rev 0.15

ECMA - Standardizing Information and Communication Systems

Working paper by ECMA TC32-TG11

TITLE: **Notes on *CSTA in IP Telecommunications***

EDITORS: Neal Oliver (Dialogic) , Tom Miller (Siemens), J. Dave Smith (Siemens)

DATE: **15 May, 2000**

Abstract: This working paper evaluates how CSTA may also be used in IPT based networks. It also identifies possible areas for further standardisation activity. The paper is intended to indicate the work of ECMA TC32-TG11 in this area and invite comments on the proposals.

Table of Contents

1. Introduction

CSTA is the ECMA and ISO/IEC standard for computer telecommunications integration (CTI). CSTA is already widely used to integrate enterprise computing and telephony activity, for applications ranging from call centers to desktops. Now that the telecom and the IP Telephony (IPT) world are converging, it is valuable to extend these services to the IPT world, to leverage investments in applications already operating in the telephony environment into IP and hybrid environments.

This working paper evaluates how CSTA may be used in IPT based networks and determines what changes are required to CSTA and to IPT standards to support their joint use. It also identifies possible areas for further standardisation activity. The paper is intended to indicate the work of ECMA TC32-TG11 in this area and invite comments on the proposals.

The approach of this paper is to develop a mapping from CSTA to an IPT system, and to show the interoperation of CSTA and an IPT system in a variety of common call control scenarios.

Although, as reflected in this paper, the work is initially focusing on H.323 IP systems, future work will provide additional scenarios demonstrating CSTA in other IP environments, such as SIP, MGCP, H.248 (Megaco), and packet cable[CABLELAB].

To ensure a common level of understanding Annex A gives a short introduction to CSTA whilst Annex B gives a similar introduction to H.323 (based upon the understanding of ECMA TC32-TG11)

2. A CSTA/H.323 Mapping

The summary of CSTA and H.323 in the annexes should make the following observation clear: with respect to CSTA, the H.323 recommendation describes the behaviour of a possible switching sub-domain; and with respect to H.323, CSTA describes services that can be carried out by appropriate behaviour of the entities in an H.323 system.

In order to show a mapping between CSTA and an IPT system, this working paper introduces components that interact with the IPT (e.g., H.323) system on behalf of the CSTA Computing Domain.

The following definitions apply to the CSTA/H.323 mapping:

Marshall: The Marshall provides a CSTA view of the IPT environment to the computing function (i.e., to applications). It implements the services and events defined by the CSTA Services specification, using the Deputies and their corresponding endpoints to represent CSTA Devices.

Deputy: an entity tightly coupled to an IPT endpoint and acting as an adaptor between the IPT endpoint and the Marshall.

CSTA-IPT Mapping: the interface between a Marshall and a Deputy developed in the course of the Working Paper development to explore the interoperability of CSTA with different IPT environments. The definition of this mapping in the Working Paper is a proof-by-construction, and is neither a Standard nor a Recommendation.

The following figure depicts the relationships of these components:

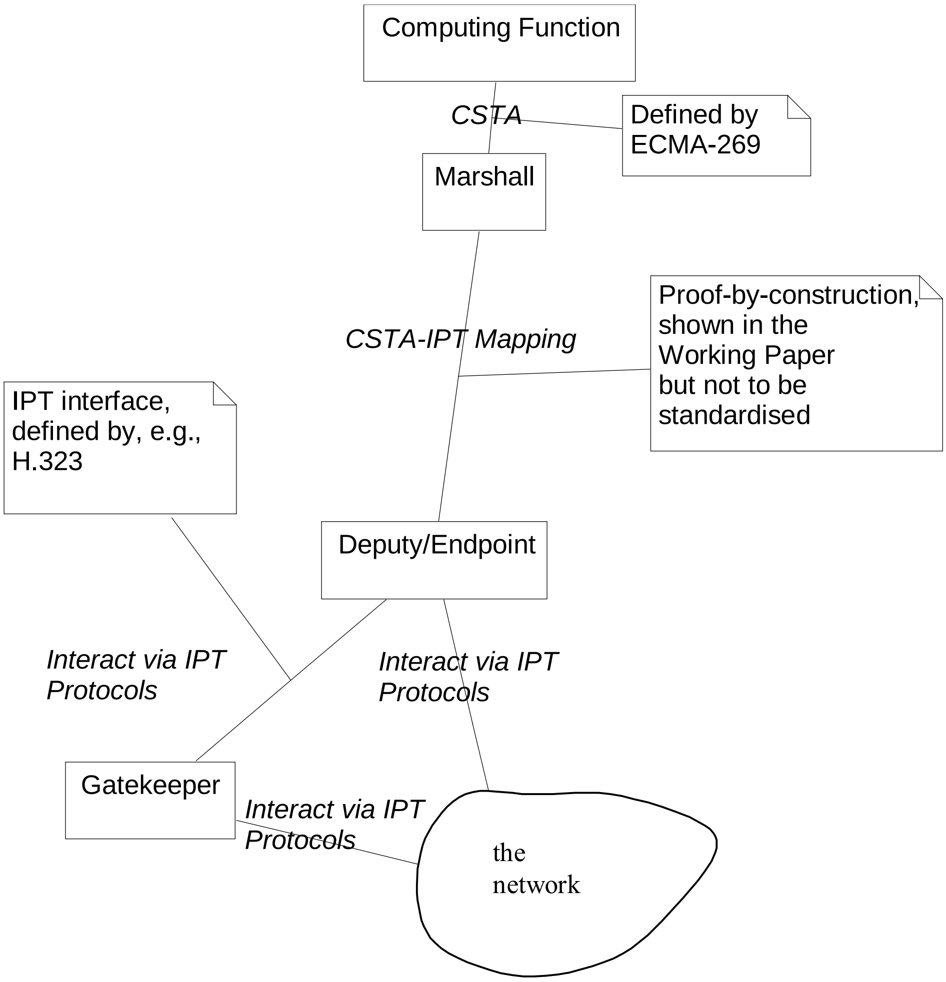
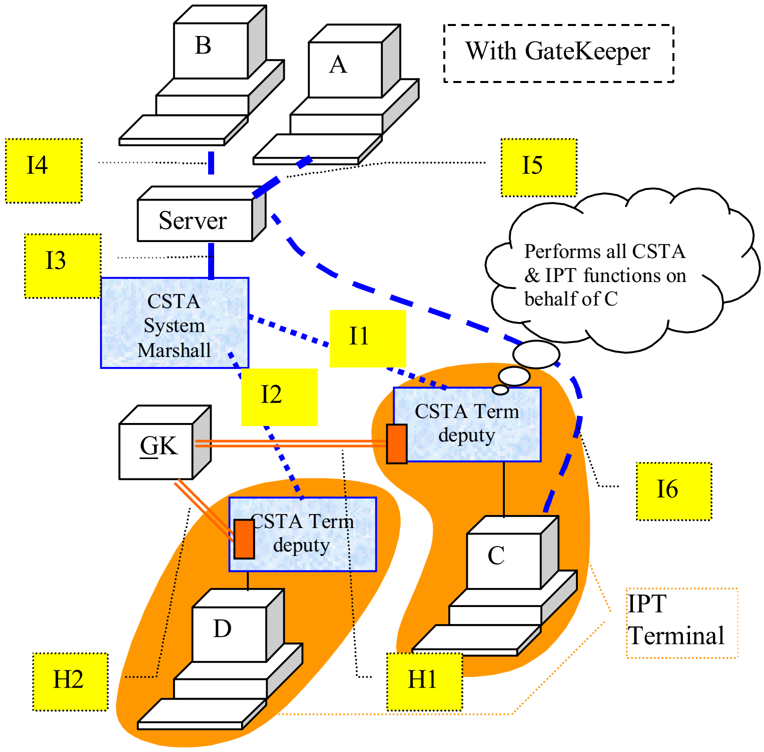


Figure 1 - Relationship of CSTA and IPT conceptual components

The following figure depicts an example CSTA/H.323 system in which a Marshall and Deputies appear together with H.323 components:



Key:
A: CSTA application
B: CSTA application
C: Combination IPT terminal and CSTA client
D: IPT terminal
GK: Gatekeeper
I6: API (from CSTA client to server)
I5: API
I4: API
I3: CSTA interface
I2: CSTA 'like' interface
I1: CSTA 'like' interface
H2: IPT interface
H1: IPT interface

Server is an optional part of the architecture (and is not defined by CSTA) which may be used to support multiple CSTA applications over the CSTA interface.

The IPT terminal (e.g. H.323 endpoint) is a combination of the CSTA deputy, with its IPT functionality, and the host PC

Figure 2 - Merged CSTA/IPT Environment

The addition of the CSTA System Marshall to the IPT environment is not sufficient to support CSTA system requirements. CSTA requires that third-party clients be able to establish calls on behalf of terminals; in the case of H.323 the protocols provide only for first-party signalling and control. Thus, an additional CSTA component must be associated with Terminals, whose function is to receive CSTA commands and convert them into appropriate IPT messages, e.g. H.225.0/H.245 messages for an H.323 system. A third-party MakeCall, for example, would be converted into a command to one of the parties of the call to originate an H.323 call.

Because the CSTA Deputy actually implements IPT interfaces (e.g. H.225.0 and H.245 protocol stacks for H.323), the components labelled C and D in the figure are not in fact terminals; they might be called Stations, and the pair of Station and CSTA Deputy together form an H.323 Terminal with an additional CSTA interface[1].

2.1. Overview

The proposed architecture for a merged CSTA/IPT environment can be used in a number of ways. For the purpose of this working paper, a relatively simple approach has been taken, and does not imply a

[1] This should not be taken to imply any specific implementation restriction but is simply used for diagrammatic clarity

specific approach being adopted by ECMA TC32-TG11. The approach is based upon the concept of keeping the CSTA Deputy as simple as possible as one will be required for each terminal.

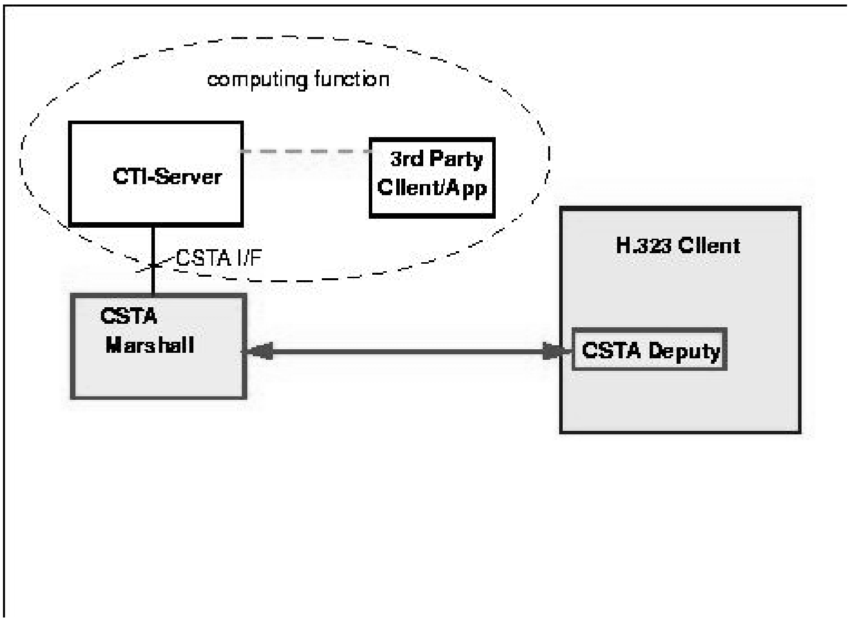


Figure 3 - CSTA Marshall/Deputy H.323 System

The CSTA Marshall maintains the current status of all the terminals that are CSTA enabled. This requires each CSTA Deputy to notify the CSTA Marshall when a change of state occurs in their respective terminal, whether due to IPT action or to manual action by the terminal user.

The CSTA Marshall is responsible for responding to CSTA messages from the CSTA applications (via the server) and only involves the CSTA Deputy when required. As the CSTA Marshall is kept informed of terminal state changes it is able to generate CSTA events and status reports without further recourse to the CSTA Deputy.

Each CSTA Deputy must register with the CSTA Marshall which will create a map of transport address to CSTA device identifiers. By the process of registration the CSTA Marshall has a complete picture of all the CSTA enabled terminals which are active and is able to report this through the CSTA Capabilities Exchange services. It will also be able to use the CSTA Capabilities Exchanges services to notify CSTA applications when terminals are first detected or are removed.

CSTA service requests which require terminal action are passed to the appropriate CSTA Deputy via a CSTA 'like' interface (I1 & I2). This same interface is used to notify the CSTA Marshall of the result of actions and state changes.

Hence, the CSTA Marshall acts as the CSTA Switching Function.

3. Flow Diagrams

Because of the complexity of the protocols involved, “proving out” the concept of a merged CSTA/IPT environment requires testing it against many scenarios. This section reports our progress in this activity.

Some scenarios are derived from the H.323 standard, for cases in which some important H.323 behaviour has no counterpart in CSTA.

In the following subsections, a number of general assumptions are made:

- Clients D1, D2, D3 support the Deputy concept.

- All Deputies register with the Marshall. After registration, all Deputies send their activities to the Marshall.
  Note: As an alternative, there could be a pre-configuration of Deputies in a Marshall, so that all Deputies are “implicitly” registered. This alternative is not shown in the following flows.

- The CSTA multi-step acknowledgement model is in use, unless otherwise indicated.

- The CSTA fixed-view model is in effect, unless otherwise indicated.

- Unless otherwise indicated, each flow uses the H.323 direct endpoint call signalling mode, not the gatekeeper-routed call signalling mode.

- Messages between the Marshall and the Deputy follow the following conventions:

  - Each message is prefixed by “DM”;

  - Service requests have the suffix “Req”;

  - Service responses have the suffix “Ack”.

- Messages and events between the Marshall and the Proxy follow the convention of using the prefix “MP”.

- Abbreviations used in the flow diagrams include:

  - C: CSTA Call ID

  - HC: H.323 Call ID (which is globally unique)

  - HCID: H.323 Conference ID

- The Marshall supports mapping between H.323 Call IDs and CSTA Call IDs.

- The CSTA Application Working Domain is dynamically populated via registration by Deputies.

3.1. Scenario: Deputy Registration with a Marshall

In this scenario, client Terminals, with their associated Deputies, register with a Marshall. This scenario has no direct counterpart in the CSTA scenarios, because in CSTA, stations are assumed to be configured into a switching sub-domain via a management function out of the scope of the protocol.

In this scenario, the Terminal registers both with the Gatekeeper function and the Marshall. The H.323 recommendation does not mandate Gatekeepers in H.323 systems, so the Gatekeeper Registration actions may not take place. Deputy registration with a Marshall, however, is expected in our proposal, as it allows the Marshall to provide some centralised services for Terminals.

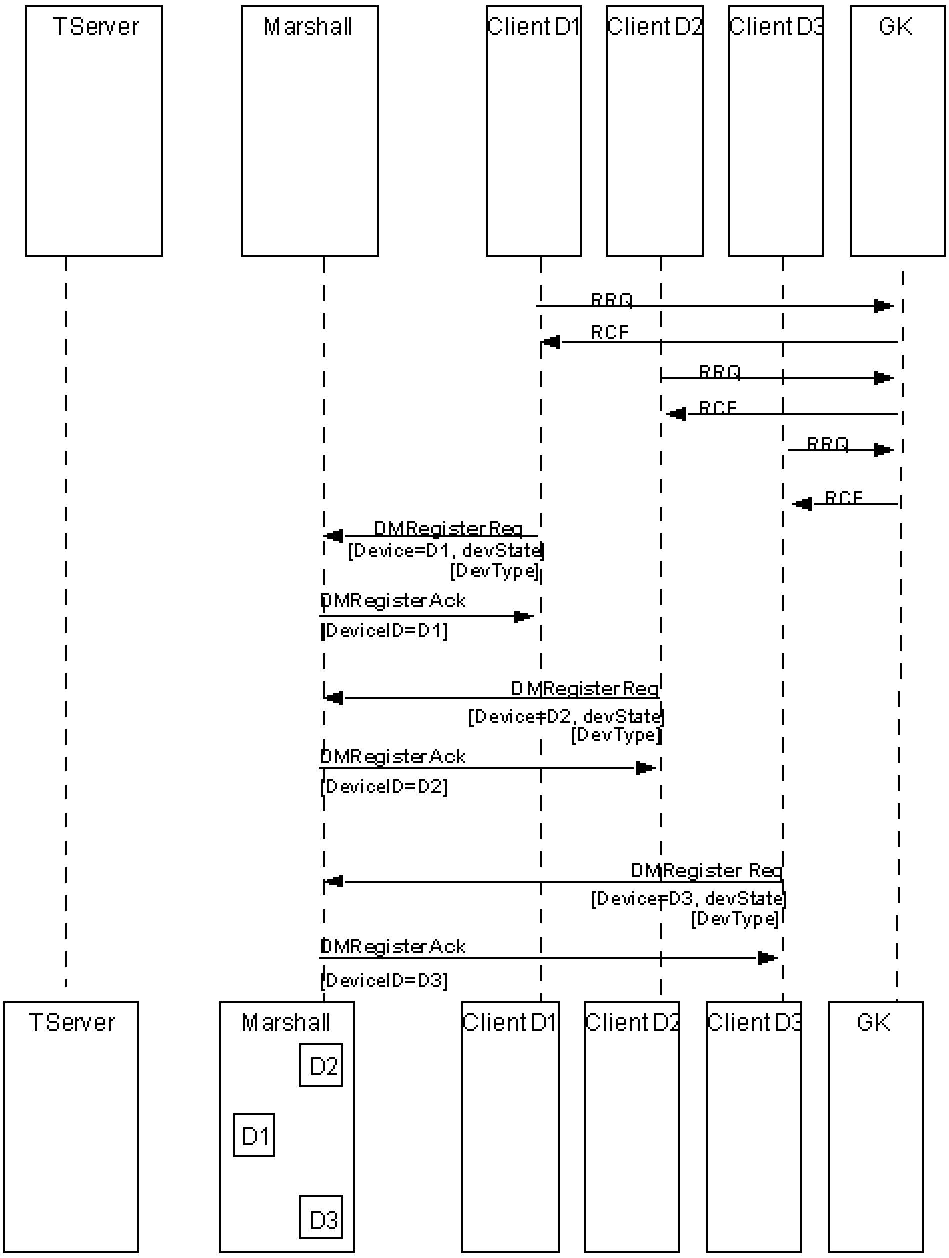


Figure 4 - Flow Diagram: Deputy/Marshall Registration

3.2. Scenario: Get Switching Function and Monitor Start

In this scenario, the server collects the list of devices configured into the application working domain. The Marshall reports back on the devices that have previously been registered with the Marshall, depicted in Figure 4.

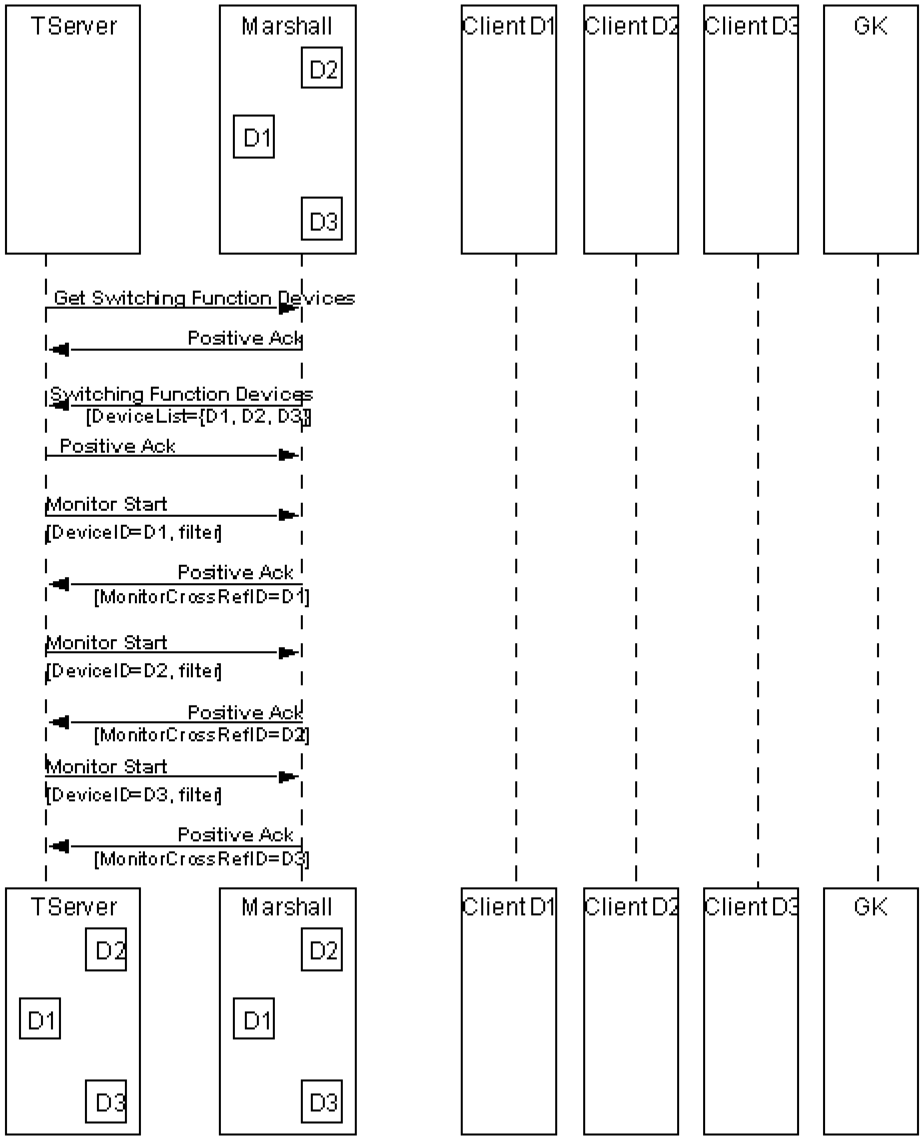


Figure 5 - Flow Diagram: Get Switching Functions and Monitor Start

3.3. Scenario: Snapshot Device

In this scenario, the server invokes the CSTA Snapshot Device service, obtaining a current status of Device D1. The Marshall is presumed to have cached this information, so all interaction takes place between the Server and the Marshall.

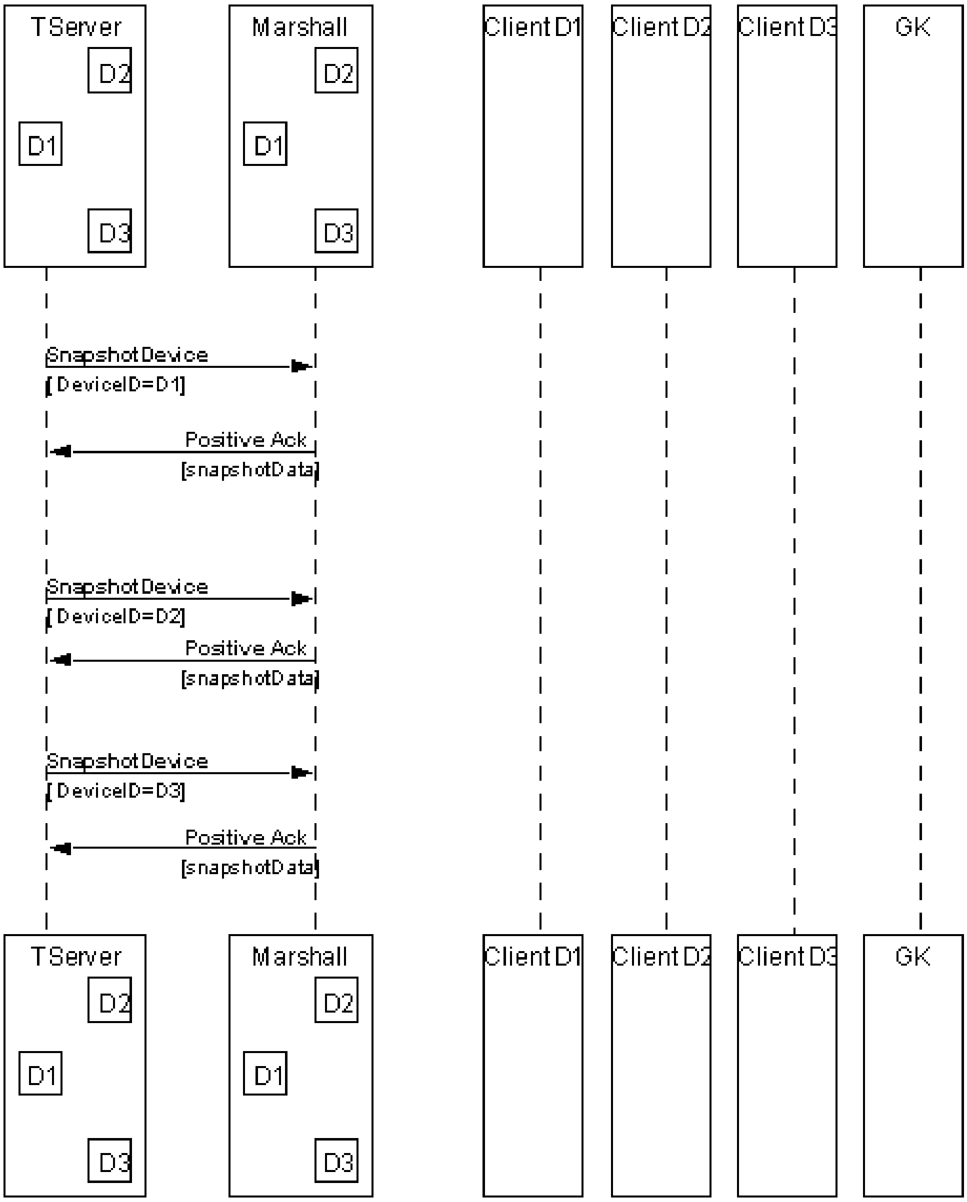


Figure 6 - Flow Diagram: Snapshot Device

3.4. Scenario: Make Call

In this scenario, a call is made from D1 to D2. The Gatekeeper mediates call setup, but thereafter signalling and control flows directly between the two terminals.

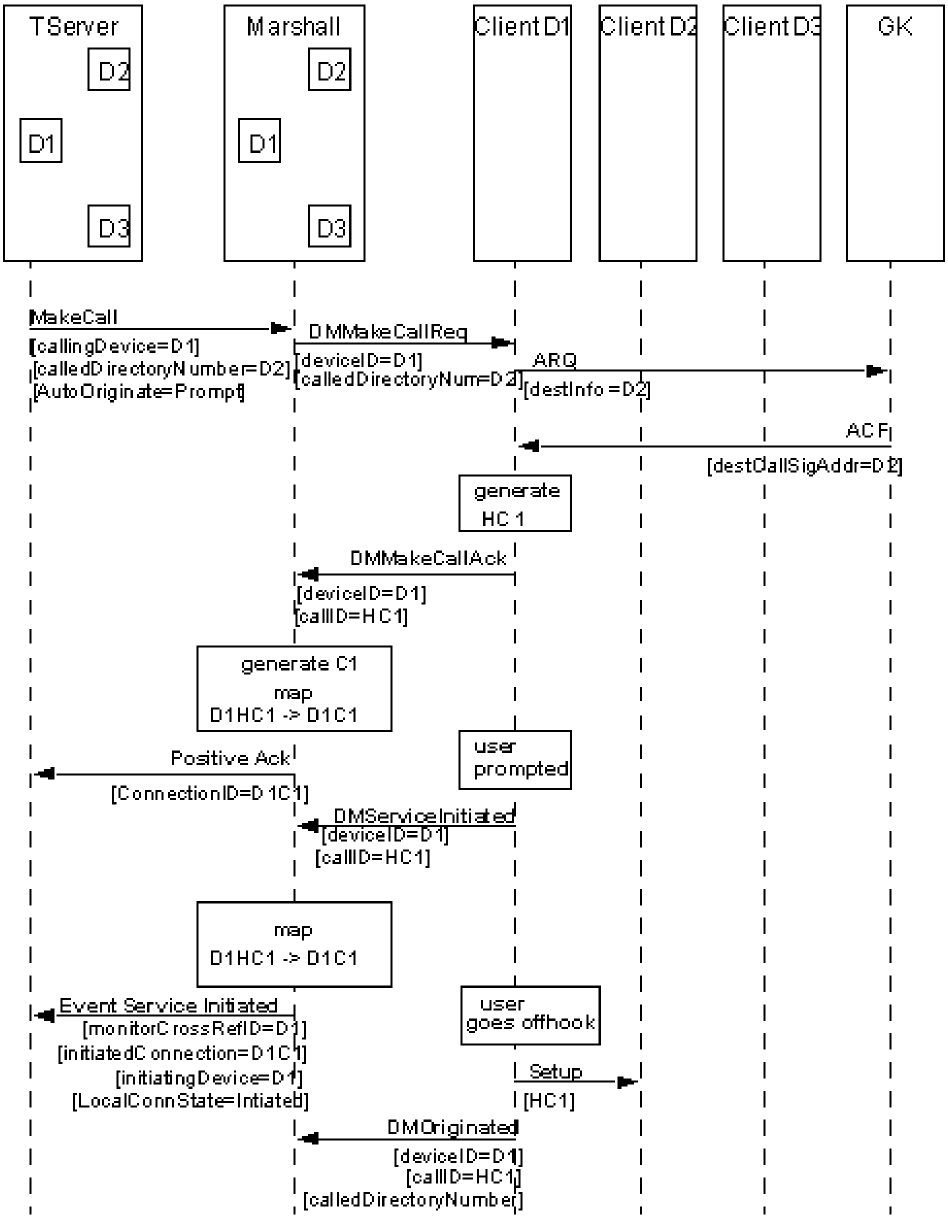


Figure 7 - Flow Diagram: Make Call (part 1 of 3)

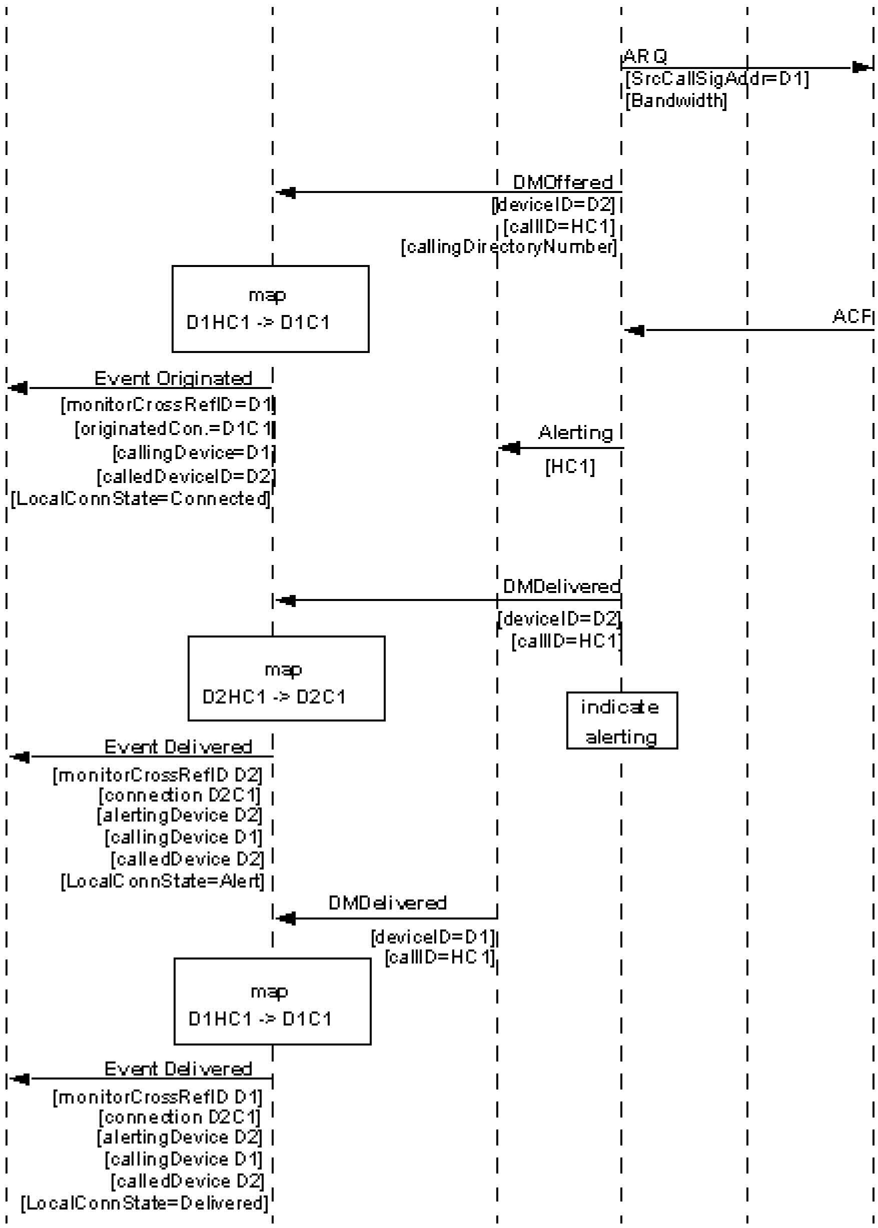


Figure 8 - Flow Diagram: Make Call (part 2 of 3)

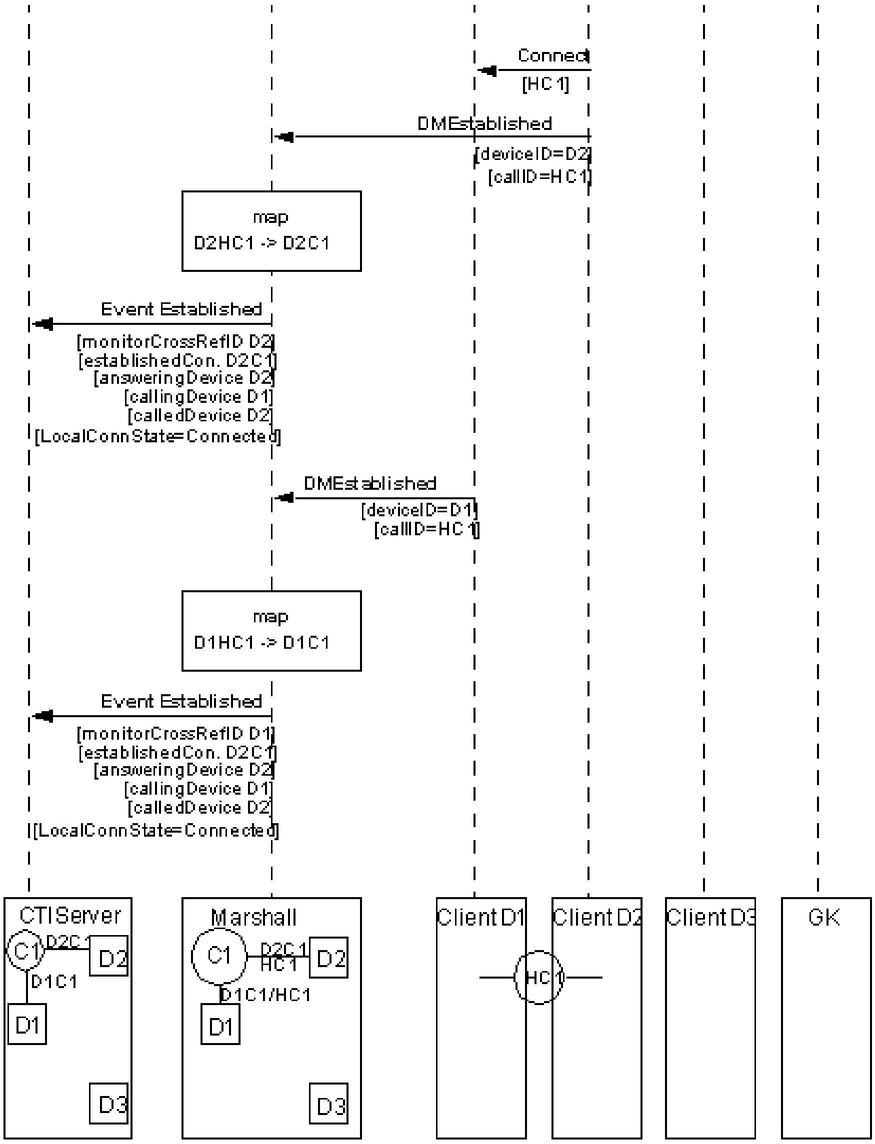


Figure 9 – Flow Diagram: Make Call (part 3 of 3)

3.5. Scenario: Consultation Call

In this scenario, which is part of a conference/transfer call scenario, a consultation call is made from D1 to D3. Because a call goes on hold as part of this scenario, the flow includes elements of the Call Hold supplementary service of H.450.4.

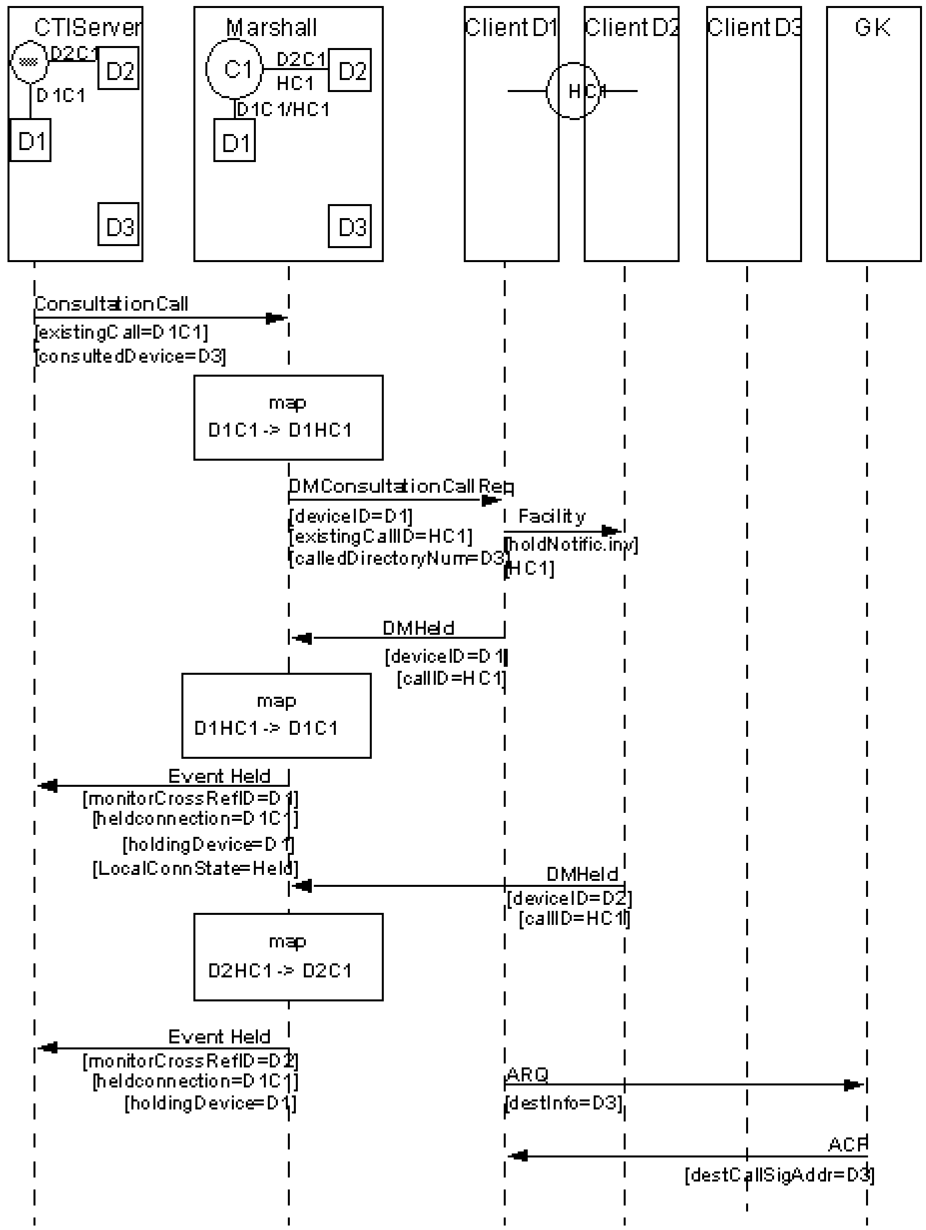


Figure 10 – Flow Diagram: Consultation Call (part 1 of 4)

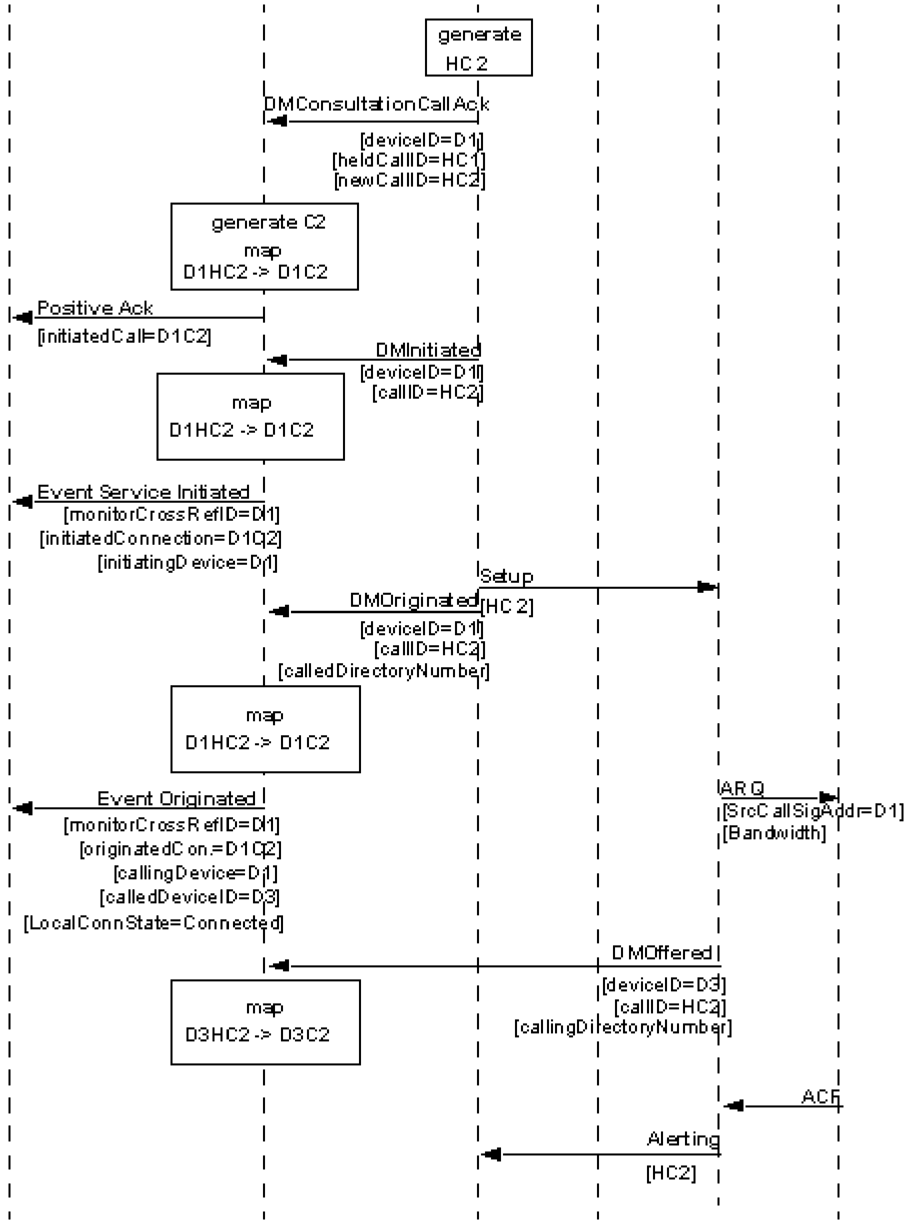


Figure 11 – Flow Diagram: Consultation Call (part 2 of 4)

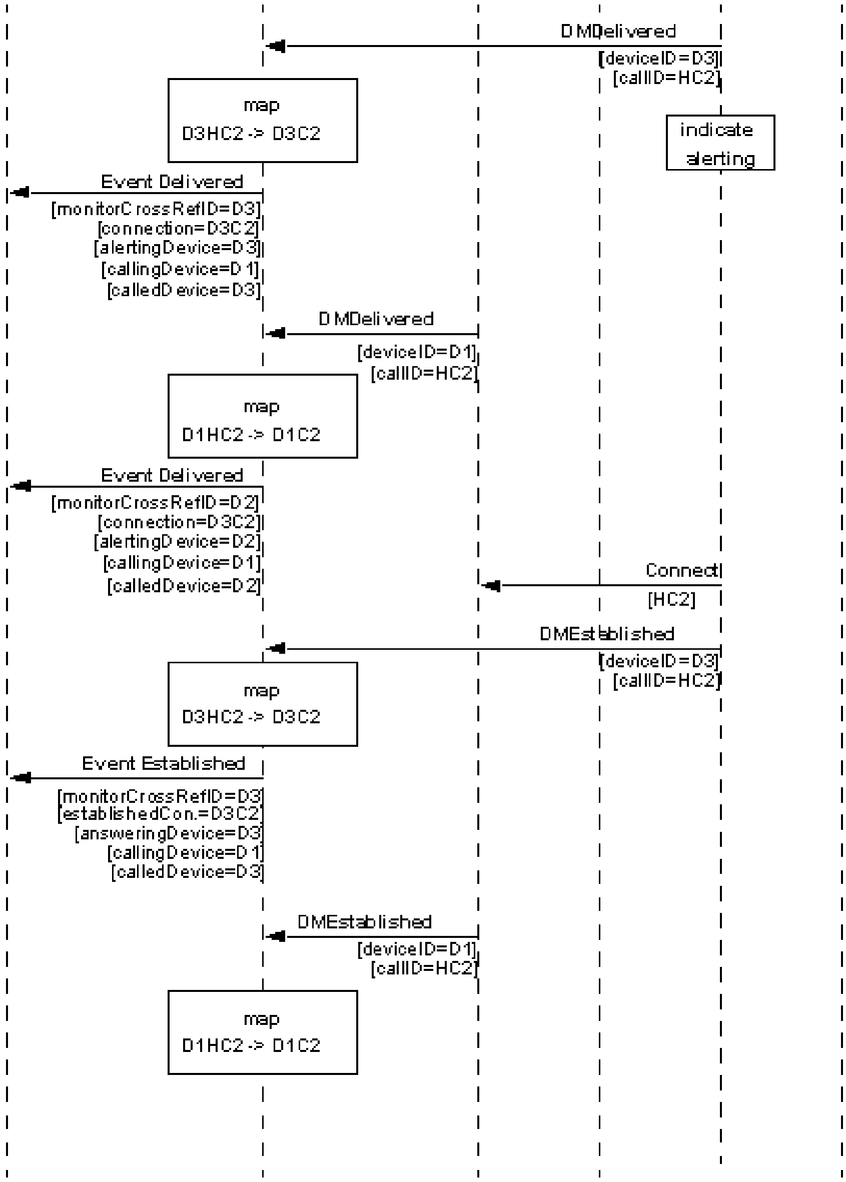


Figure 12 – Flow Diagram: Consultation Call (part 3 of 4)

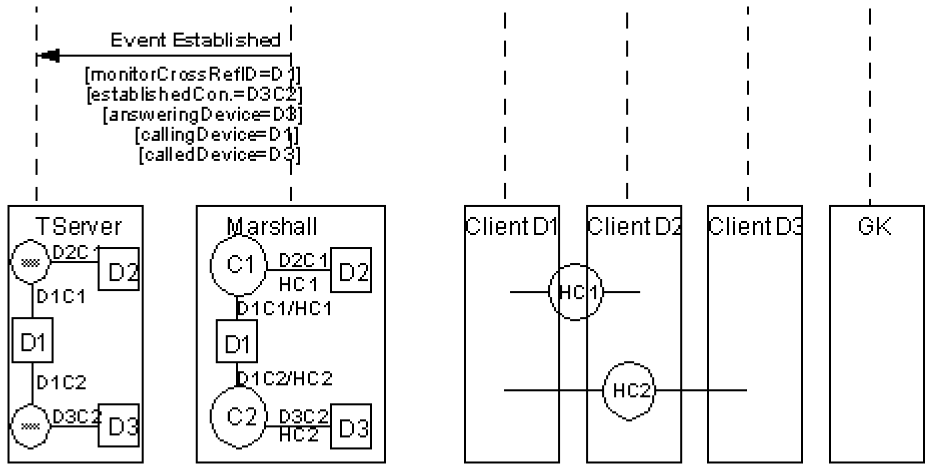


Figure 13 – Flow Diagram: Consultation Call (part 4 of 4)

3.6. Scenario: Transfer

This scenario is a continuation of 3.5. Device D1 transfers D2 to be in a call with D3. The flow includes elements of the Call Transfer supplementary service of H.450.2.

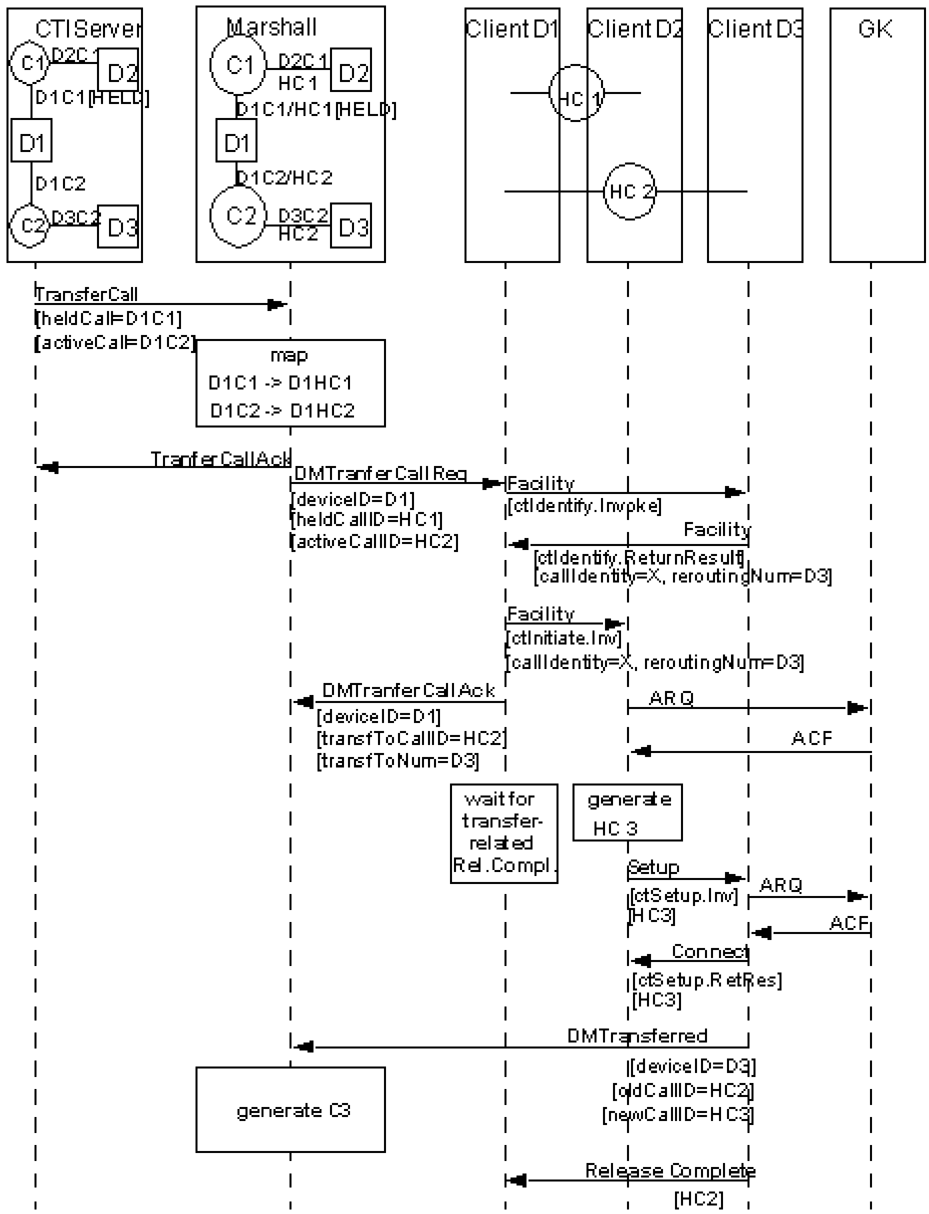


Figure 14 – Flow Diagram: Transfer Call (part 1 of 3)

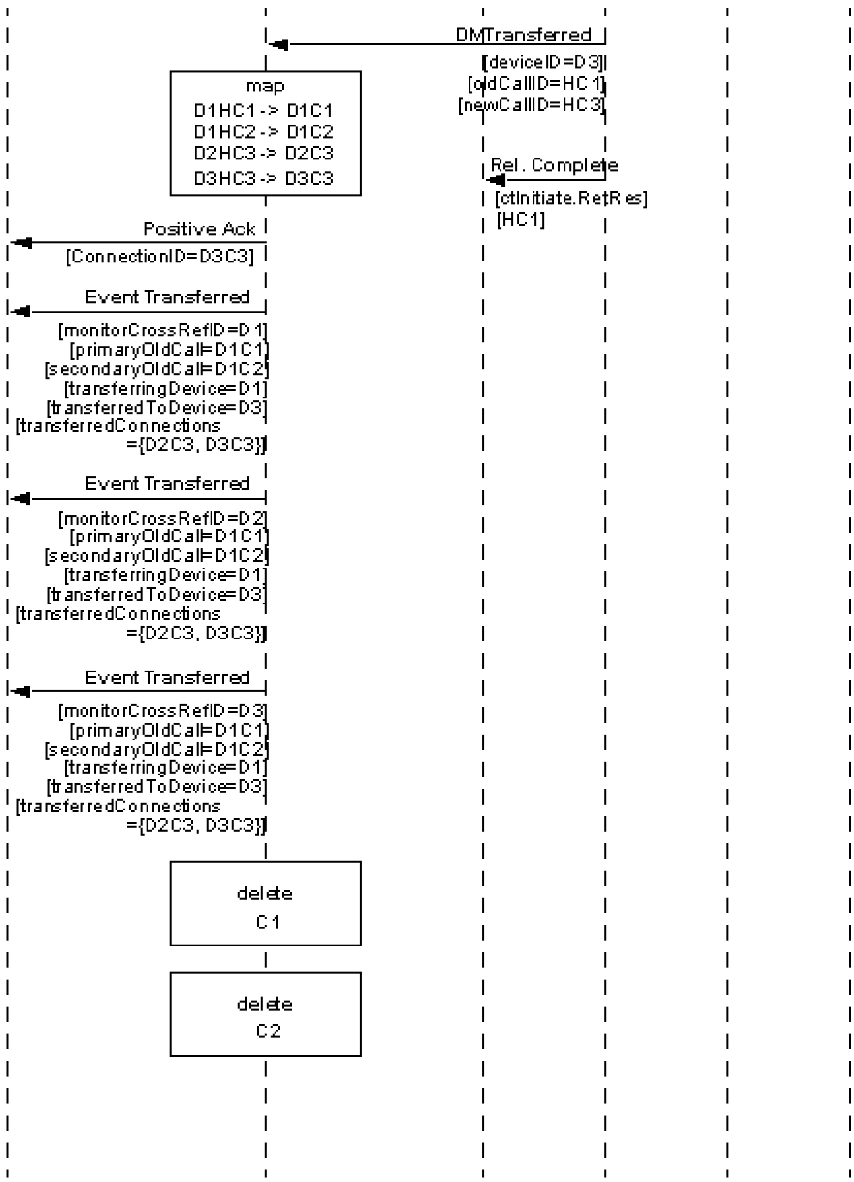


Figure 15 - Flow Diagram: Transfer Call (part 2 of 3)

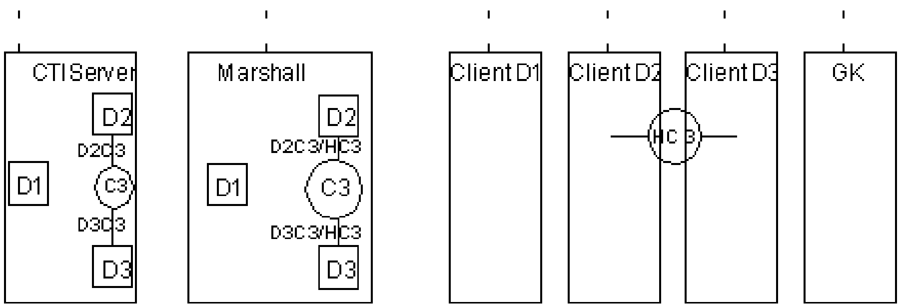


Figure 16 - Flow Diagram: Transfer Call (part 3 of 3)

3.7. Scenario: Set Non-Immediate Forwarding (Busy, NoAnswer, or DND)

In this scenario, a terminal arranges to forward calls offered to it. It makes use of the Call Diversion supplementary service of H.450.3.

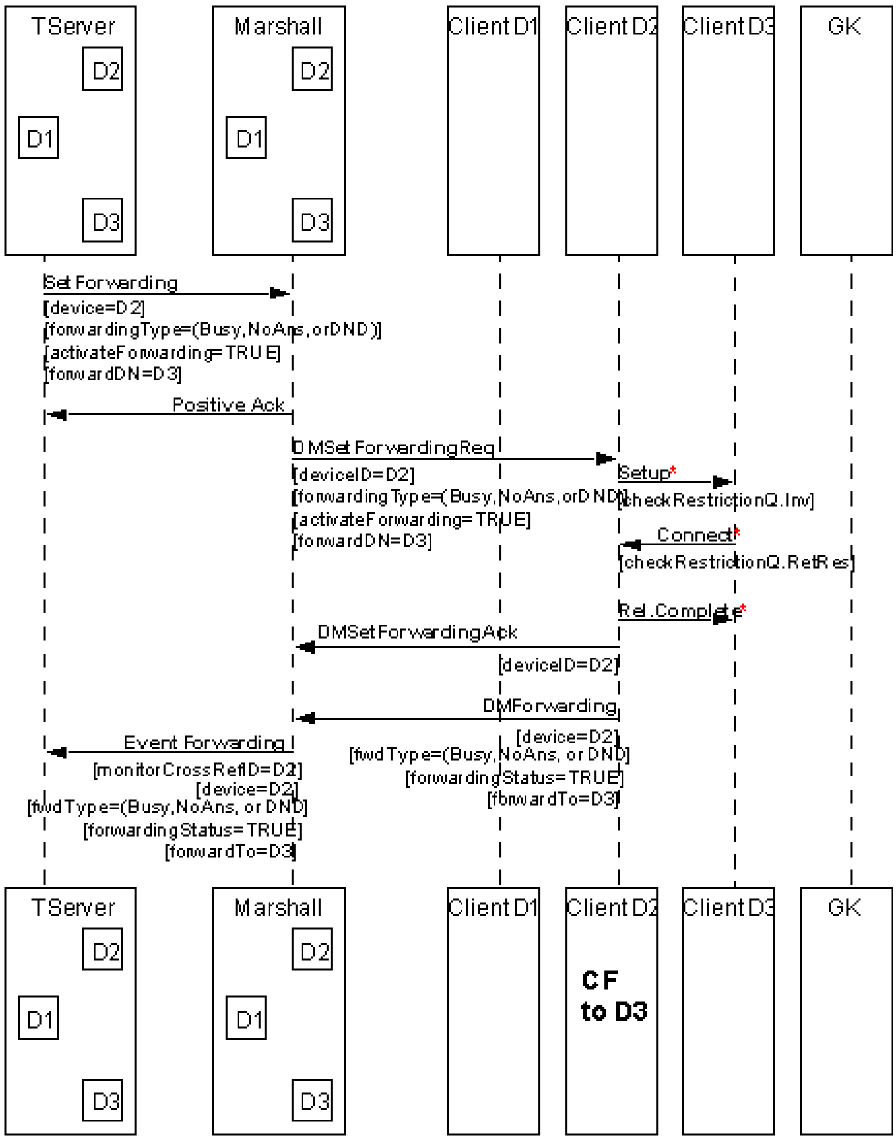


Figure 17 - Flow Diagram: Set Non-Immediate Forwarding

* The message flow of the call forward verification flow follows H.450.3.

3.8. Scenario: Deactivation of Forwarding when Busy

In this scenario, Terminal D2, which has previously set up forwarding, deactivates the service.

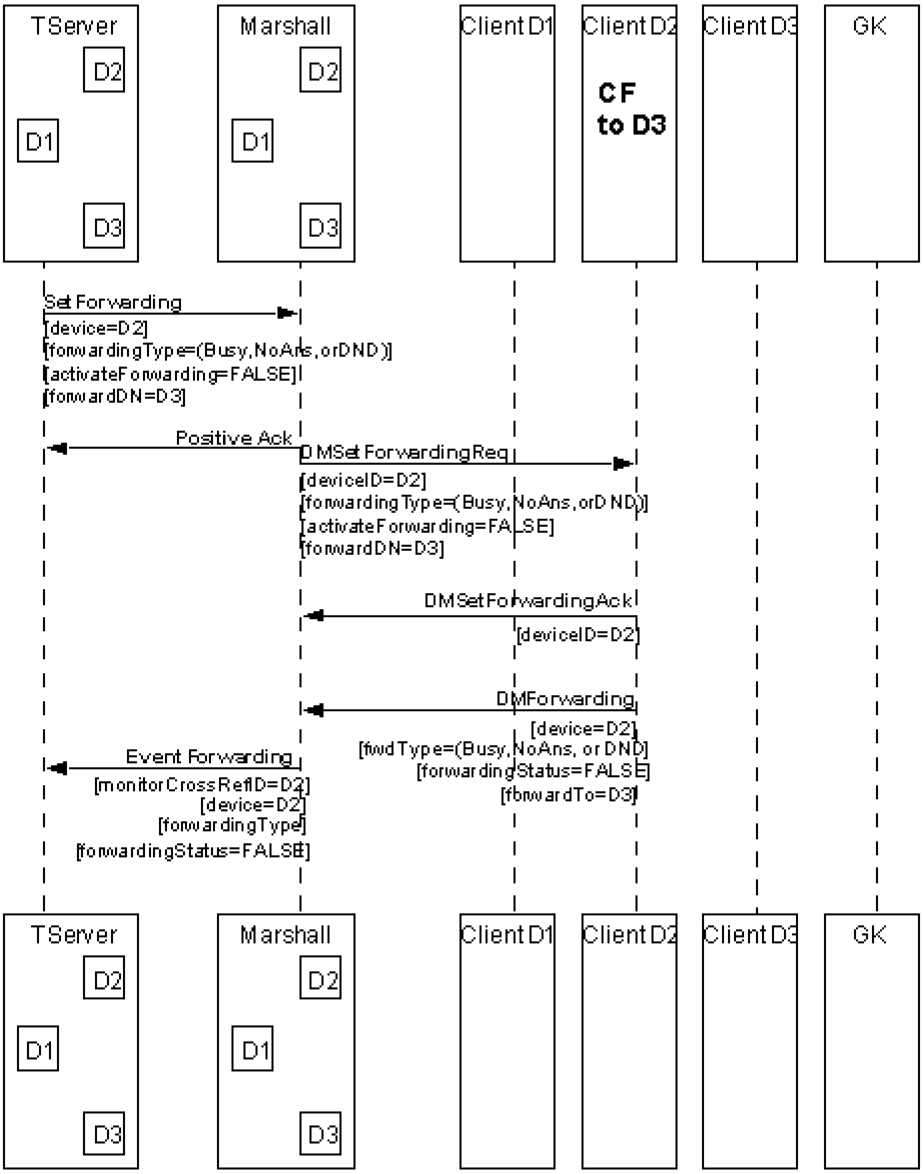


Figure 18 - Flow Diagram: Deactivation of Forwarding while Busy

3.9. Scenario: Clear Connection

This scenario depicts a single connection of an established call being cleared.

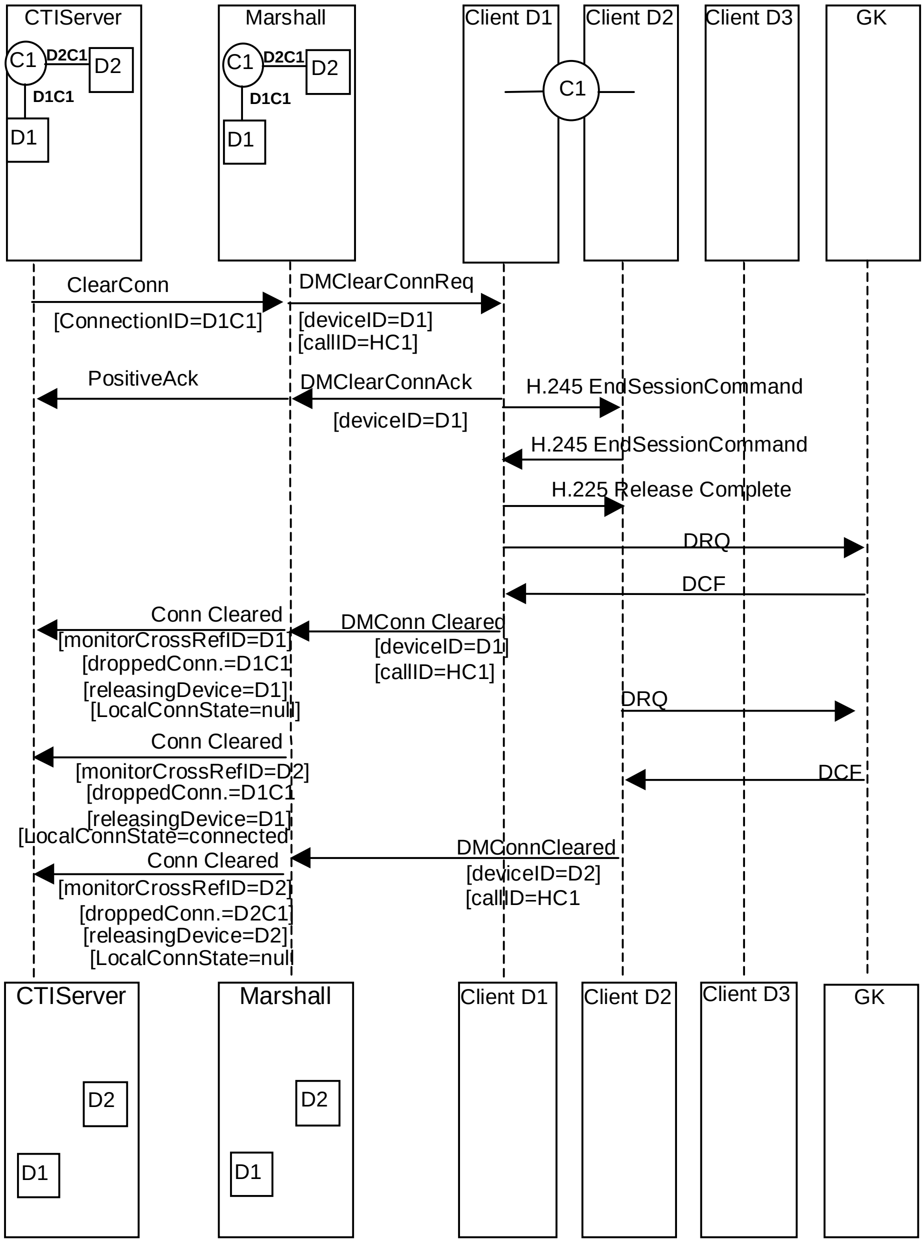


Figure 19 - Flow Diagram: Clear Connection (2 party call)

3.10. Scenario: Clear Call (2 party Call)

This scenario depicts clearing an established call between two Terminals.

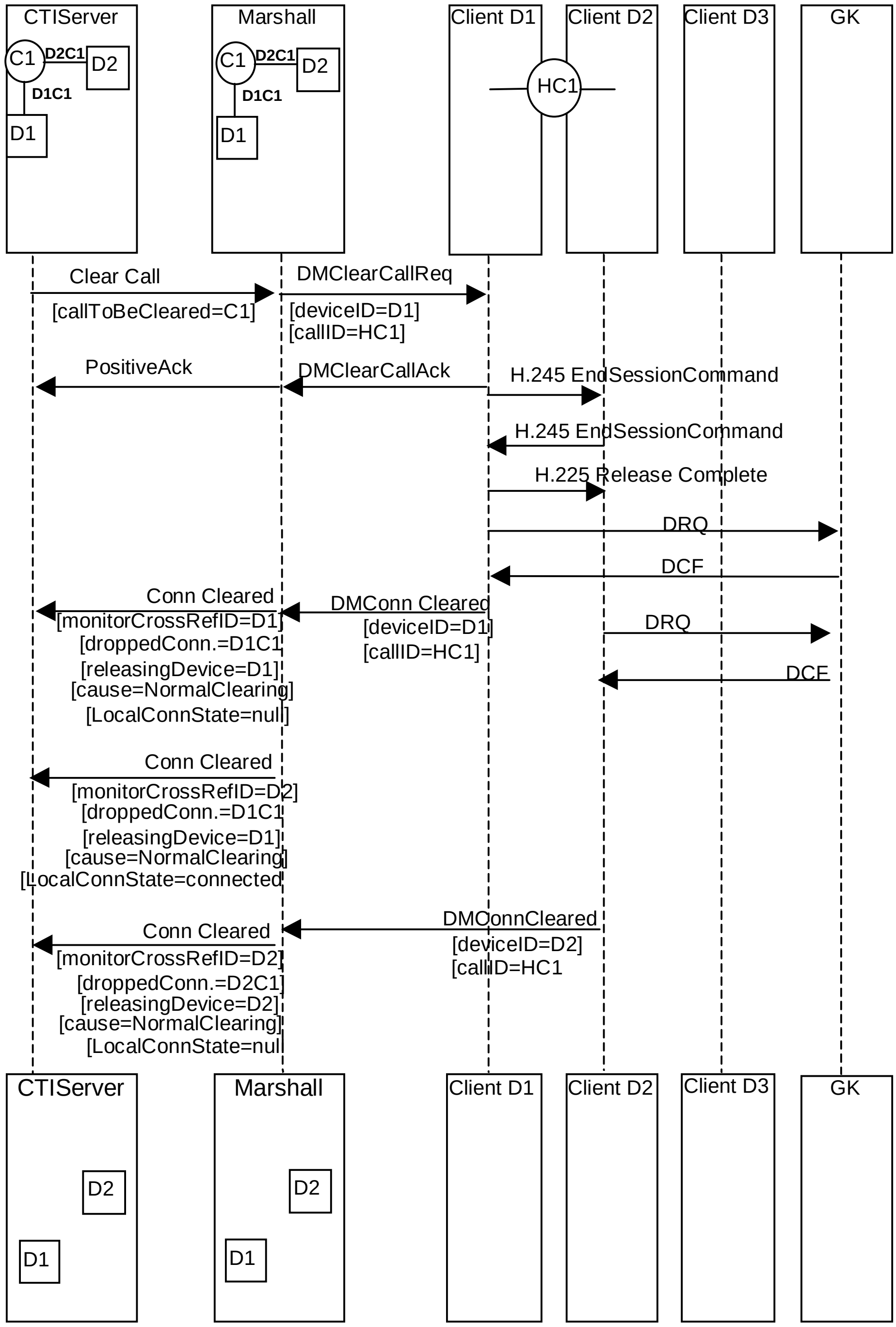


Figure 20 - Flow Diagram: Clear Call (2 party call)

3.11. Scenario: Single Step/Ad Hoc 3-Party Conference (from 2-Party Call)

This scenario assumes that a call between Terminals D1 and D2 has been previously established. A single-step conference operation is performed to add Terminal D3. The scenario shows a possible problem in mapping between CSTA Connection IDs and H.323 Call IDs.

This scenario is still unstable; its flow diagram is not included in the present revision of the Working Paper.

3.12. Scenario: Party Conference (from Consultation)

This scenario is a continuation of 3.5, in which a consultation call is made. This scenario depicts the creation of a three-party call including the consulted party. In this scenario, one of the existing devices (D1) is the MCU.

This scenario is still unstable; its flow diagram is not included in the present revision of the Working Paper.

3.13. Scenario: Conference from Consultation with Separate MCU

This scenario is a variation of 3.5, in which the MCU is a separate device from the Terminals.

This scenario is still unstable; its flow diagram is not included in the present revision of the Working Paper.

3.14. Scenario: Clear Call (3 Party Call)

This scenario depicts clearing an established call between three Terminals in a conference call. It differs from 3.10 in that mapping issues between CSTA Connection IDs and H.323 Call IDs must be resolved.

This scenario is still unstable; its flow diagram is not included in the present revision of the Working Paper.

3.15. Scenario: Clear Connection, Disconnect a Single Party from a 3 Party Call

This scenario depicts clearing a single connection from a 3-party call.

This scenario is still unstable; its flow diagram is not included in the present revision of the Working Paper.

4. Conclusions and Open Issues

This present Working Paper represents work in progress. Some of the issues currently under investigation include:

1. The work done on scenarios for Conference Call and Clear Call show that mapping between a CSTA Connection ID to an H.323 Call ID does not work with all scenarios. Here, a mapping between the CSTA Call ID and the H.323 Conference ID (CID) may be necessary, and requires investigation.
2. It is still an open issue whether these special scenarios require an additional mapping of the CSTA Connection ID to the H.323 Call ID as well as the mapping of the CSTA Call ID to the H.323 CID, or if it is sufficient merely to map between the CSTA Call ID to the H.323 CID for all scenarios.
3. The proxy concept for the Call Forwarding scenarios requires another proprietary interface between the Deputies and the proxy. The gatekeeper routed proxy concept of H.450.3 is not well documented and requires further investigation.

In addition to the issues listed above that are specifically designated for further investigation, the authors of this Working Paper solicit external contributions and comments on the following areas:

1. conferencing;
2. failure modes;
3. planned shutdown and re-registration;
4. immediate forwarding;
5. interrelationships involving multiple registrations, multiple gatekeepers, and multiple marshalls;
6. mobility;
7. user support of multiple calls;
8. user registration and multiple zones;
9. media control within a call.

References

[H.225] *Call Signaling Protocols and Media Stream Packetization for Packet Based Multimedia Communications Systems*, ITU-T Recommendation H.225.0 (02/98), 1998.

[H.245] Control Protocol for Multimedia Communication, ITU-T Recommendation H.245, February 6, 1998

[H.248] Gateway Control Protocol, ITU-T Recommendation H.248, July 30, 1999.

[H.323] Packet based multimedia communications systems, version 3, ITU-T SG 16, May 1999.

[H.450.1] Generic functional protocol for the support of supplementary services in H.323, ITU-T Recommendation H.450.1, February 1998.

[H.450.2] Call transfer supplementary service for H.323, ITU-T Recommendation H.450.2, February 1998.

[H.450.3] Call diversion supplementary service for H.323, ITU-T Recommendation H.450.3, February 1998.

[H.450.4] Call Hold Supplementary Service for H.323, ITU-T Recommendation H.450.4, September 1998.

[H.450.5] Call Park and Call Pickup Supplementary Services for H.323, ITU-T Recommendation H.450.5, September 1998.

[H.450.6] Call Waiting Supplementary Service for H.323, ITU-T Recommendation H.450.6, September 1998.

[H.450.7] Message Waiting Indication Supplementary Service for H.323, ITU-T Recommendation H.450.7, September 1998.

[ECMA-269] *Services for Computer Supported Telecommunications Applications (CSTA) Phase III*, 3rd Edition, December 1998, ECMA-269.

[Lindbergh-97] Lindbergh, David, *H.323: Multimedia Conferencing for Packet Switched Networks*, Proceedings ITCA-97, Washington, DC June, 1997. Powerpoint presentation "9706_itca_h323.ppt".

[CABLELAB] Packet Cable Interim Specifications, http://www.packetcable.com/packetcable_specs.html, Cable Labs, 1999.

[Q.931] "Digital Subscriber Signalling System No. 1 (DSS 1) – ISDN User-Network Interface Layer 3 Specification for Basic Call Control", ITU-T Recommendation Q.931, 1993.

[RTP] "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889 (1996).

[SIP] "Session Initialization Protocol (SIP)", IETF RFC 2543, 1999.

[TG16/99/8] Internet Screenphone Reference Forum Specification – Release 1, ECMA contribution ECMA/TC32-TG16/99/8.

Annex A: A Short Introduction to CSTA

This section provides a short introduction to CSTA as an aid to understanding the analysis concerning interoperability with H.323 that follows. This description is very high-level, and cannot replace reading the standard itself, which runs to many hundreds of pages. The description will touch on the CSTA system architecture, which will allow the reader to understand the conceptual components of a CSTA system in preparation for understanding their relationship with H.323 terminals, gatekeepers, and other components. After that, elements of the CSTA processing model, including devices, calls, and connections, and services are summarized. Finally, a short example of a CSTA interaction is displayed to integrate the various concepts.

This introduction is not intended to be a tutorial or reference to the standard and omits aspects of the standard which are not necessary to introduce for the purposes of this working paper.

A.1. Target O/S CSTA System Architecture

Figure 21 is a simplified depiction of a possible configuration of the CSTA architecture, showing the most important components. In this configuration, the component labelled "Call Control Activity" is commonly a private switch (e.g., a PBX), to which trunks, lines, station sets, and other telecommunications devices are attached. The vague term "Call Control Activity" is used because its implementation is very dependent on the telecommunications environment; in the case of non Gatekeeper-mediated H.323, for example, this box is an IP network. The attached devices interact with the switch and/or with each other to perform signalling (via various telecommunications protocols such as ISDN, T1/E1, H.323) and to transmit and receive media stream data.

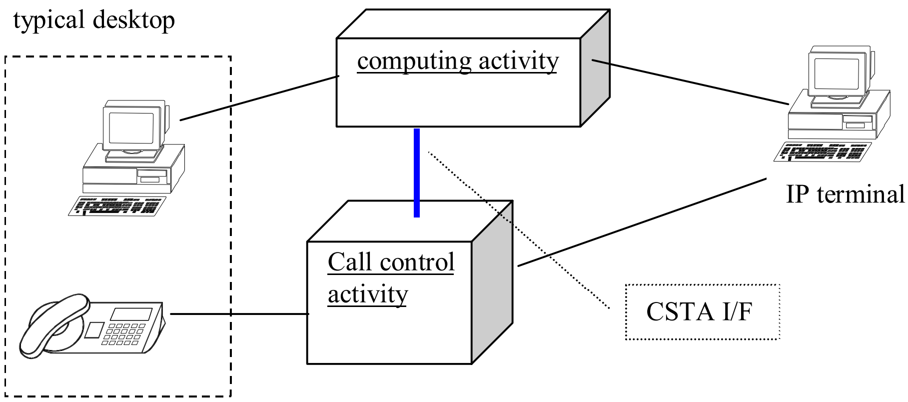


Figure 21 - CSTA Architecture

The component labelled "Computing Activity" is a computer system that provides access to call control services on behalf of applications. Access is provided via a *switchlink* (a data communications channel between the two components on which the CSTA Protocol operates). The function within the Computing Activity component that uses CSTA to invoke call control services may be an application, or it may be a CTI Service Provider that in turn provides services to client applications (e.g., by using the Java Telephony API, or JTAPI, framework).

A.2. CSTA Processing Model

The CSTA Standard (see [ECMA-269]) does not describe components, but rather *subdomains* and *functions*, as depicted in Figure 22. The Call Control Activity component and the attached telecommunications devices together comprise a *CSTA switching subdomain*. A switching sub-domain

has a common namespace for devices, connections, conferences, and other constituent objects, as well as the capability of creating “internal” calls between devices within the sub-domain.
Multiple switching sub-domains may interoperate, allowing calls to be created and moved between devices in different sub-domains, but namespaces, capabilities, etc., are not guaranteed to be consistent between sub-domains.

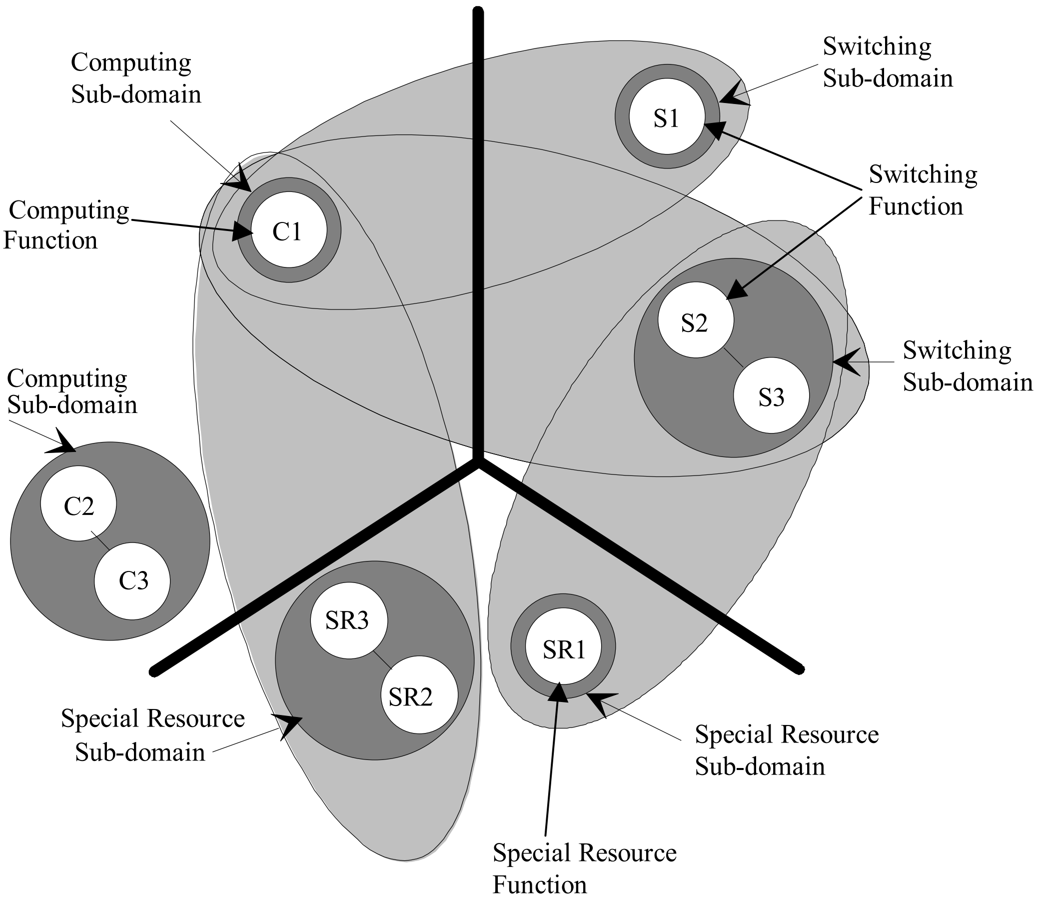


Figure 22 - CSTA Subdomain Model

The *computing sub-domain* consists of one or more software components, which may run independently of each other or which may interact with each other. The switching sub-domain consists of hardware and software components that interact with each other so as to provide voice and data calls. The *special resource sub-domain* contains hardware and software components providing a variety of services, primarily in the domain of media processing (e.g., interactive voice response).

There is an interface between the computing, switching, and special resource sub-domains, referred to as a *service boundary*. A computing sub-domain component and a switching sub-domain component interact with each other by sending *service requests* (a message requesting that a particular action be performed), *service responses* (a message issued in response to a service request, describing its disposition), and *events* (a message other than a request or response, not itself requiring a response and typically providing information about a change of state of some component) to each other across the service boundary.

The active component of the switching sub-domain (analogous to the Call Control Activity component of Figure 21) is the *switching function.* It is configured, initialized, and stopped by management services. Telephony services and events (e.g., calls between phones within the switching sub-domains, inbound/outbound calls from/to external devices) may take place in the switching sub-domain without regard of whether objects in the computing sub-domain are interacting with it. Software components in the computing sub-domain that request call control services do so in addition to the services that the switching function performs autonomously or which are performed by the manual interaction of a user with a device.

The active component of the computing sub-domain (analogous to the Computing Activity component of Figure 21) is the computing function. It interacts with a switching function via the CSTA protocol.

A.2.1. CSTA Devices

A CSTA device (or, usually, just device) is an abstraction in the switching sub-domain that represents various types of telephony endpoints and allows access to telephony services. A device can be a single endpoint (e.g., phone), a composition of more basic devices (e.g., a multiline phone) or multiple endpoints that identify a single conceptual entity (e.g., ACD group). All devices in a switching sub-domain are endpoints; they either interface to an off-net device (i.e., a device outside of the switching sub-domain), or are manipulated by a user.

A device is represented by its attributes (e.g., its identifier, type, profile, the physical and call states in which it might be set), the services which it provides, and the features which it provides or which modify its operation. Attributes, services provided, and features are referred to collectively in this working paper as *device interfaces*. Attributes can be observed and set (resulting in control of the device) by the computing sub-domain. Services and features may be invoked from the computing sub-domain, resulting in a call control operation taking place involving the device.

A device may have a *logical element*, with device interfaces used in call control operations, as well as a *physical element*, with interfaces used by an end user. Examples of a physical element are a microphone or handset.

A *logical element* is the set of device interfaces that are used to observe and control a call, and to control the media stream paths and associated signal processing facilities that are used by the device when involved in a call. It can determine the state and other attributes of a call, and it may change the state or attributes of a call. If it is part of a device with a physical element, it may arrange for the media stream associated with the call to be accessible to the physical element (e.g., causing the data stream to be available to the speaker or microphone).

A.2.2. CSTA Calls and Connections

A *call* is a communications relationship between one or more devices, as depicted in Figure 23. A call comes into existence when a service or a user initiates a call from a device, or when a call from outside the switching sub-domain arrives at a device (e.g., a trunk). A call terminates when no devices in the switching sub-domain are connected to it.

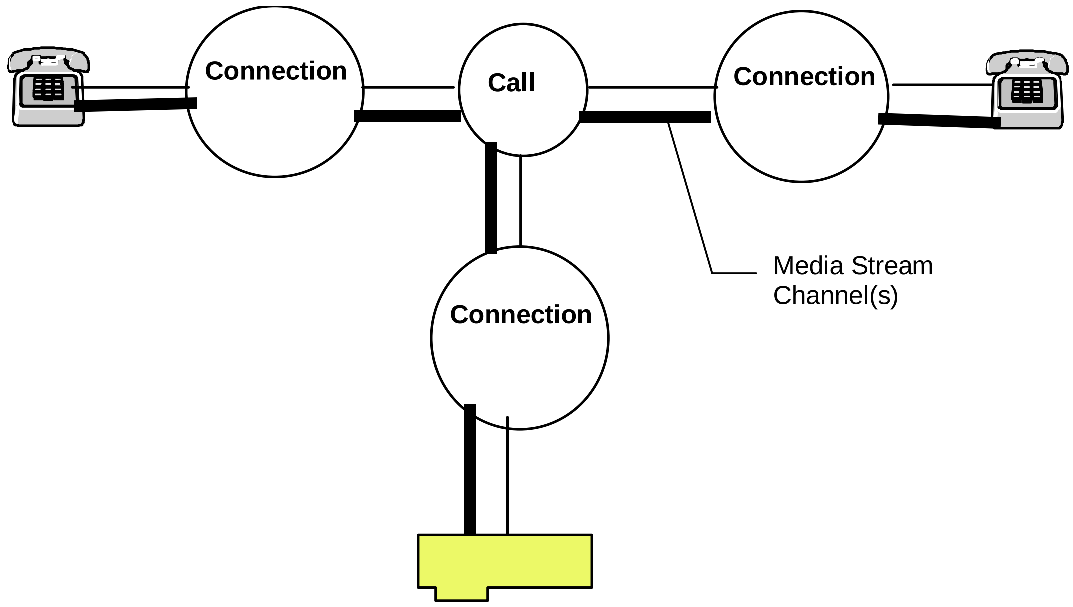


Figure 23 - Calls and Connections

The existence of a media stream between the devices participating in a call is implied by CSTA, but not explicitly defined. Further, when more than two devices are connected to a call, signal processing functionality is implied that sums the output media streams of all connected devices. Services making use of signal processing functionality in the course of call control services (e.g., conferencing, dialing) do not explicitly manipulate the hardware and/or software performing the functions; they simply fail with an appropriate error message if the resources are not available or insufficient to accomplish a particular service.

Calls also have associated with them permanent data, called *correlator data*, and data transferred between devices on demand, called *user data*. These data sources are described in more detail below.

A connection is a switching sub-domain representation of the relationship between a device and a call.

A connection's *connection identifier* consists of at least a device identifier or a call identifier, but typically a device identifier/call identifier pair. A connection identifier consisting of only a call or device identifier may be used only in explicitly defined circumstances where such an identifier is not ambiguous.

Connection identifiers must be unique for a service boundary (e.g., a session between a computing sub-domain and a switching sub-domain).

Connection identifiers may change during their lifetime, because call identifiers may change during the lifetime of a call. When this happens, events are generated by the switching sub-domain which document the change of a connection identifier.

A connection has associated with it a media stream, identified by a *Media Stream Identifier*. A device capable of performing the media services Attach Media Serviceand Detach Media Service makes the data of the media stream on its connection available to a media server.

A connection has an associated *connection state*, an attribute that characterizes the relationship of a call and a particular device to each other. A connection moves from one connection state to another as a call progresses, as depicted in Figure 24.

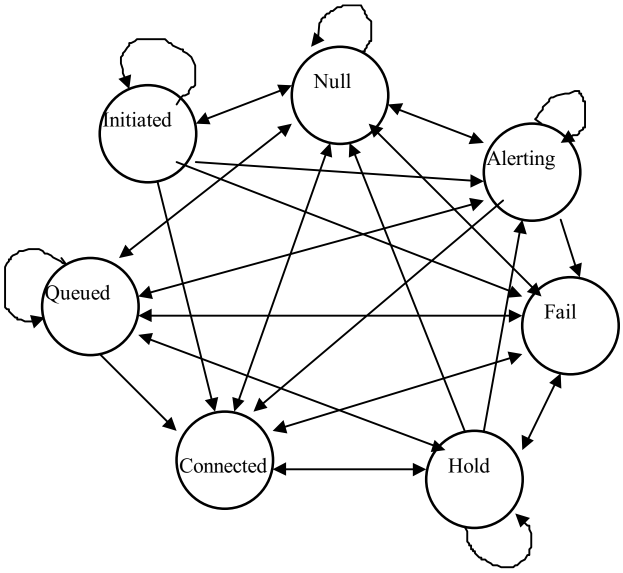


Figure 24 - Connection State Model

A switching sub-domain may have a connection state model with more states than are shown here, if the resulting state transition model contains the model of the above figure as a subgraph and if the information provided on connection state events is not affected.

A connection may be manipulated by the computing sub-domain by invoking services such as Make Call, Transfer Call, and observed by the computing sub-domain by monitoring events associated with it.

A.2.3. CSTA Media

A media stream is the channel through which data is transmitted between devices connected on a call. It is associated with a media stream identifier that is static for the lifetime of the connection, and a *media stream flow direction* (transmit, receive, or bidirectional). It is composed of one or more media stream channels, which are the pathways over which the data is carried. The characteristics of the media stream channels (e.g., bit rate, delay tolerance) may be different for each channel in the media stream, with the exception of the media stream flow direction (transmit, receive, or bidirectional) which is associated with the media stream.

The computing function cannot directly observe or control the media stream, in the sense of providing media processing algorithms that act on the data. For a call not requiring media processing, it can specify or *mediaCallCharacteristics* parameters that encode QOS information. For calls requiring interactive voice response (IVR) functionality, a switching function may support Voice Services, which conceptually controls the operation of a voice response subsystem and allows voice files to be recorded and played. A switching function may also support the *Media Attachment* services, which allow a device in the switching sub-domain to make its media stream available to an external system that may perform media processing on it.

A.2.4. CSTA Services

A interaction between the computing sub-domain and the switching sub-domain to perform some call control action. A service may be originated by either domain; certain services define actions performed by the switching sub-domain on behalf of the computing sub-domain, and certain services are performed by the computing sub-domain on behalf of the switching sub-domain.

A service consists of a service request message sent from the requestor (domain) to the acceptor, and a corresponding response from the acceptor to the requestor. Each message contains service-specific parameters.

Either a switching sub-domain or a computing sub-domain may be a service requestor. In CSTA, most services are requested by the computing sub-domain.

For each service, a given acceptor may interact with the requestor via either a single-step or multi-step "model". In the single-step model, a positive response means that the service was completed successfully ("success" is defined by the service), and a negative (i.e., error) response means that the service failed. In the multi-step model, a positive response means that the request appeared to be a valid request, and that the acceptor is in some stage of performing the service, while a negative response means that the request was rejected as invalid for some reason, and that the acceptor did not perform any action in response to the request. In the multi-step model, the response corresponding to a single-step service response is returned to the requestor via an event; the particular events received are service-dependent.

The services, events, and flows are far too voluminous to be enumerated in this paper. The reader is referred to [ECMA-269] for details of the service and event definitions, and to [VOL4] for details of the call flow definitions.

A.2.5. Example CSTA Flow

Figure 25 depicts the message flow between a Computing Function and a Switching Function for a simple scenario in which a call is made from an originating Terminal and a destination Terminal. The flow shows a single service invocation and its corresponding acknowledgement message, followed by events depicting stages of the call setup. The additional events flow to the Computing Function because, prior to the invocation of the MakeCall Service, a *monitor* was set up on the originating Terminal, indicating to the Switching Function an interest in events pertaining to that particular device.

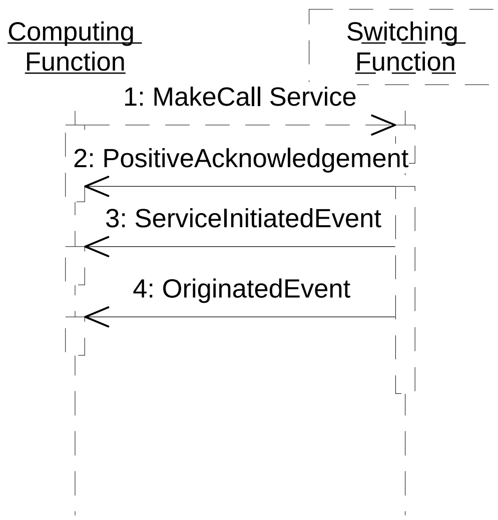


Figure 25 - CSTA MakeCall with complete Dialling Sequence

Annex B: A Short Introduction to H.323

Ultimately, the work described in this working paper will be extended to analyze interoperability between CSTA and all of the major IP Telephony standards, including SIP and Megaco; for the purposes of this paper, however, only H.323 will be described.

This introduction is not intended to be a tutorial or reference to the standard and omits aspects of the standard which are not necessary to introduce for the purposes of this working paper.

The reader familiar with H.323 will realise that the scope of this technology includes not only standards referenced in H.323, such as H.225 and H.245, but also an additional set of standards and draft standards defining supplementary services, collectively referred to as H.450. We refer to the collection of all of these relevant standards as “H.323” in the interest of a concise nomenclature.

B.1. H.323 Architecture

H.323 is an ITU-T Recommendation describing “terminals and other entities that provide multimedia communications services over packet based networks”. It defines a variety of entities, including Terminals, Gateways, Gatekeepers, Multipoint Controllers, Multipoint Processors, and Multipoint Control Units, and describes signalling protocols and procedures by which they interoperate over a packet network to support various kinds of audio, video, and data calls. Figure 26 (adapted from [H.323]) depicts the environment in which H.323 operates and shows the scope of the H.323 standard.

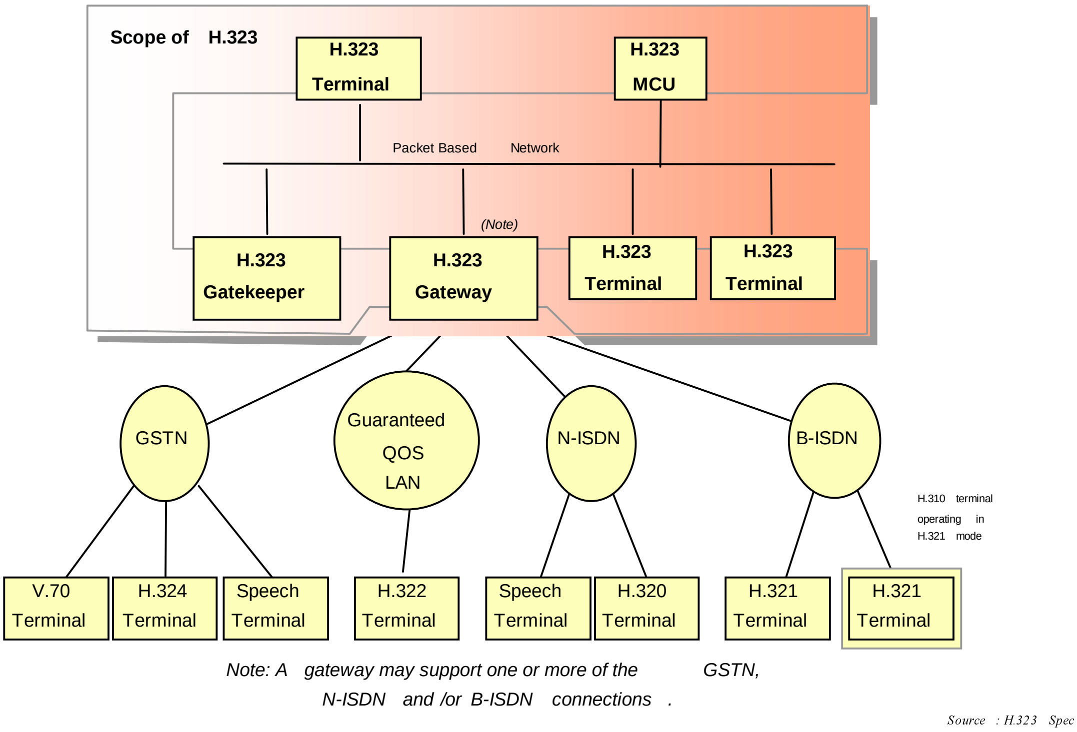


Figure 26 - H.323 High-level Architecture (adapted from [H.323])

B.1.1. H.323 Functional Units

H.323 does not specify particular devices or components, but rather describes functions, which may be implemented in separate components or in a single multi-function component.

The most important functional units are

- *terminals*, through which users invoke telecommunication services, and
- *gatekeepers*, which when present provide address translation, access control, directory, registration, bandwidth management, and other services for terminals whose users wish to communicate.

Other functional units in the H.323 architecture, which will not be covered in detail in this working paper, are:

- Gateway: An endpoint on the network which supports communications between an H.323 terminal and terminals on a switched circuit network, or to another gateway. The Gateway has the characteristics of an H.323 Terminal on the network.
- Multipoint Controller: an entity that controls multiple terminals participating in a conference.
- Multipoint Processor: an entity that performs audio/video mixing in support of a multi-user conference.
- Proxy: an entity (defined in H.450) supporting call diversion and call forwarding services for terminals.

H.323 is an architectural standard which uses the following interface standards to perform the communication required within the H.323 network. These interface standards can be functionally categorised as:

- H.225.0 (Call Signalling): is used by terminals to register with the Gatekeeper, determine endpoint addresses from *aliases* (a handy name), and to request permission to use the network.
- H.225.0 (Call Control): is analogous to Q.931 in that it is used to control call activity. It also identifies the logical channels that are used to carry the bearer (i.e. RTP/RTCP for audio etc)
- H.245 (Control Channel) is used to negotiate media to be used in the call
- H.450 (Supplementary Services): definition of call conferencing, transfer, diversion, forwarding, and other supplementary services that may be supported by H.323 functional units.

B.1.2. Calls in an H.323 Environment

In H.323, setting up and operating a call is a fairly complex process, because it must take into account not only the greater degree of control that a terminal may have over media streams, but also the different system configurations that may be in effect. Figure 27 depicts one such configuration.

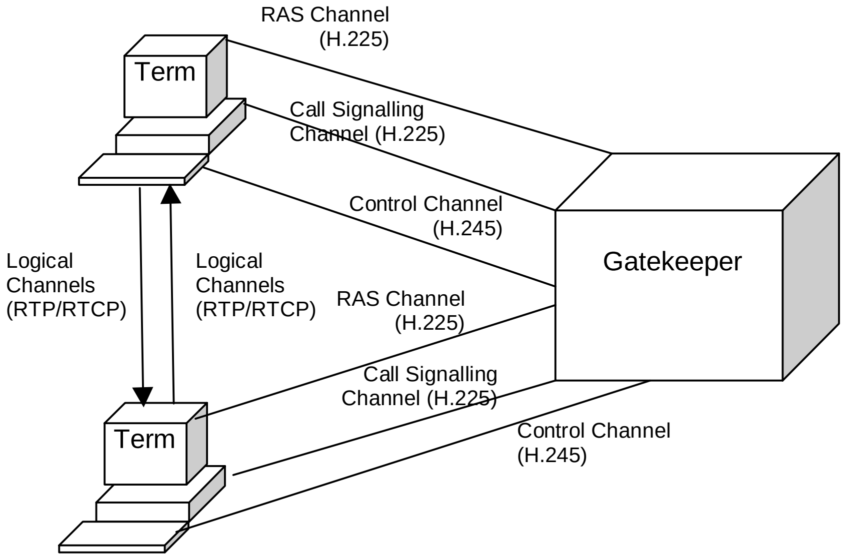


Figure 27 - Possible System Configuration for an H.323 Call

In this configuration, a call exists between two terminals. The call setup was mediated by a Gatekeeper, through which the terminals located each other, received permission to make and answer a call, negotiated capabilities, and set up the logical channels for media stream data.

Each of the control and signalling channels shown in this figure have different logical functions, and are specified by standards referred to by H.323. The H.225.0 standard ([H.225]) describes the signalling protocol between two endpoints, whether through a direct interaction or mediated by a Gatekeeper. The H.245 standard ([H.245]) describes the control protocol by which the terminals agree on a master/slave relationship for the call, set up logical channels, and operate the call. The behaviour of the logical channels is specified by the RTP/RTCP protocols ([RFC1889]).

The role of the Gatekeeper is crucial to the function of the configuration:

- The Gatekeeper is allowed to be omitted. In this case the H.225.0 address of a terminal to be called has to be known by the initiating terminal. All interfaces are direct between the terminals involved in a simple (no MCU involvement) call.

- If the Gatekeeper is present then all terminals must register with it and use it. When present the Gatekeeper can be used for:

  - Addressing and admission. In this case the Gatekeeper is used by each terminal in a call to discover the H.225.0 address of another terminal from an alias, and to request permission to contact that terminal.

  - Gatekeeper routed. In this case the Gatekeeper, in addition to providing addressing and admission, also acts as a signaling endpoint representing other terminals in a call. This means that all signaling passes through the Gatekeeper and only the RTP bearer channels connect directly between the terminals involved in a call (as shown in Figure 27).

B.1.3. Example H.323 Flow

Figure 28 depicts the message flow between two Terminals and a Gatekeeper as a call is made from Terminal1 to Terminal2. This flow assumes that the terminals wish to use the Gatekeeper to mediate the call; the all signalling and control messages therefore flow through the Gatekeeper on route from source to destination.

Notice that, in contrast to the message flow of Figure 25, the message flow takes place between entities in the Switching Domain, rather than between Switching and Computing Functions in two different domains.

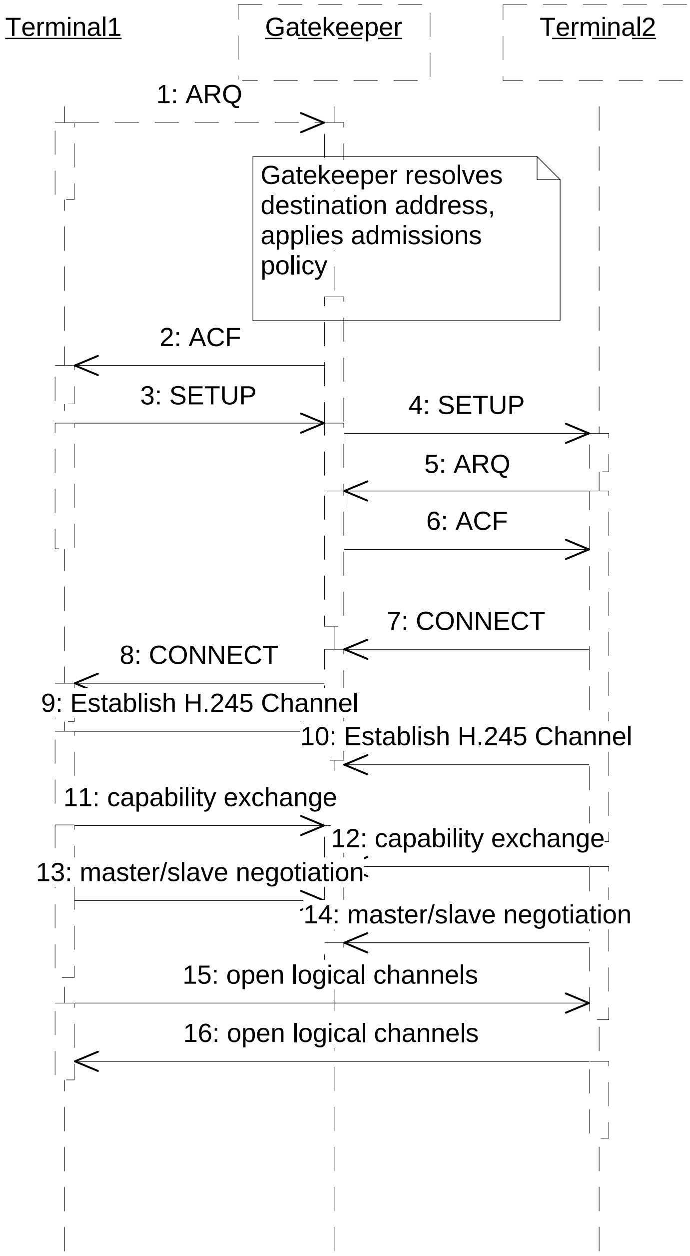


Figure 28 - Making a Gatekeeper-Mediated Call