described in 10.6.1.2.

A gatekeeper may be able to act as the destination for both sides of the call transfer, in which case it performs actions appropriate for both transferred and transferred-to endpoints, with the exception that messages to be exchanged between transferred endpoint and transferred-to endpoint become information flows internal to the gatekeeper.

When a gatekeeper is handling the call transfer signalling on behalf of an endpoint, the gatekeeper may choose to relay call transfer APDUs (callTransferComplete, callTransferActive, callTransferUpdate) to the endpoint.

10.6.1.1 Gatekeeper actions on the primary call

The callTransferInitiate invoke APDU sent from the transferring endpoint A to endpoint B may be intercepted by the GK, i.e. the callTransferInitiate invoke APDU does not reach the transferred endpoint B. The GK shall perform further procedures as defined for a transferred endpoint in clause 8. This includes the sending of a SETUP message with a callTransferSetup invoke APDU and an optional callTransferUpdate invoke APDU to the transferred-to endpoint C, the reception of the ALERTING or CONNECT message with a callTransferSetup return result APDU and, optionally, a callTransferUpdate invoke APDU as well as the sending of a RELEASE COMPLETE message on the primary call to endpoint A, including a callTransferInitiate return result APDU.

In addition to the call transfer procedures defined in clause 8, the gatekeeper may send a FACILITY message with a callTransferComplete invoke APDU to endpoint B to inform endpoint B that it has

been transferred ("joining"). The callTransferComplete invoke APDU shall be sent upon reception of an ALERTING or a CONNECT message from the transferred-to endpoint C. Argument redirectionNumber shall contain the number of the transferred-to user C, argument endDesignation shall have the value "primaryEnd", argument callStatus shall be set to either "answered" (if CONNECT was received) or "alerting" (if ALERTING was received).

If the transferred-to endpoint initially responded to the SETUP with an ALERTING message, a FACILITY message containing a callTransferActive invoke APDU may be sent to endpoint B upon reception of a CONNECT message from the transferred-to endpoint.

For the call which was originally active between endpoints A (the transferring party) and B (the transferred party), endpoint B shall be instructed to connect a new set of media channels, assuming that media channels were not connected to the gatekeeper acting as an MCU. To accomplish this, the gatekeeper shall use the H.323 procedures for third party rerouting. This requires the gatekeeper to send an empty terminal capability set (one which indicates that the remote entity has no receive capabilities) to endpoints A and B, causing A and B to close their logical channels. The gatekeeper shall then exchange the H.245 command "end session" with endpoint A and send a RELEASE COMPLETE message containing callTransferInitiate return result APDU to release the call signalling channel.

When the gatekeeper receives a non-empty terminal capability set from endpoint C, the gatekeeper shall forward the capability set to endpoint B to cause it to reset its H.245 associated state to that which it was in when H.245 has just completed (the first) terminal capability set exchange in the initial call establishment sequence; it shall take part in master/slave determination, and open appropriate logical channels with endpoint C.

10.6.1.2 Gatekeeper actions on the secondary call

The callTransferIdentify invoke APDU sent from the transferring endpoint A to endpoint C may be intercepted by a GK, i.e. the callTransferIdentify invoke APDU does not reach the transferred-to endpoint C. The gatekeeper shall perform further procedures as defined for a transferred-to endpoint in clause 9. This includes sending ALERTING and CONNECT messages (as appropriate to reflect the state of the secondary call) in response to the SETUP with a callTransferSetup invoke APDU received from endpoint B. In addition, the gatekeeper shall perform procedures specified in this subclause.

The gatekeeper must ensure that it is able to receive and handle call signalling from the transferred-to endpoint (or its gatekeeper).

If media channels associated with the secondary call are already open at the time of sending the callTransferSetup return result APDU (typically in case of transfer into active, but may also apply to transfer into alerting), they need to be closed. For this purpose, the gatekeeper shall instruct endpoint C to close its media channels to endpoint A using the H.323 procedures for third party rerouting. This requires the gatekeeper to send an empty terminal capability set (one which indicates that the remote entity has no receive capabilities) to endpoints A and C, causing them to close their logical channels. The gatekeeper shall exchange the H.245 command "end session" with endpoint A and send a RELEASE COMPLETE message to release the call signalling channel.

If no media channels associated with the secondary call are open at the time of sending the callTransferSetup return result APDU (may be true e.g. in case of transfer into alerting), the gatekeeper shall exchange the H.245 command "end session" with endpoint A (if an H.245 channel was established between endpoints A and C) and send a RELEASE COMPLETE message to endpoint A to release the call signalling channel. The CONNECT message received from endpoint C shall be sent towards the transferred endpoint B.

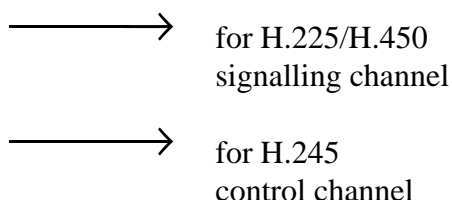
In addition to the call transfer procedures defined in clause 9, the gatekeeper may send a FACILITY message with a callTransferComplete invoke APDU to endpoint C to inform it that it has been transferred ("joining"). The callTransferComplete invoke APDU shall be sent to the transferred-to endpoint C at the time the first of either an ALERTING or a CONNECT message containing a callTransferSetup return result APDU is sent to the transferred endpoint B. Argument redirectionNumber shall contain the number of the transferred user B, argument endDesignation shall have the value "secondaryEnd", and argument callStatus shall be set to "answered".

When H.245 control signalling is established with the transferred terminal (terminal B or its gatekeeper), the gatekeeper shall send a TerminalCapabilitySet representing endpoint C's capabilities.

11 Dynamic description of transfer with rerouting

The following subclauses show examples of call transfer scenarios by describing the operational model and corresponding signalling flows.

The following notation is used:



NOTE – Within this clause, "ct" is used as an abbreviation for the term "callTransfer" as part of call transfer operation names (Example: ctInitiate.Invoke refers to callTransferInitiate invoke APDU).

11.1 Transfer with rerouting without secondary call

11.1.1 Operational model for Transfer if no gatekeeper is involved in the service

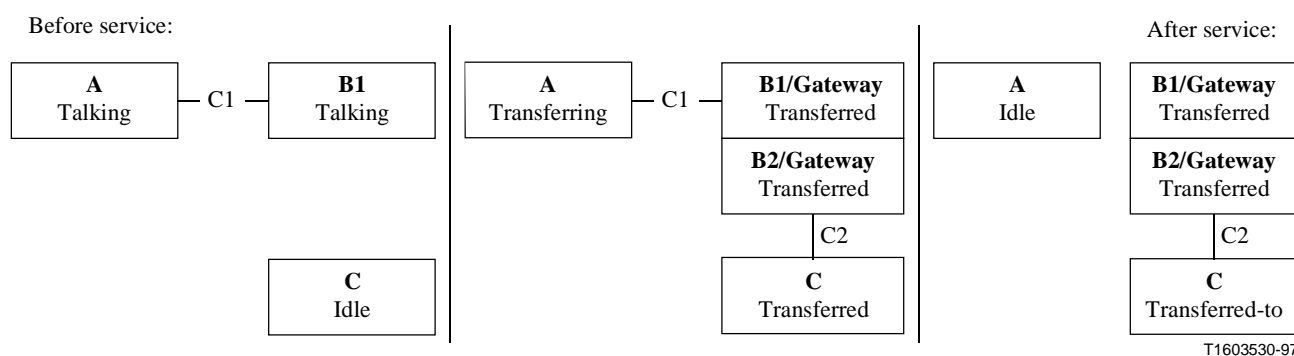


Figure 1/H.450.2 – Model for transfer with rerouting without secondary call

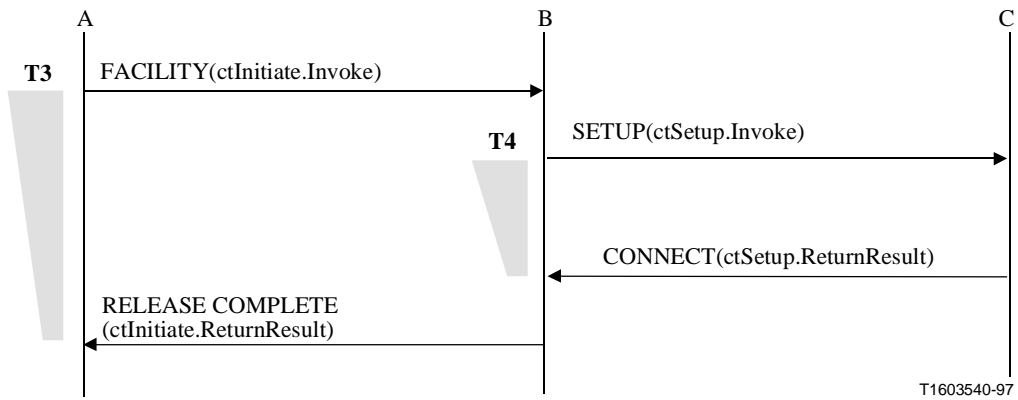
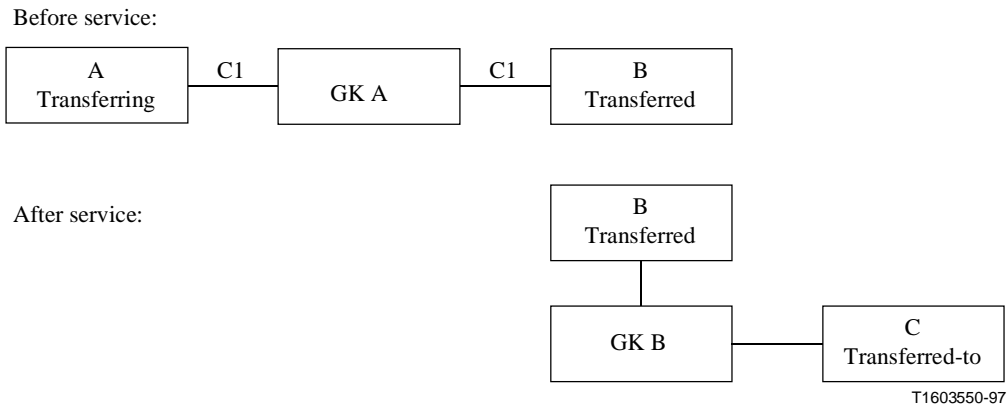


Figure 2/H.450.2 – Signalling flow for transfer with rerouting without secondary call

11.1.2 Operational model for transfer using gatekeeper routed model

11.1.2.1 Gatekeeper is transparent for call transfer operations



NOTE – In this example, gatekeeper routed model is used for the call between endpoints A and B as well as for the call between endpoint B and endpoint C. In this example, the endpoints A, B are registered with different gatekeepers.

Figure 3/H.450.2 – Operational model for transfer using gatekeeper routed model – GK is transparent for call transfer operations

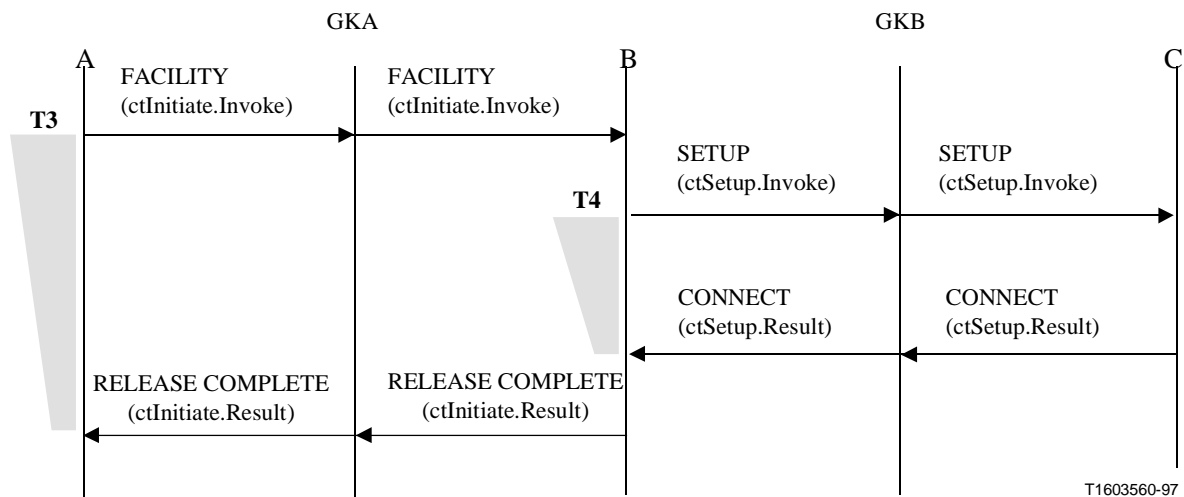
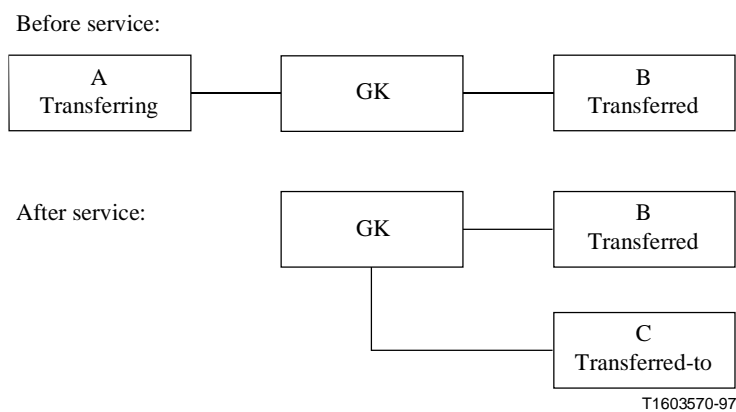


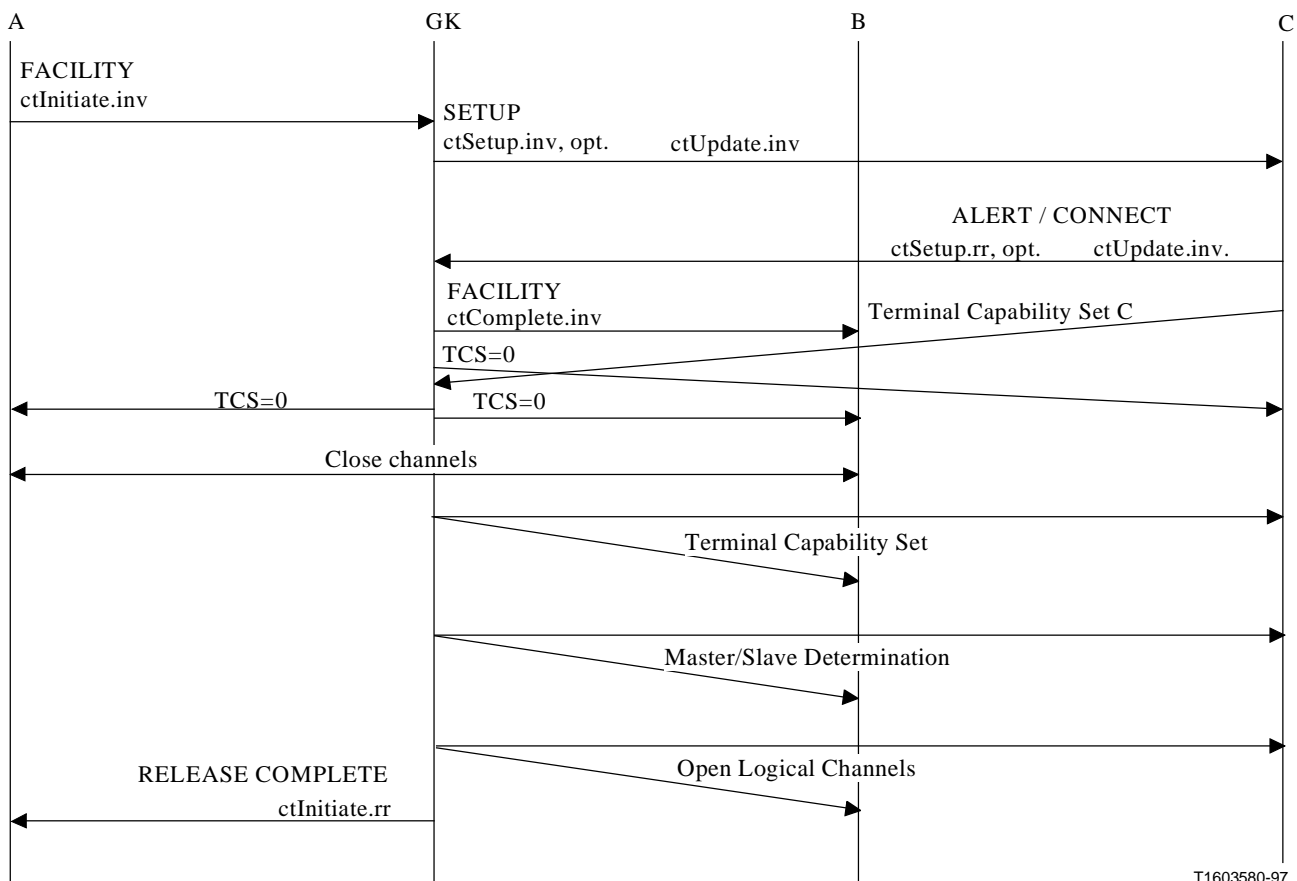
Figure 4/H.450.2 – Signalling flow for transfer using gatekeeper routed model – GK is transparent for call transfer operations

11.1.2.2 Call transfer supplementary service performed by gatekeeper



NOTE – In this example, gatekeeper routed model is used for the call between endpoints A and B. The gatekeeper intercepts the callTransferInitiate.inv apdu.

Figure 5/H.450.2 – Operational model for transfer using gatekeeper routed model – GK intercepts call transfer APDUs



TCS=0 means Empty Terminal Capability Set

Figure 6/H.450.2 – Signalling flow for transfer using gatekeeper routed model – GK intercepts call transfer

11.2 Transfer with secondary call

11.2.1 Operational model for transfer if no gatekeeper is involved in the service

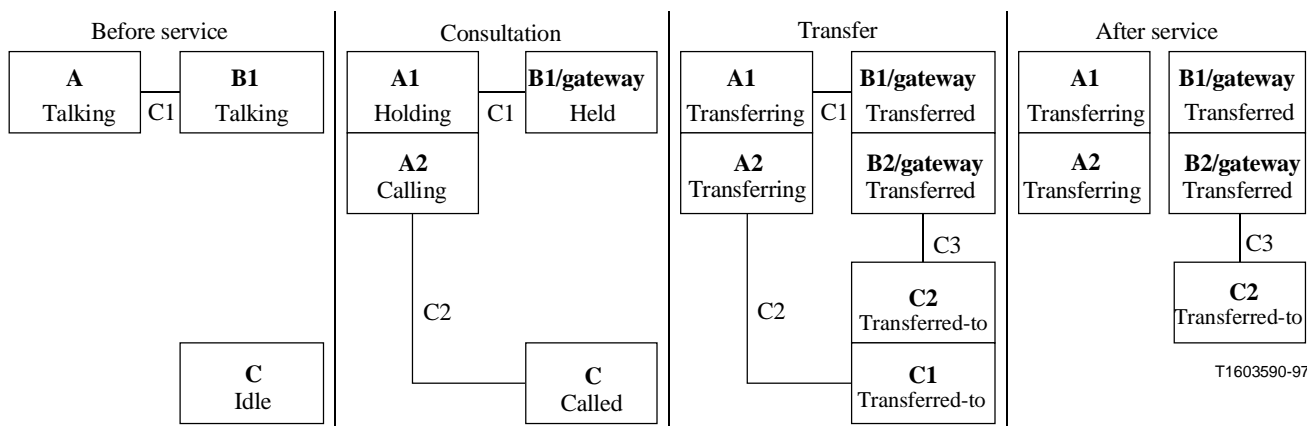


Figure 7/H.450.2 – Operational model for transfer with secondary call

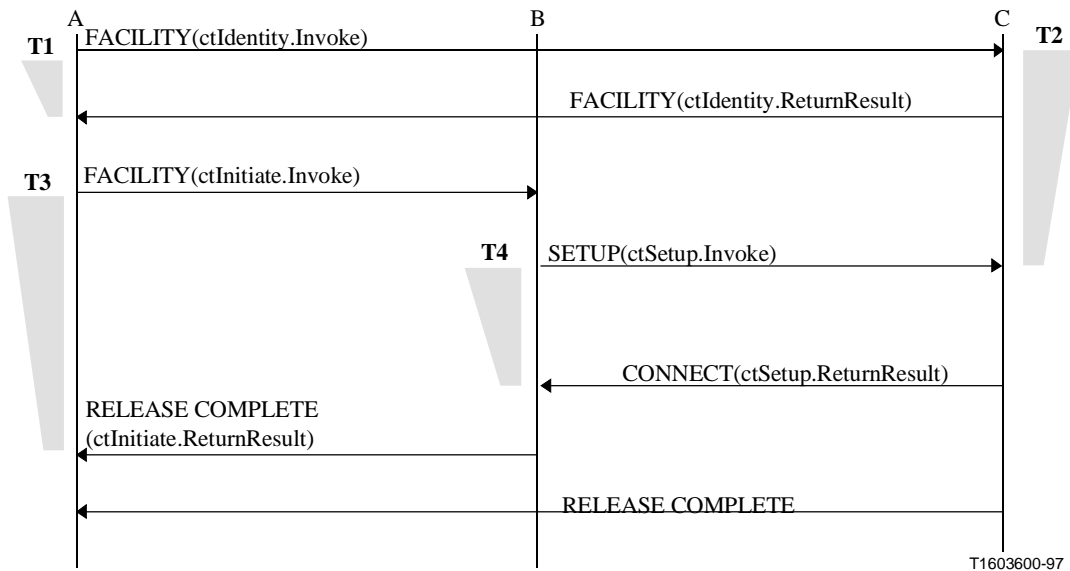


Figure 8/H.450.2 – Signalling flow for transfer with secondary call

11.2.2 Operational model for transfer using gatekeeper routed model

It is possible to have a mixed model where one gatekeeper involved in the transfer handles the call transfer for its endpoint (as described in 11.2.2.2) while another gatekeeper involved in the same transfer allows transparent operation (see 11.2.2.1).

11.2.2.1 Gatekeeper is transparent for call transfer operations

Figures 9 and 10 show signalling paths where each end of both the primary and secondary calls is gatekeeper routed. Where an end of a call uses direct signalling, that end reduces to the model shown in 11.2.1 without a gatekeeper.

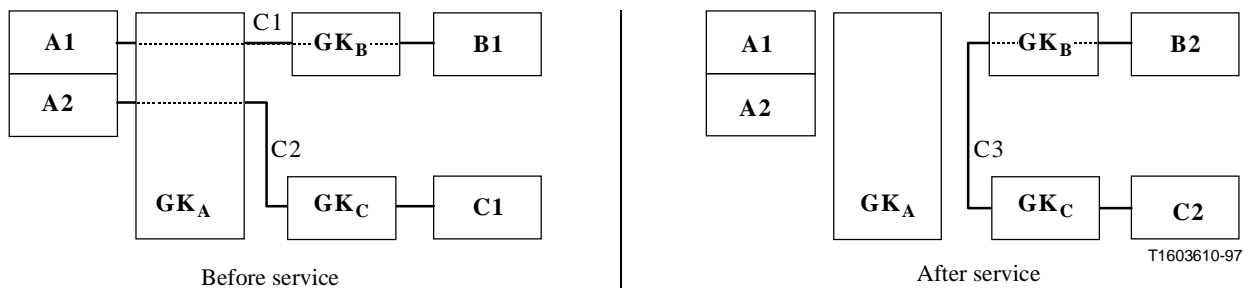


Figure 9/H.450.2 – Operational model for call transfer with consultation – GK is transparent for call transfer operations

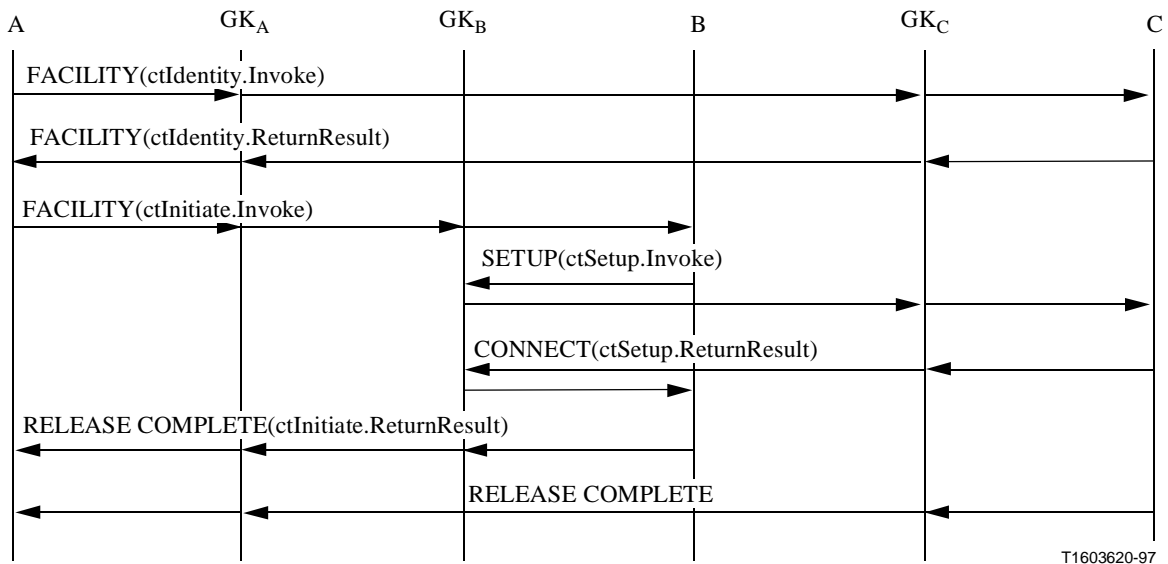


Figure 10/H.450.2 – Signalling flow for call transfer with consultation – Gatekeeper is transparent for call transfer operations

11.2.2.2 Call transfer supplementary service performed by gatekeeper

This subclause gives examples on call models for call transfer with secondary call with one or more gatekeepers performing the call transfer supplementary service. In Figures 11, 12 and 13, there may be one or more gatekeepers on the call signalling path. The call transfer signalling may be handled by any gatekeeper on the path. For that reason, only the gatekeeper handling the call transfer is shown (as GK_p on the primary call, and GK_s on the secondary call). Other gatekeepers on the call signalling path which are not shown are transparent for call transfer APDUs.

Note that in Figures 12 and 13, appropriate signalling to establish gatekeeper routing at the transferred-to endpoint (user C) is required.

For some selected call models, signalling flows are given in Figures 11 to 18.

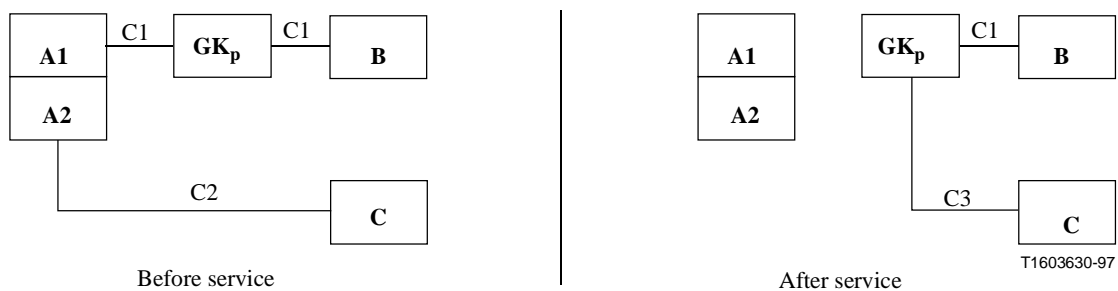
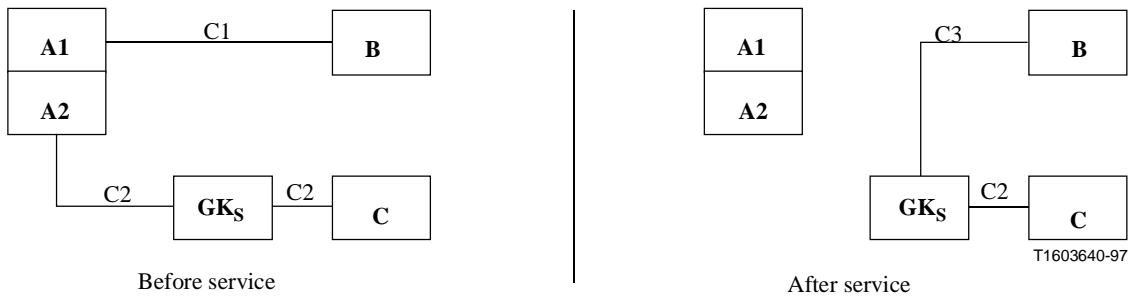
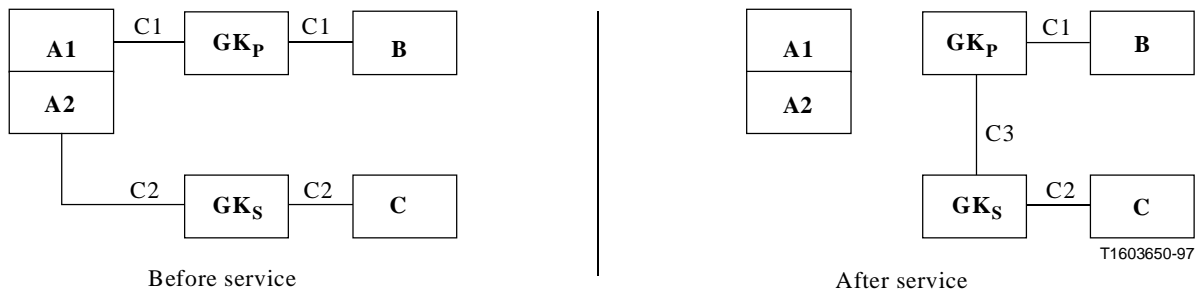


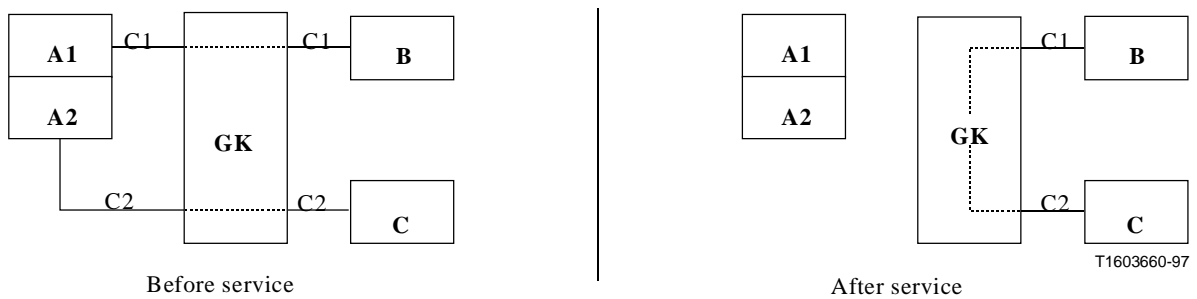
Figure 11/H.450.2 – Call transfer with consultation – Primary call routed through gatekeeper



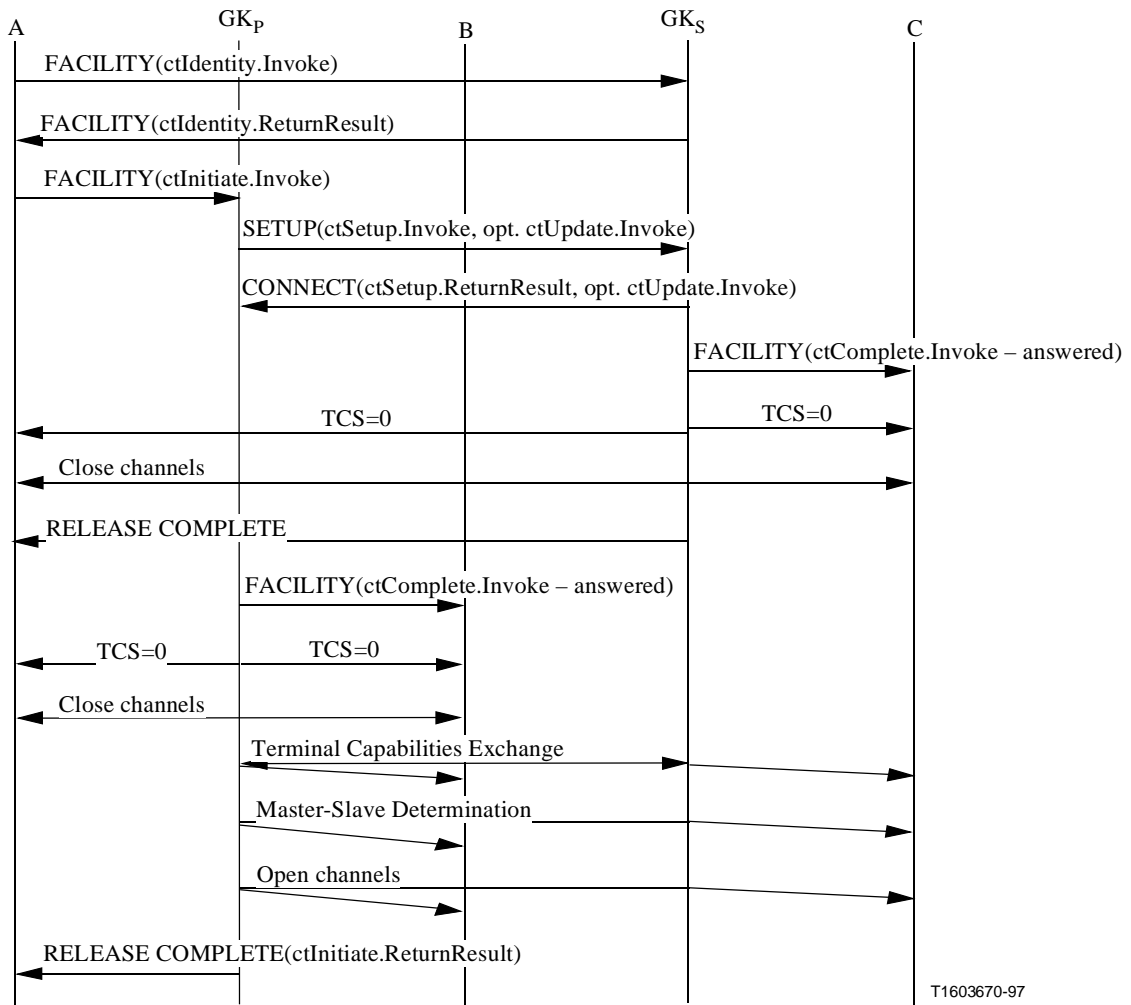
**Figure 12/H.450.2 – Call transfer with consultation –
Secondary call routed through gatekeeper**



**Figure 13/H.450.2 – Call transfer with consultation –
Primary and secondary calls routed through different gatekeepers**



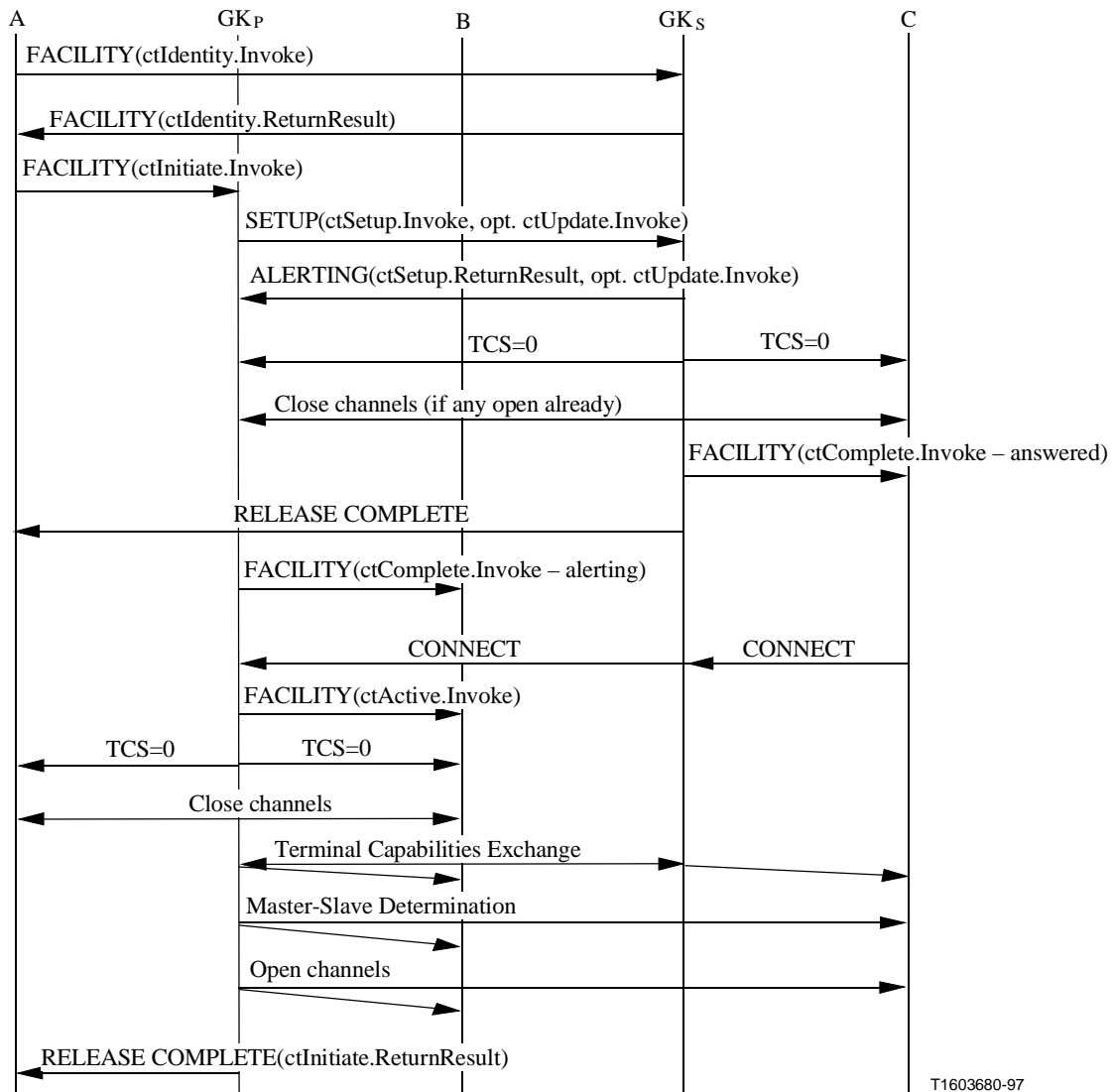
**Figure 14/H.450.2 – Call transfer with consultation –
Primary and secondary calls routed through common gatekeeper**



T1603670-97

TCS=0 means Empty Terminal Capability Set

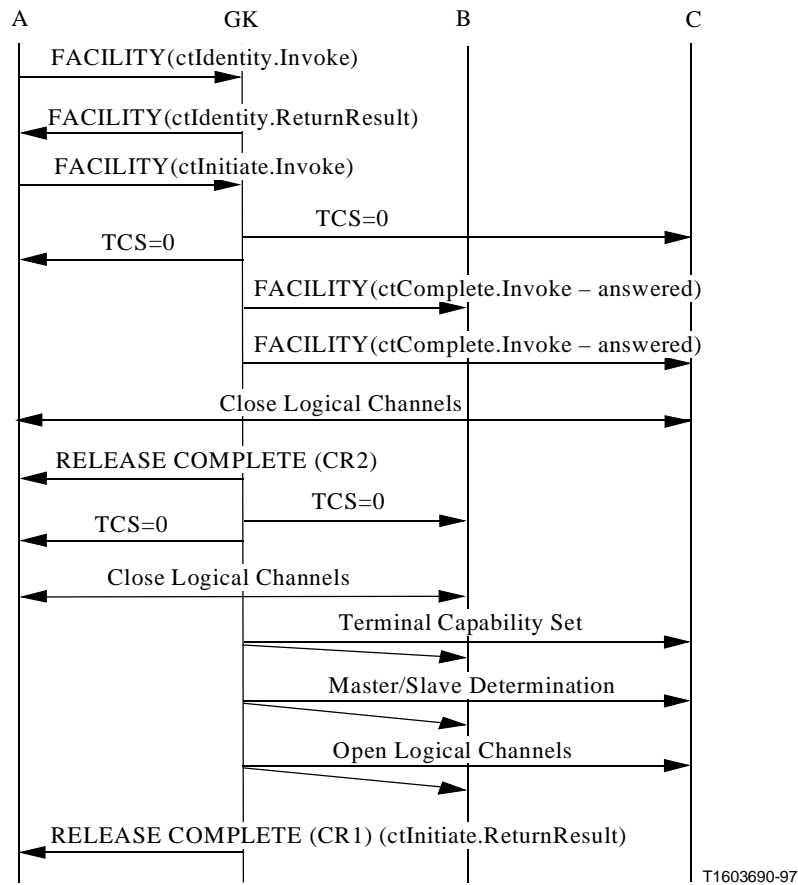
Figure 15/H.450.2 – Call transfer with consultation – Distinct gatekeeper: Secondary call connected (based on operational model given in Figure 13)



T1603680-97

TCS=0 means Empty Terminal Capability Set

Figure 16/H.450.2 – Call transfer with consultation – Distinct gatekeeper: Secondary call alerting (based on operational model given in Figure 13)



TCS=0 means Empty Terminal Capability Set

Figure 17/H.450.2 – Call transfer with consultation – Common gatekeeper: Secondary call connected (based on operational model given in Figure 14)

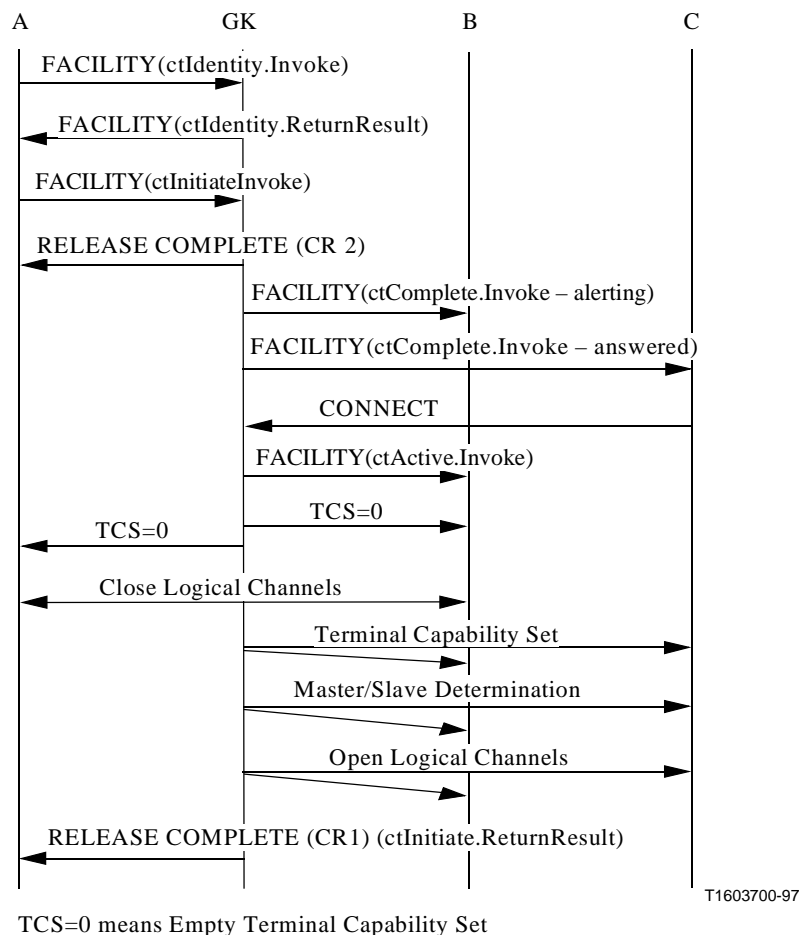


Figure 18/H.450.2 – Call transfer with consultation – Common gatekeeper: Secondary call alerting (based on operational model given in Figure 14)

11.3 Communication between transferring signalling entity TRGSE and TRGSE user

11.3.1 Table of primitives

Communication between the TRGSE and TRGSE user is performed using the primitives shown in Table 1.

Table 1/H.450.2 – TRGSE primitives

Generic name	Type			
	request	indication	response	confirm
CTInitiate	PARAMETERS	Not defined ^{a)}	Not defined	– ^{b)}
CTIdentify ^{c)}	–	Not defined	Not defined	PARAMETERS
CTAbandon	Not defined	–	Not defined	Not defined

^{a)} "Not defined" means that this primitive is not defined.
^{b)} "–" means no parameters.
^{c)} Only applicable in case of transfer with secondary call.

11.3.2 Primitive definition

The definition of these primitives is as follows:

- a) The CTInitiate.request primitive is used to make the transfer request.
- b) The CTInitiate.confirm primitive is used to indicate the result of the transfer request.
- c) The CTIdentify.request primitive is used to request a call identity from TRTSE.
- d) The CTIdentify.confirm primitive is used to confirm the call identity of TRTSE.
- e) The CTAbandon.indication primitive is used to indicate transfer unsuccessful notification to the TRTSE.

11.3.3 Parameter definition

CTInitiate.req parameters

- callIdentity: call identifier of secondary call;
- reroutingNumber: address of user C;
- extension: Placeholder for vendor-specific extensions.

CTIdentify.conf parameters

- callIdentity: call identifier of secondary call;
- reroutingNumber: address of user C;
- extension: Placeholder for vendor-specific extensions.

11.3.4 States

The following states are used to specify the allowed sequence of primitives between the TRGSE and TRGSE user:

CT-Idle

Supplementary service Call Transfer is not operating.

CT-Await-Identify-Response

A callTransferIdentify invoke APDU has been sent to the TRTSE. This state is used during transfer by rerouting with secondary call.

CT-Await-Initiate-Response

A callTransferInitiate invoke APDU has been sent to the TRDSE. This state is used during transfer by rerouting.

11.4 Communication between transferred signalling entity TRDSE and TRDSE user

11.4.1 Table of primitives

Communication between the TRDSE and TRDSE user is performed using the primitives shown in Table 2.

Table 2/H.450.2 – TRDSE Primitives

Generic name	Type			
	request	indication	response	confirm
CTInitiate	Not defined ^{a)}	PARAMETERS	– ^{b)}	Not defined
CTSetup	PARAMETERS	Not defined	Not defined	–
SubaddressTransfer	PARAMETERS	PARAMETERS	Not defined	Not defined
CTUpdate	PARAMETERS	PARAMETERS	Not defined	Not defined
CTComplete	PARAMETERS ^{c)}	PARAMETERS ^{d)}	Not defined	Not defined
CTActive	PARAMETERS ^{c)}	PARAMETERS ^{d)}	Not defined	Not defined
^{a)} "Not defined" means that this primitive is not defined. ^{b)} "–" means no parameters. ^{c)} Applicable only for a gatekeeper, if it performs the call transfer (see 10.6.1). ^{d)} Applicable for the transferred endpoint, if a gatekeeper performs the call transfer procedures.				

11.4.2 Primitive definition

The definition of these primitives is as follows:

- a) The CTInitiate.indication primitive is used to notify about transfer initiate request.
- b) The CTInitiate.response primitive is used to indicate success of the transfer.
- c) The CTSetup.request primitive is used to request call establishment from TRTSE.
- d) The CTSetup.confirm primitive is used to indicate success of call establishment to TRTSE.
- e) The CTUpdate.request primitive is used to request sending of identification information to the TRTSE.
- f) The CTUpdate.indication primitive is used to notify TRDSE user about TRTSE identification.
- g) The SubaddressTransfer.request primitive is used to request transmission of TRDSE subaddress to the TRTSE.
- h) The SubaddressTransfer.indication primitive is used to notify TRTSE subaddress to the TRDSE user.
- i) The CTComplete.request primitive is used to request sending of call transfer information from the TRDSE located in gatekeeper to the TRDSE located in the transferred terminal endpoint (i.e. gatekeeper performs call transfer for the transferred endpoint).
- j) The CTComplete.indication primitive is used to indicate call transfer information from the TRDSE to the TRDSE user (i.e. gatekeeper has performed call transfer for the transferred endpoint).
- k) The CTActive.request primitive is used to request sending of call transfer active information from TRDSE located in a gatekeeper to the TRDSE located in the transferred terminal endpoint (i.e. gatekeeper performs transfer ringing for the transferred endpoint with secondary call existing).
- l) The CTActive.indication primitive is used to indicate to the transferred user that the TRTSE user has become active (only if gatekeeper performs call transfer).

11.4.3 Parameter definition

CTInitiate.ind parameters

- callIdentity: call identifier of secondary call;
- reroutingNumber: address of user C;
- extension: Placeholder for vendor-specific extensions.

CTSetup.req parameters

- callIdentity: call identifier of secondary call;
- transferringNumber: address of user A (optional);
- extension: Placeholder for vendor-specific extensions.

SubaddressTransfer.xxx parameters

- redirectionSubaddress: Party Subaddress;
- extension: Placeholder for vendor-specific extensions.

CTUpdate.xxx parameters

- redirectionNumber: address of remote party (B or C);
- redirectionInfo: name information of remote party (B or C);
- H.225 Information element (optional);
- extension: Placeholder for vendor-specific extensions.

CTComplete.xxx parameters

- redirectionNumber: address of new connected party (party C);
- basicCallInfoElements: H.225 Information element (optional);
- redirectionInfo: name information of new connected party (party C);
- callStatus: reflects the H.225 call state of the new connected party;
- extension: Placeholder for vendor-specific extensions.

CTActive.xxx parameters

- connectedAddress: address of user C;
- basicCallInfoElements: H.225 Information element (optional);
- connectedInfo: name information of user C (optional);
- extension: placeholder.

11.4.4 States

The following states are used to specify the allowed sequence of primitives between the TRDSE and TRDSE user.

CT-Idle

Supplementary service Call Transfer is not operating.

CT-Await-Setup-Response

A callTransferSetup invoke APDU has been sent to the TRTSE. This state is used during transfer by rerouting.

CT-Await-Connect

The Primary Call has been transferred to an alerting TRTSE, and the TRDSE has been notified. A CONNECT message indicating answering by the TRTSE is awaited.

11.5 Communication between transferred-to signalling entity TRTSE and TRTSE user

11.5.1 Table of primitives

Communication between the TRTSE and TRTSE user is performed using the primitives shown in Table 3.

Table 3/H.450.2 – TRTSE primitives

Generic name	Type			
	request	indication	response	confirm
CTSetup	Not defined ^{a)}	PARAMETERS	_ ^{b)}	Not defined
CTUpdate	PARAMETERS	PARAMETERS	Not defined	Not defined
CTAbandon	Not defined	–	Not defined	Not defined
CTIdentify ^{c)}	Not defined	–	PARAMETERS	Not defined
SubaddressTransfer	PARAMETERS	PARAMETERS	Not defined	Not defined
CTComplete	PARAMETERS	PARAMETERS	Not defined	Not defined
a) "Not defined" means that this primitive is not defined.				
b) "–" means no parameters.				
c) Only applicable in case of transfer with secondary call.				

11.5.2 Primitive definition

The definition of these primitives is as follows:

- a) The CTSetup.indication primitive is used to notify about setup request.
- b) The CTSetup.response primitive is used to indicate acceptance of the call.
- c) The CTUpdate.request primitive is used to request update of identification information.
- d) The CTUpdate.indication primitive is used to notify about updated identification information.
- e) The CTIdentify.indication primitive is used to request a call identification.
- f) The CTIdentify.response primitive is used to give response to identify request.
- g) The CTAbandon.indication primitive is used to notify about unsuccessful transfer attempt.
- h) The SubaddressTransfer.request primitive is used to request transmission of subaddress to the TRDSE.
- i) The SubaddressTransfer.indication primitive is used to notify TRDSE subaddress to the TRTSE user.
- j) The CTComplete.request primitive may be used by a GKs to request sending of call transfer information to the transferred-to user.
- k) The CTComplete.indication primitive is used to indicate call transfer information to the transferred-to endpoint.

11.5.3 Parameter definition

CTSetup.ind parameters

- callIdentity: identifier of secondary call;
- transferringNumber: address of user A (optional);
- extension: Placeholder for vendor-specific extensions.

SubaddressTransfer.xxx parameters

- redirectionSubaddress: Party Subaddress;
- extension: Placeholder for vendor-specific extensions.

CTUpdate.xxx parameters

- redirectionNumber: address of party C;
- redirectionInfo: name information of party B or C;
- basicCallInfoElements: H.225 Information element (optional);
- extension: Placeholder for vendor-specific extensions.

CTComplete.xxx parameters

- redirectionNumber: address of new connected party (party B);
- basicCallInfoElements: H.225 Information element (optional);
- redirectionInfo: name information of new connected party (party B);
- callStatus: reflects the H.225 call state of the new connected party;
- extension: Placeholder for vendor-specific extensions.

11.5.4 States

The following states are used to specify the allowed sequence of primitives between the TRTSE and TRTSE user.

CT-Idle

Supplementary service Call Transfer is not operating.

CT-Await-Setup

A callTransferIdentify return result APDU has been sent to the TRGSE. This state is used during transfer by rerouting.

11.6 Peer-to-peer communication for transfer with rerouting

11.6.1 Messages

The FACILITY message may be used to request or acknowledge a supplementary service. For more information, see Recommendation H.450.1.

The SETUP message is sent by a calling entity to indicate its desire to set up a connection to a called entity. For more information, see 7.3.11/H.225.0.

The CONNECT message is sent by the called entity to the calling entity to indicate acceptance of the call by the called entity. For more information, see 7.3.3/H.225.0.

The RELEASE COMPLETE message is sent by a terminal to indicate release of the call if the reliable call signalling channel is open. For more information, see 7.3.10/H.225.0.

11.6.2 Timers

CT-T1 – Timer CT-T1 shall operate at the TRGSE during state CT-Await-Identify-Response. Its purpose is to protect against the absence of a response to the CTIdentify.request.

CT-T2 – Timer CT-T2 shall operate at the TRTSE during state CT-Await-Setup. Its purpose is to protect against failure of completion of the call transfer operation.

CT-T3 – Timer CT-T3 shall operate at the TRGSE during state CT-Await-Initiate-Response. Its purpose is to protect against failure to establish the new connection.

CT-T4 – Timer CT-T4 may optionally operate at the TRDSE during state CT-Await-Setup-Response. Its purpose is to protect against failure to establish the new connection.

NOTE – Default value ranges for timers are for further study.

11.6.3 Counters

No counter variables are used for the call transfer supplementary service.

11.6.4 Message flow examples

11.6.4.1 Transfer with rerouting (without consultation)

Description from user point of view:

User A (transferring party): communicating with B; selects C; request B to connect to C (Media inherited); transfer accepted; idle.

User B (transferred party): communicating with A; receives request of transfer; confirms media; communicating with C.

User C (transferred-to party): idle; receives notification of incoming call; accepts call; confirms media; communicating with B.

See Table 4.

11.6.4.2 Failing case of transfer with rerouting (without Secondary call)

Scenario Description from user point of view:

User A (transferring party): communicating with B; selects C; request B to connect to C (media inherited); unsuccessful transfer; communicating with B.

User B (transferred party): communicating with A; receives request for transfer; unsuccessful transfer; communicating with A.

User C (transferred to party): idle; unsuccessful call establishment from B; idle.

See Table 5.

11.6.4.3 Transfer with secondary call (consultation)

Description from user point of view:

User A (transferring party): communicating with B; B on hold (optional); user selects C and media; requests connection to C; consultation active; request B to connect to C; transfer accepted; idle.

User B (transferred party): communicating with A; receives request for transfer; confirms media; communicating with C.

User C (transferred-to party): idle; receives notification of incoming call; accepts call; confirms media; communicating with A; receives notification of incoming call; accepts call; confirms media; communicating with B.

See Table 6.




Table 4/H.450.2 – Transfer with rerouting (without consultation)

Row No.	User/Application action	a) Primitive b) State c) Timer	Note	H.225, H.323, H.245, H.450	Note	a) Primitive b) State c) Timer	User/Application action
1	MM Terminal A			Network		MM Terminal B/Gateway	
2	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists UDP path for audio open						
3	Request terminal B to transfer call to C enter state: CT-Await-Initiate-Response	a) Transferring/TRGSE <CTInitiate.req> b) CT-Await_Initiate-Response c) CT-T3		H.225 <FACILITY> → H4501SupplementaryService APDU: invoke callTransferInitiate reroutingNumber=address C callIdentity (=empty)		a) Transferred TRDSE <CTInitiate.ind> b) CT-Idle c) None	Receive transfer_request
4	MM Terminal B/Gateway			Network		MM Terminal C	
5	<i>Establish TCP path for H.225 call signalling</i>						
6	Request for call establishment to User C Select Media User C address	a) Transferred TRDSE <CTSetup.req> <CTUpdate.req> b) CT-Await_Setup_Response c) T4 (optional)		H.225 <SETUP> → H4501SupplementaryService APDU: invoke – callTransferSetup – callTransferUpdate – redirectionNumber		a) Transferred to TRTSE <CTSetup.ind> <CTUpdate.ind_ind> b) CT-Idle c) None	Receive H.225 setup.indication
7	Indication to Appl/User B of MM Terminal C general availability	a) Transferred TRDSE <CTSetup.conf> <CTUpdate.ind> b) CT-Idle c) None		H.225 <CONNECT> ← H4501SupplementaryService APDU: returnResult – callTransferSetup invoke – callTransferUpdate – redirectionNumber		a) Transferred to TRTSE <CTSetup.resp> <CTUpdate.req> b) CT-Idle c) None	Call accept by user C or by Application C
8	MM Terminal A			Network		MM Terminal B/Gateway	
9	<i>Close logical channels</i>						
10	Receive H.225 release.indication	a) Transferring TRGSE <CTInitiate_conf> b) CT-Idle c) None		H.225 <RELEASE COMPLETE> ← H4501SupplementaryService APDU: returnResult – callTransferInitiate		a) Transferred TRDSE <CTInitiate_resp> b) CT-Idle c) None	Release call to TE A
11	MM Terminal B/Gateway			Network		MM Terminal C	
12	<i>Exchange terminal capabilities open logical channels (H.245)</i>						

Table 5/H.450.2 – Failing case of transfer with rerouting (without secondary call)

Row No.	User/Application action	a) Primitive b) State c) Timer	Note	H.225, H.323, H.245, H.450	Note	a) Primitive b) State c) Timer	User/Application action
13	MM Terminal A			Network		MM Terminal B/Gateway	
14	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists UDP path for audio open						
15	Request terminal B to transfer call to C enter state: CT-Await-Initiate-Response	a) Transferring TRGSE <CT-Initiate-req> b) CT-Await-Initiate-Response c) CT-T3		H.225 <FACILITY> → H4501SupplementaryService APDU: invoke – callTransferInitiate – reroutingNumber=address C – callIdentity (=empty)		a) Transferred TRDSE <CTInitiate.ind> b) CT-Idle c) None	Receive transfer_request
16	MM Terminal A/Gateway			Network		MM Terminal C	
17	Request for call establishment to TE C	<i>Establishment of basic call fails</i>					
18	MM Terminal A			Network		MM Terminal B/Gateway	
19		a) Transfer ring TRGSE <CTInitiate_resp_rej> b) CT-idle c) None		H.225 <FACILITY> ← H4501SupplementaryService APDU: returnError – callTransferInitiate		a) Transferred TRDSE <CTInitiate.conf_rej> b) CT-idle c) None	
20	A communicating with B						

Table 6/H.450.2 – Transfer with secondary call (consultation)

Row No.	User/Application action	a) Primitive b) State c) Timer	Note	H.225, H.323, H.245, H.450	Note	a) Primitive b) State c) Timer	User/Application action
21	MM Terminal A			Network		MM Terminal B	
22	Active Basic Call between TE A and TE B Capabilities exchanged H.225 connection still exists UDP path for audio open						
23	MM Terminal A			Network		MM Terminal C	
24	<i>Establishment of basic call</i>						
25	<i>Consultation active</i>						
26	Identification of TE C	a) Transferring TRGSE <CTIdentify.req> b) CT-Await_Identify-Response c) CT-T1		H.225 <FACILITY>  H4501SupplementaryService APDU: invoke – callTransferIdentify	1	a) Transferred-to TRTSE <CTIdentify.ind> b) CT-Idle c) None	
27	Receive Identification	a) Transferring TRGSE <CTIdentify.conf> b) CT-Idle c) None		H.225 <FACILITY>  H4501SupplementaryService APDU: returnResult – callTransferIdentify – callIdentity – reroutingNumber=address C	2	a) Transferred-to TRTSE <CTIdentify.resp> b) CT-Await-Setup c) CT-T2	Determine ability of CT enter state: CT-Await-Setup
28	MM Terminal A			Network		MM Terminal B/Gateway	
29	<i>Transfer with rerouting (Table 4 with non-empty callIdentity)</i>						
30	MM Terminal A			Network		MM Terminal C	
31	<i>Close logical channels</i>						
32	Receive H.225 release.indication	a) Transferring TRGSE <Release.ind> b) CT-Idle c) None		H.225 <RELEASE COMPLETE> 		a) Transferred-to TRTSE <Release.req> b) CT-Idle c) None	
1	Optional						
2	Optional						

12 Operations in support of SS-CT

The following operations defined in Abstract Syntax Notation No. 1 shall apply:

Call-Transfer-Operations

{itu-t recommendation h 450 2 version1(0) call-transfer-operations(0)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS OPERATION, ERROR FROM Remote-Operations-Information-Objects
 {joint-iso-itu-t remote-operations(4)
 informationObjects(5) version1(0)}
EXTENSION, Extension{} FROM
 Manufacturer-specific-service-extension-definition
 {itu-t recommendation h 450 1 version1(0) msi-definition(18)}
notAvailable, supplementaryServiceInteractionNotAllowed,
invalidCallState FROM H4501-General-Error-Listf
 {itu-t recommendation h 450 1 version1(0) general-error-list(1)}
EndpointAddress, PartySubaddress FROM Addressing-Data-Elements
 {itu-t recommendation h 450 1 version1(0)
 addressing-data-elements(9)}
NonStandardParameter FROM H323-MESSAGES -- see H.225.0
H225InformationElement FROM H225-generic-parameters-definition
 {itu-t recommendation h 450 1 version1(0)
 h225-generic-parameters(6)};

H323CallTransferOperations OPERATION ::=

{ callTransferIdentify | callTransferAbandon | callTransferInitiate |
 callTransferSetup | callTransferUpdate | subaddressTransfer |
 callTransferComplete | callTransferActive }

callTransferIdentify OPERATION ::=

-- sent from transferring endpoint to transferred-to endpoint

ARGUMENT DummyArg OPTIONAL TRUE
RESULT CTIdentifyRes
ERRORS { notAvailable | invalidCallState |
 supplementaryServiceInteractionNotAllowed | unspecified }
CODE local:7

callTransferAbandon OPERATION ::=

-- sent from transferring to transferred-to endpoint

ARGUMENT DummyArg OPTIONAL TRUE
RETURN RESULT FALSE
ALWAYS RESPONDS FALSE
CODE local:8

callTransferInitiate OPERATION ::=

-- sent from transferring to transferred endpoint

ARGUMENT CTInitiateArg
RESULT DummyRes OPTIONAL TRUE
ERRORS { notAvailable | invalidCallState | invalidReroutingNumber |
 unrecognizedCallIdentity | establishmentFailure |
 supplementaryServiceInteractionNotAllowed | unspecified }
CODE local:9

```

callTransferSetup      OPERATION ::=
{
  ARGUMENT      CTSetupArg
  RESULT        DummyRes OPTIONAL TRUE
  ERRORS        { notAvailable | invalidCallState | invalidReroutingNumber |
                 unrecognizedCallIdentity | unspecified |
                 supplementaryServiceInteractionNotAllowed }
  CODE          local:10
}

```

```

callTransferUpdate    OPERATION ::=
{-- may be sent from transferred to transferred-to endpoint (in SETUP) and
-- vice versa (in CONNECT)
  ARGUMENT      CTUpdateArg
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE          local:13
}

```

```

subaddressTransfer    OPERATION ::=
{-- may be sent from transferred to the transferred-to endpoint and vice versa
  ARGUMENT      SubaddressTransferArg
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE          local:14
}

```

```

callTransferComplete OPERATION ::=
{-- sent from GK to the transferred endpoint, if GK performs call transfer
-- (see 10.6.1).
  ARGUMENT      CTCompleteArg
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE          local:12
}

```

```

callTransferActive OPERATION ::=
{-- sent from gatekeeper to the transferred endpoint, in case of transfer ringing
-- after transferred-to endpoint has answered the call (only in case of
-- transfer performed by gatekeeper)
  ARGUMENT      CTActiveArg
  RETURN RESULT FALSE
  ALWAYS RESPONDS FALSE
  CODE          local:11
}

```

```

DummyArg ::= CHOICE
{
  extensionSeq      ExtensionSeq,
  nonStandardData   NonStandardParameter }

```

```

DummyRes ::= CHOICE
{
  extensionSeq      ExtensionSeq,
  nonStandardData   NonStandardParameter }

```

```

CTInitiateArg ::= SEQUENCE
{
  callIdentity      CallIdentity,
  reroutingNumber   EndpointAddress,
  argumentExtension CHOICE
  {extensionSeq     ExtensionSeq,

```

```

        nonStandardData          NonStandardParameter }    OPTIONAL,
        ...
    }

CTSetupArg ::= SEQUENCE
{
    callIdentity                  CallIdentity,
    transferringNumber            EndpointAddress OPTIONAL,
    argumentExtension             CHOICE
    {extensionSeq                ExtensionSeq,
    nonStandardData              NonStandardParameter }    OPTIONAL,
    ...
}

CTIdentifyRes ::= SEQUENCE
{
    callIdentity                  CallIdentity,
    reroutingNumber              EndpointAddress,
    resultExtension              CHOICE
    {extensionSeq                ExtensionSeq,
    nonStandardData              NonStandardParameter }    OPTIONAL,
    ...
}

CTUpdateArg ::= SEQUENCE
{
    redirectionNumber            EndpointAddress,
    redirectionInfo              BMPString (SIZE(1..128)) OPTIONAL,
    basicCallInfoElements        H225InformationElement    OPTIONAL,
    -- Information element
    -- Progress indicator may be conveyed
    argumentExtension            CHOICE
    {extensionSeq                ExtensionSeq,
    nonStandardData              NonStandardParameter }    OPTIONAL,
    ...
}

SubaddressTransferArg ::= SEQUENCE
{
    redirectionSubaddress        PartySubaddress,
    argumentExtension            CHOICE
    {extensionSeq                ExtensionSeq,
    nonStandardData              NonStandardParameter }    OPTIONAL,
    ...
}

CTCompleteArg ::= SEQUENCE
{
    endDesignation               EndDesignation,
    redirectionNumber            EndpointAddress,
    -- number of new connected user
    basicCallInfoElements        H225InformationElement    OPTIONAL,
    -- Information element Progress indicator may be conveyed
    redirectionInfo              BMPString (SIZE(1..128)) OPTIONAL,
    callStatus                   CallStatus DEFAULT answered,
    argumentExtension            CHOICE
    {extensionSeq                ExtensionSeq,
    nonStandardData              NonStandardParameter }    OPTIONAL,
    ...
}

```

```

EndDesignation ::= ENUMERATED
    {
        primaryEnd(0),
        secondaryEnd(1),
        ...
    }

CallStatus ::= ENUMERATED
    {
        answered(0),
        alerting(1),
        ...
    }

CTActiveArg ::= SEQUENCE
    {
        connectedAddress EndpointAddress,
        -- number of new connected user
        basicCallInfoElements H225InformationElement OPTIONAL,
        -- Information element Progress indicator may be conveyed
        connectedInfo BMPString (SIZE(1..128)) OPTIONAL,
        argumentExtension CHOICE
            {
                extensionSeq ExtensionSeq,
                nonStandardData NonStandardParameter } OPTIONAL,
            }
        ...
    }

CallIdentity ::= NumericString (SIZE(0..4))
    -- size 0 means 'empty'

ExtensionSeq ::= SEQUENCE OF Extension{{ExtensionSet}}

ExtensionSet EXTENSION ::= {...}
    -- Actual values defined by individual manufacturers

unspecified ERROR ::=
    {
        PARAMETER CHOICE
            {extension Extension{{ExtensionSet}},
            nonStandardNonStandardParameter
            }
        CODE local:1008
    }

invalidReroutingNumber ERROR ::=
    {
        -- used when establishment of the new connection fails
        -- because the reroutingNumber is not a valid alias address
        CODE local:1004
    }

unrecognizedCallIdentity ERROR ::=
    {
        -- used when establishment of the new connection fails because it
        -- could not be associated with a SS-CT entity at the
        -- transferred-to endpoint
        CODE local:1005
    }

```

```

establishmentFailure          ERROR    ::=
{
  -- used when establishment of the new connection fails and
  -- no other error applies
  CODE          local:1006
}
END              -- of Call-Transfer-Operations

```

13 Specification and Description Language (SDL) diagrams

The procedures for call transfer signalling entities are described in SDL form in Figures 21 to 23.

NOTE 1 – In case of conflict with text, the text shall take precedence.

NOTE 2 – The SDLs only show call transfer messages transported on H.225.0 reliable connection. H.245 procedures (e.g. terminal capability exchange, master/slave determination, opening and closing of logical channels, etc.) are not shown.

NOTE 3 – Specific gatekeeper SDLs for call transfer using call transfer by join as given in 10.6.1 are not provided.

NOTE 4 – Within this clause, "ct" is used as an abbreviation for the term "callTransfer" as part of call transfer operation names (Example: ctInitiate.Invoke refers to callTransferInitiate invoke APDU).

13.1 Call Transfer block diagram

Figure 19 shows the block of supplementary service Call Transfer and its interfaces to other blocks. The Substructure of this block is given in Figure 20.

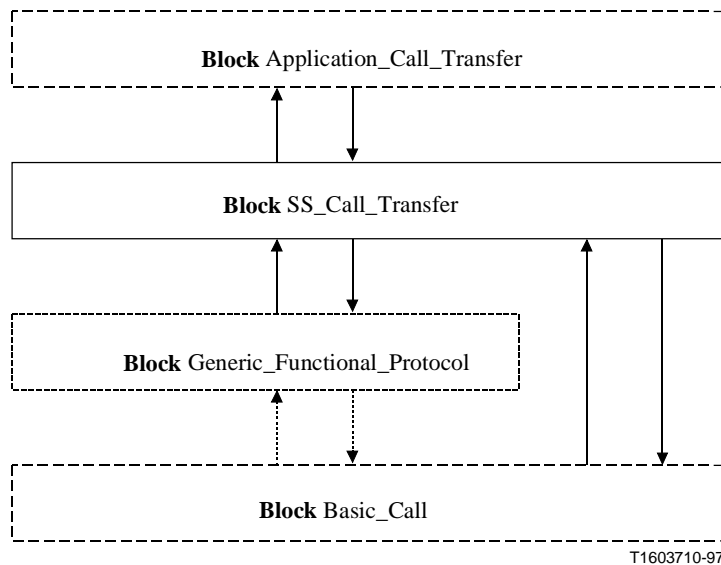
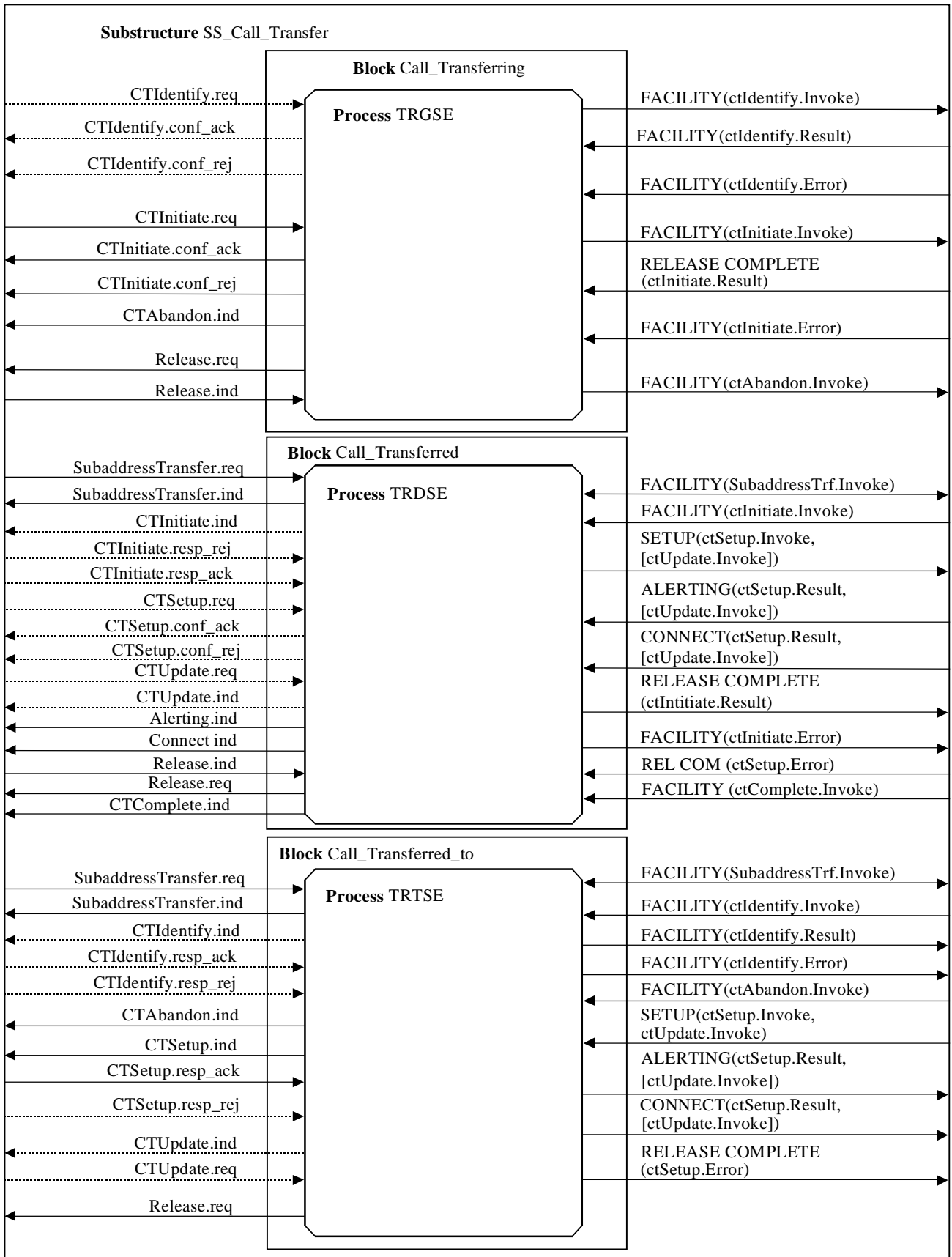


Figure 19/H.450.2 – Call Transfer block diagram

13.2 Call transfer substructure diagram

The Substructure SS_Call_Transfer consists of the processes TRGSE, TRDSE, and TSTSE. The interface signals to these blocks are shown in the Figure 20.



T1603720-97

Figure 20/H.450.2 – Call Transfer substructure diagram

13.3 Transferring signalling entity SDLs

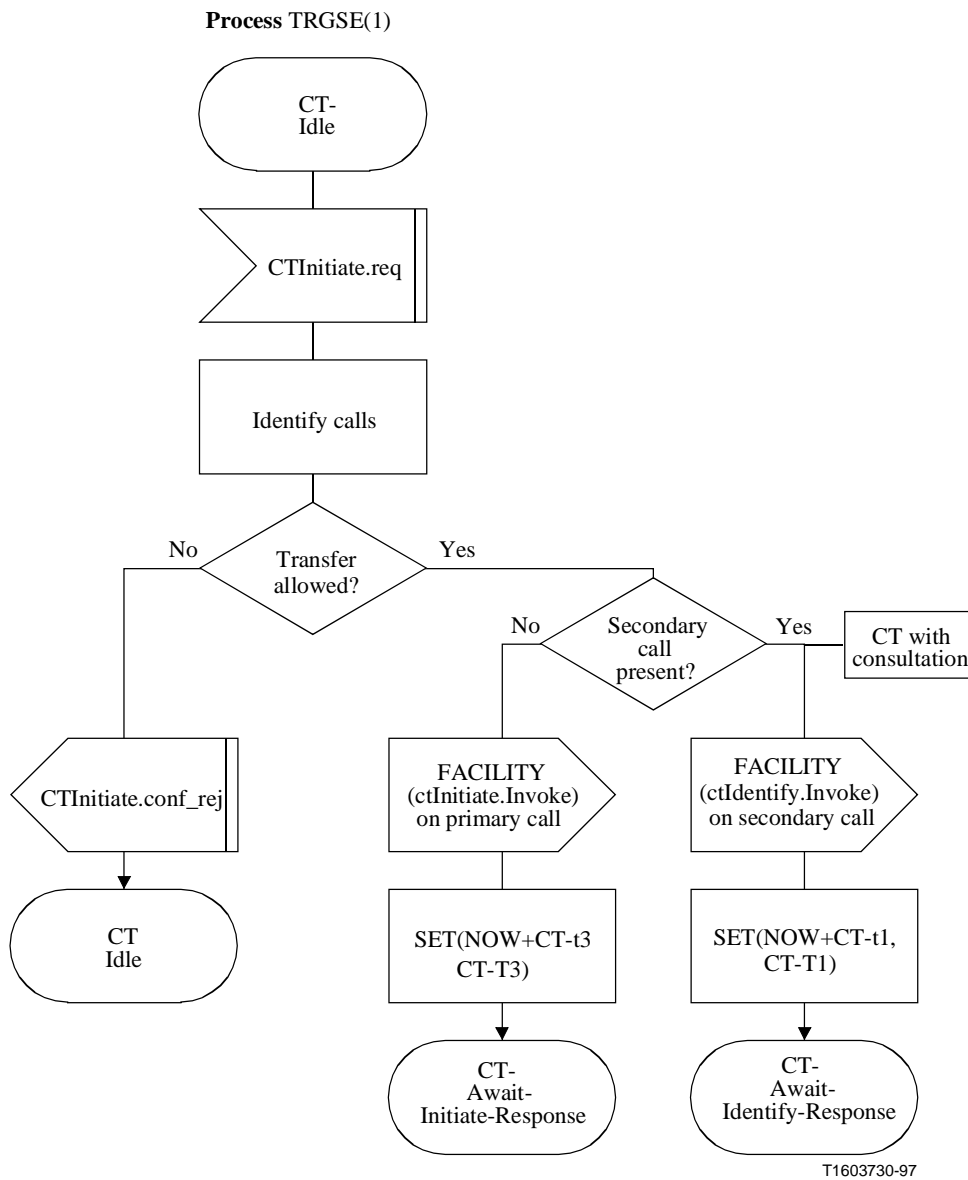
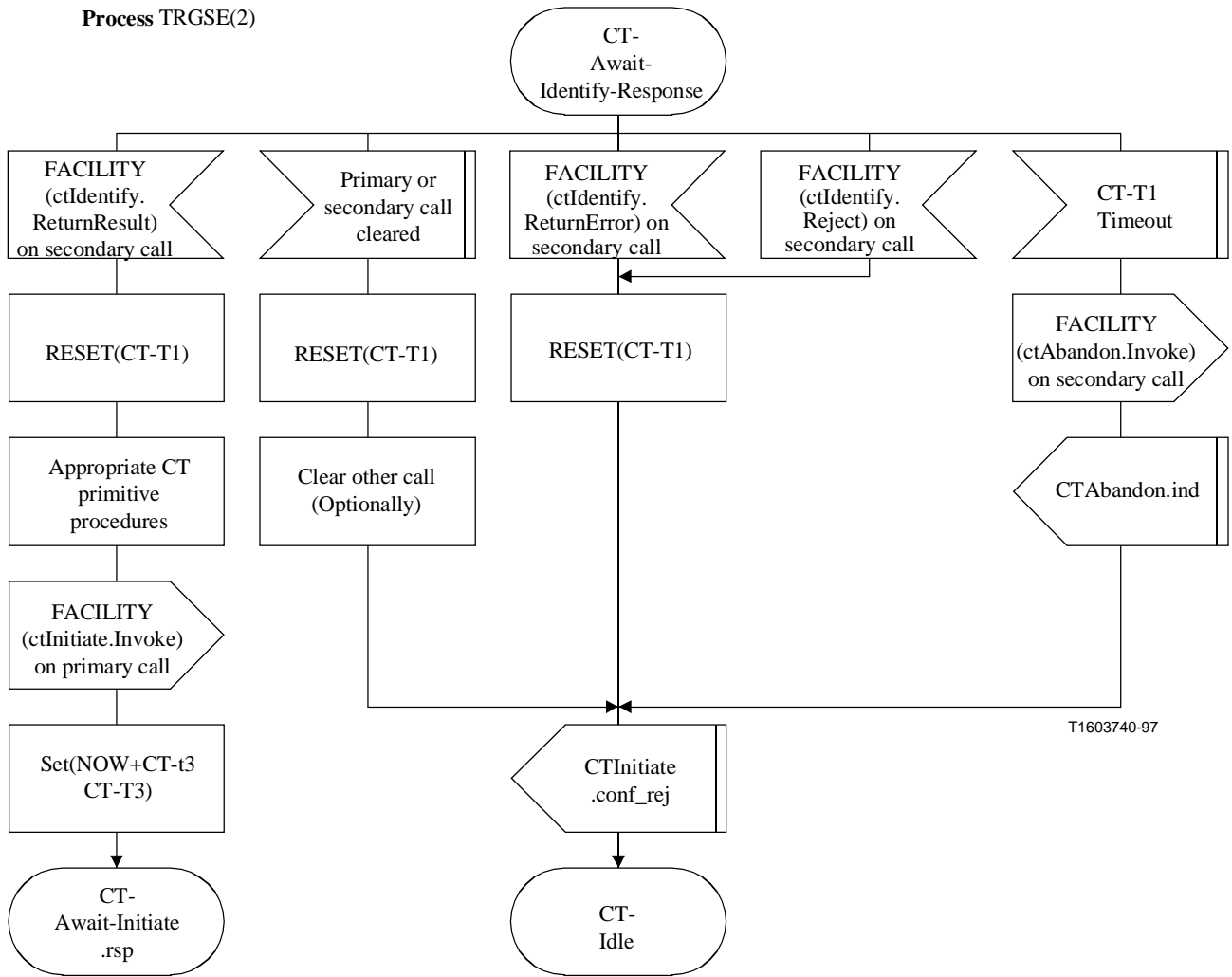


Figure 21/H.450.2 (sheet 1 of 3) – Transferring signalling entity

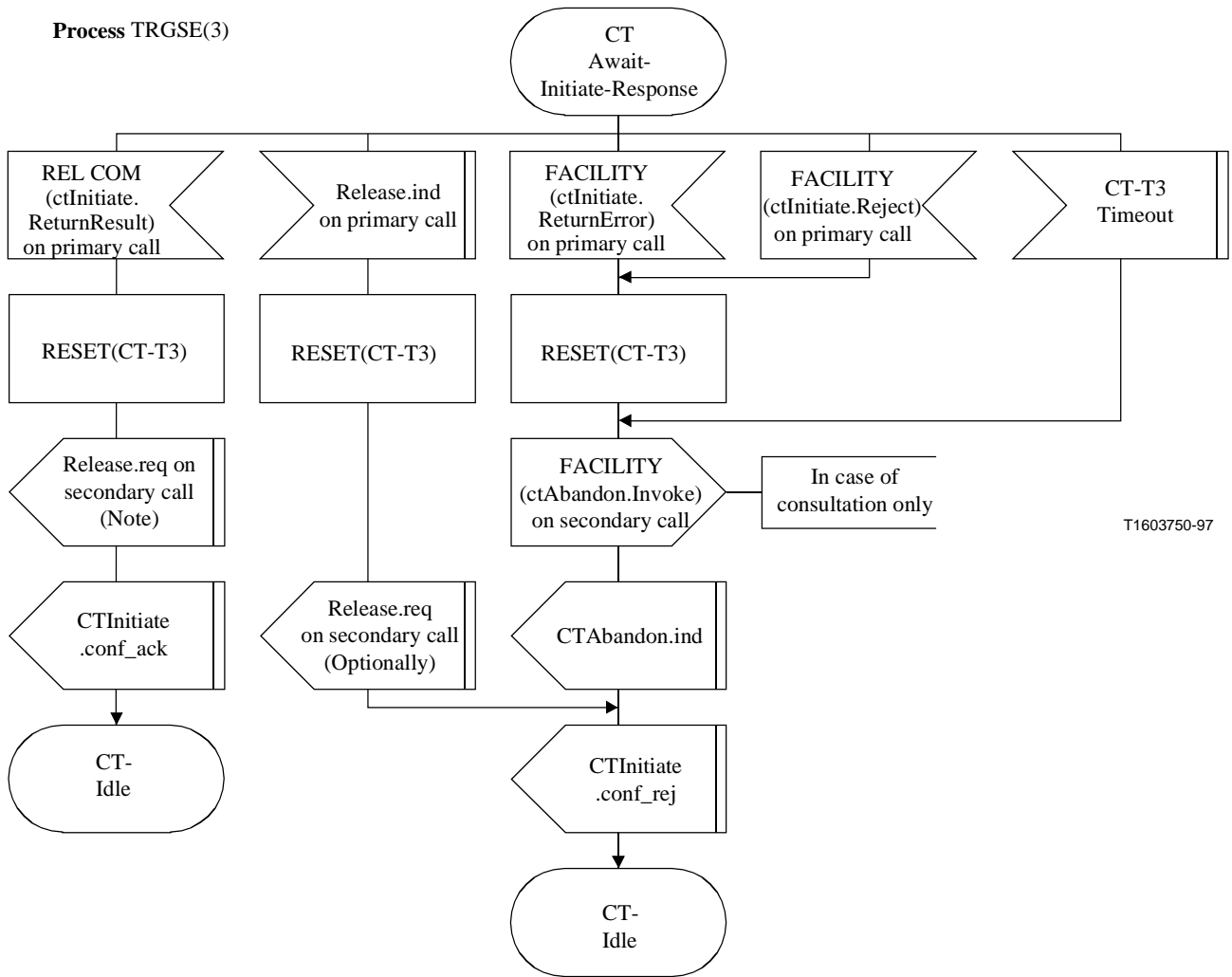
Process TRGSE(2)



T1603740-97

Figure 21/H.450.2 (sheet 2 of 3) – Transferring signalling entity

Process TRGSE(3)



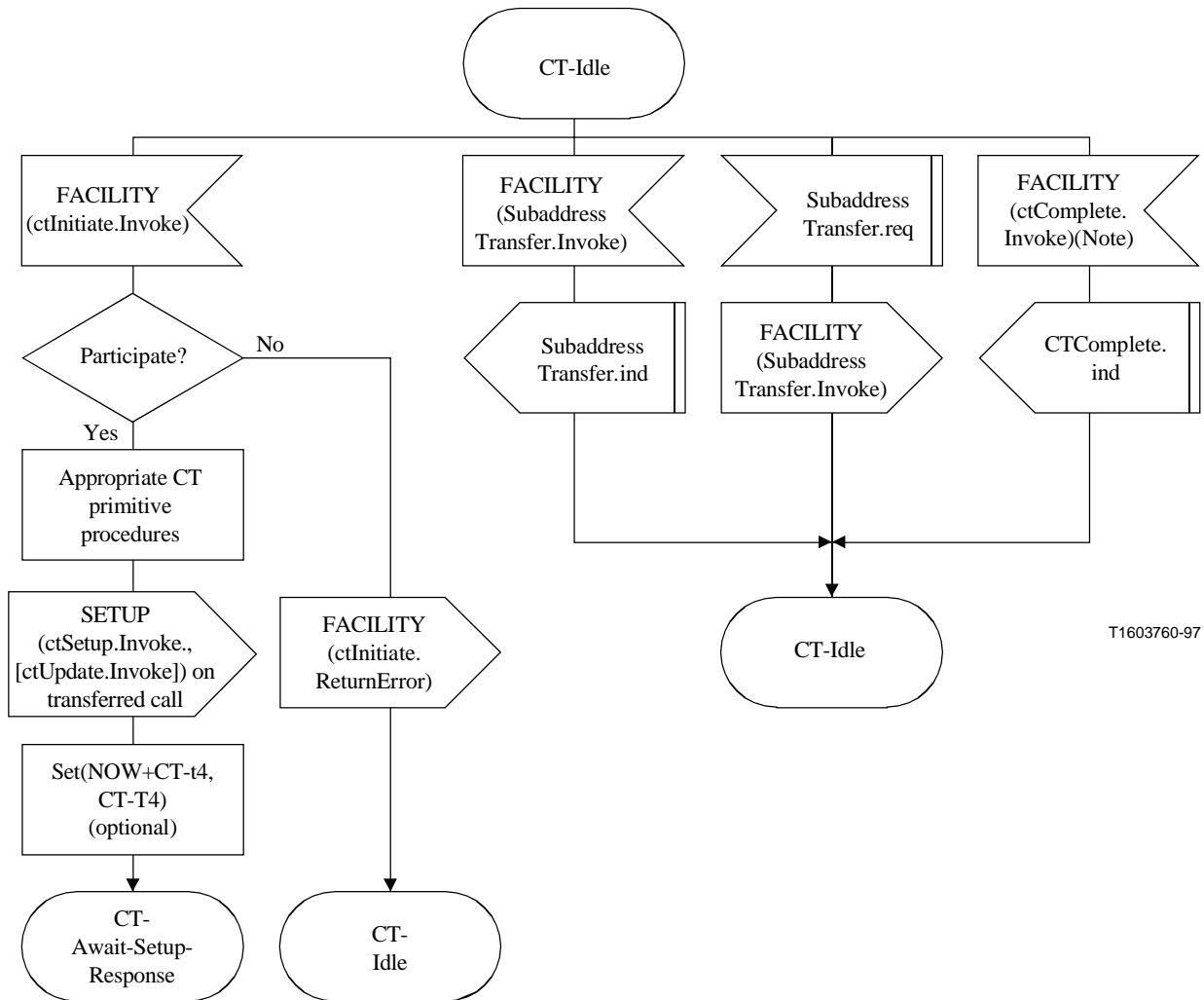
T1603750-97

NOTE – Will cause clearing of secondary call if not cleared already.

Figure 21/H.450.2 (sheet 3 of 3) – Transferring signalling entity

13.4 Transferred signalling entity SDLs

Process TRDSE(1)

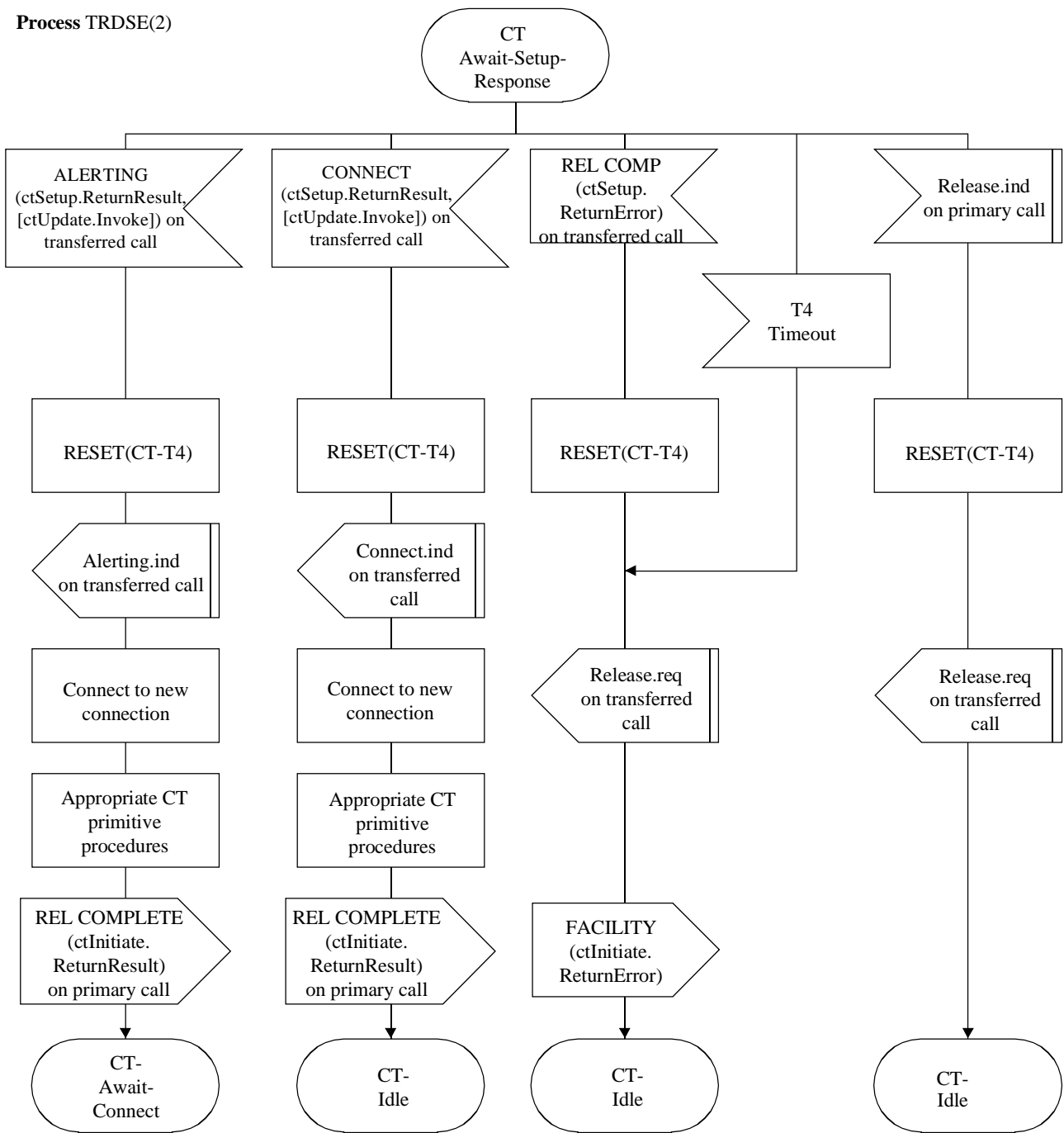


T1603760-97

NOTE – May be received from GK in GK routed mode, if call transfer is performed by GK. Subsequently, a ctActive invoke APDU may be received in case of call transfer alerting.

Figure 22/H.450.2 (sheet 1 of 3) – Primary signalling entity

Process TRDSE(2)



T1603770-97

NOTE – In case of call transfer performed by gatekeeper, a ctComplete inv.apdu shall be sent from the gatekeeper to the transferred endpoint after receiving ctSetup return result apdu.

Figure 22/H.450.2 (sheet 2 of 3) – Primary signalling entity

Process TRDSE(3)

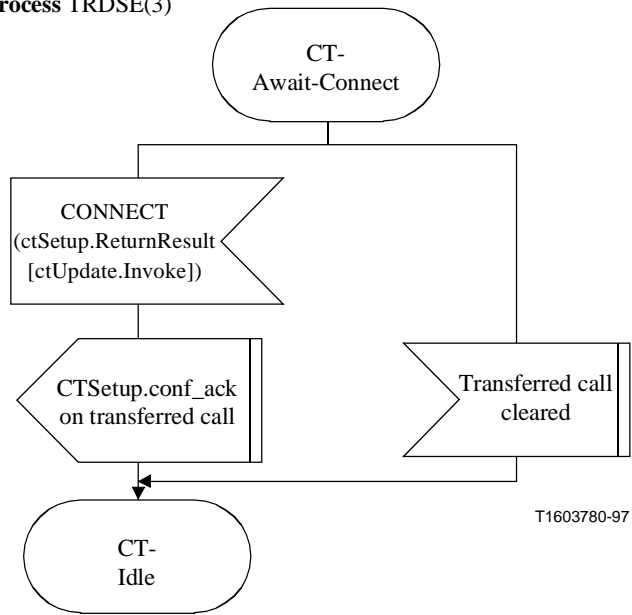


Figure 22/H.450.2 (sheet 3 of 3) – Primary signalling entity

13.5 Transferred-to signalling entity SDLs

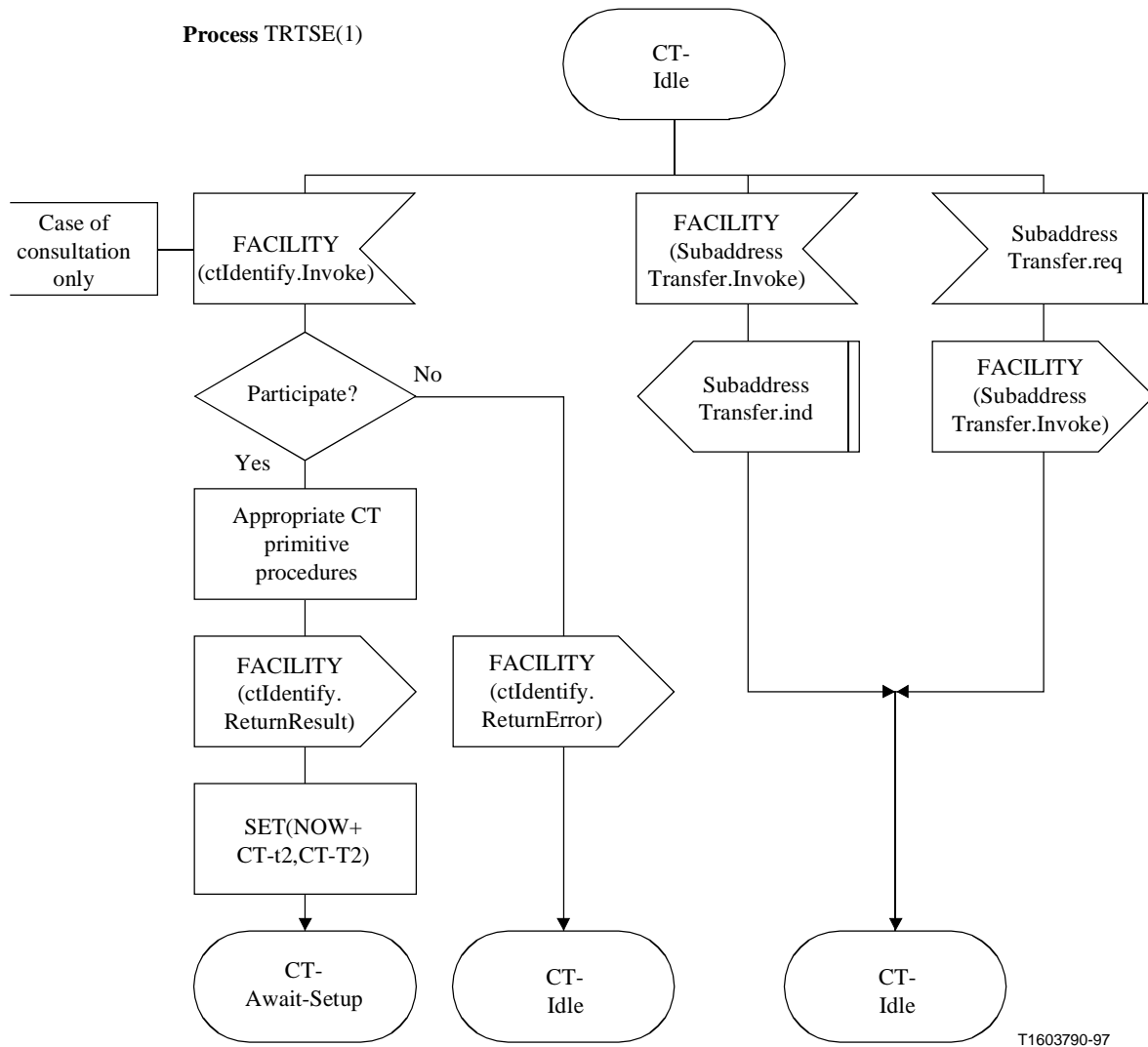


Figure 23/H.450.2 (sheet 1 of 3) – Secondary signalling entity

Process TRTSE(2)

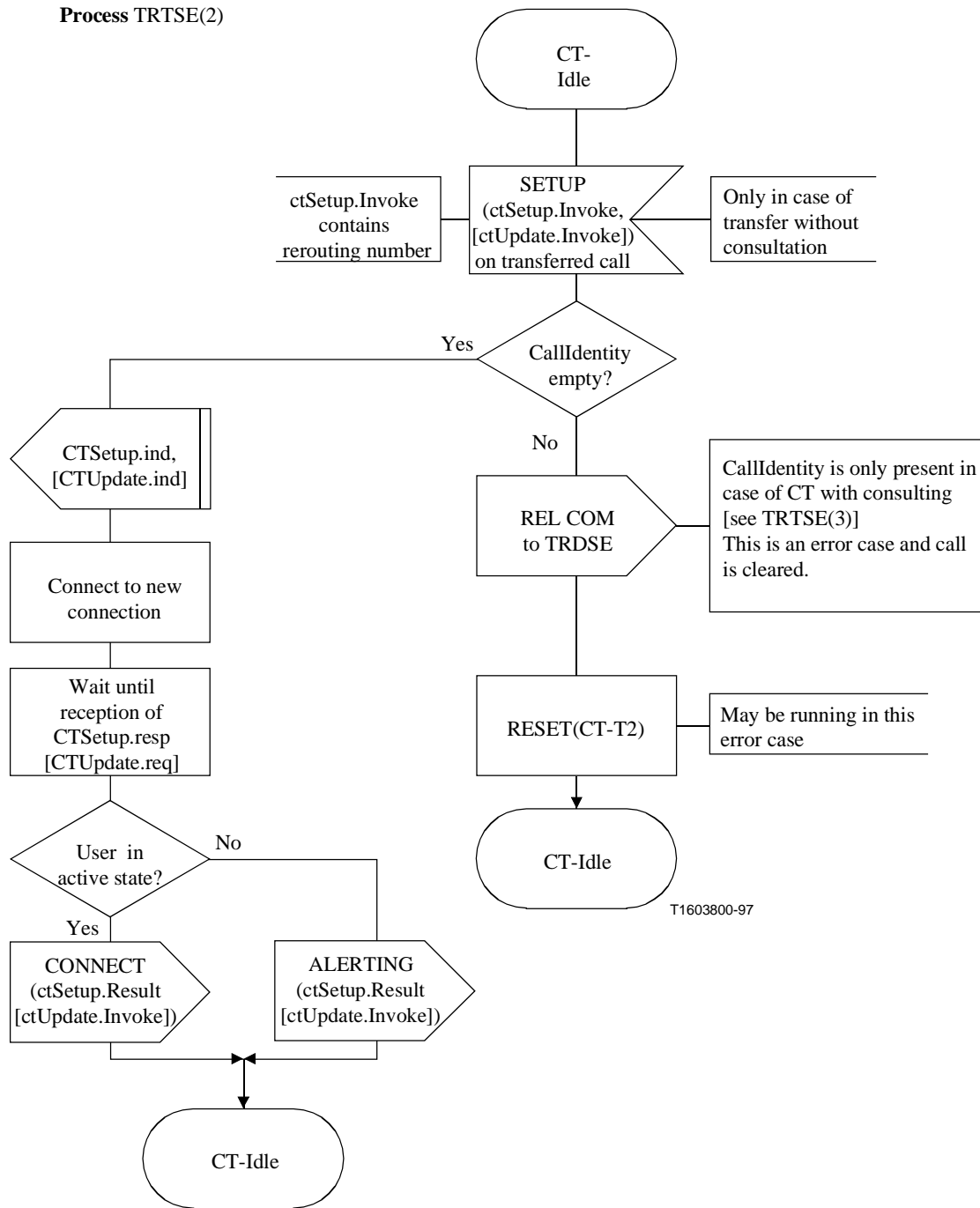


Figure 23/H.450.2 (sheet 2 of 3) – Secondary signalling entity

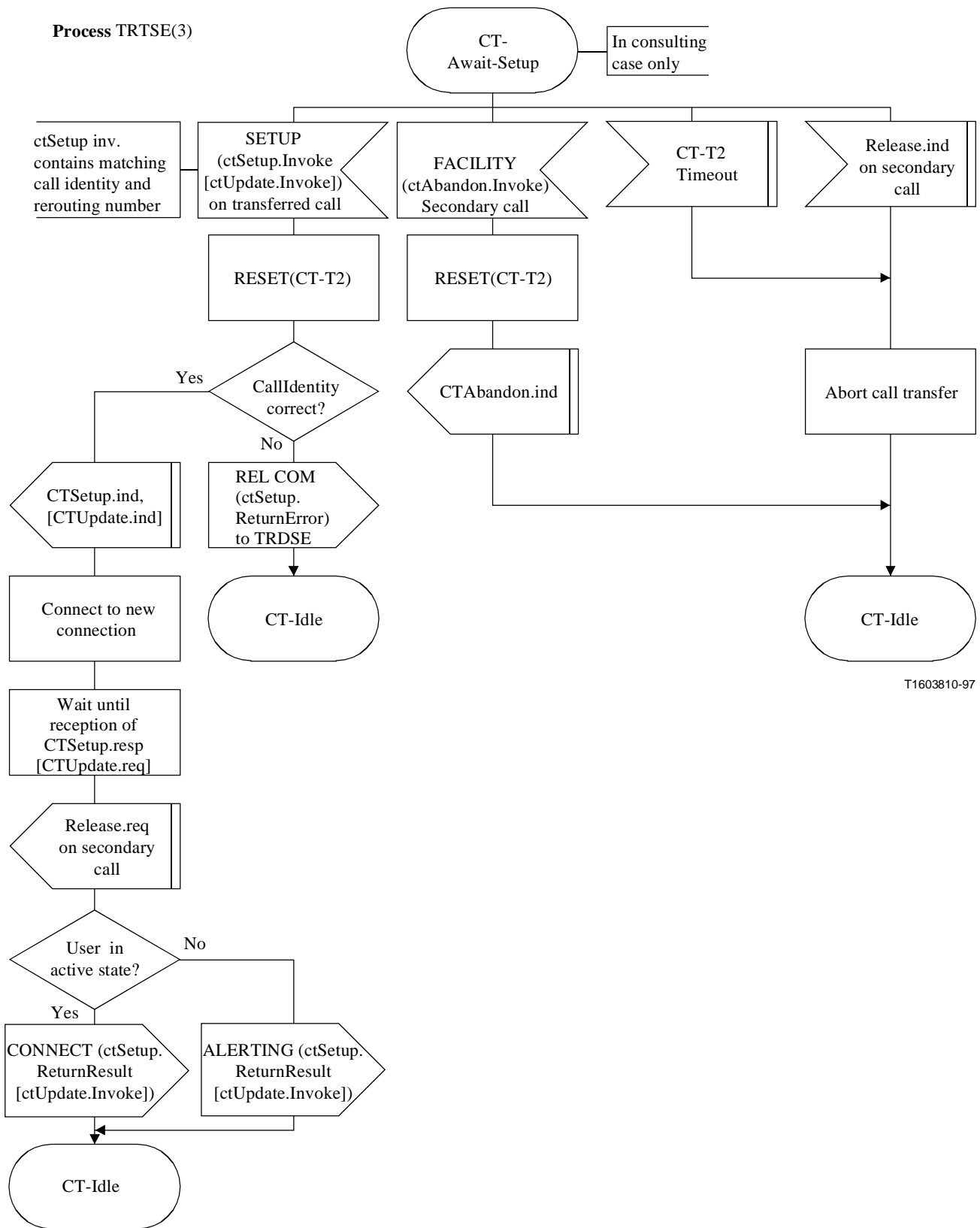


Figure 23/H.450.2 (sheet 3 of 3) – Secondary signalling entity

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