ECMA Technical Report TR/82
December 2000

ECMA

Standardizing Information and Communication Systems

Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase III

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:**http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

ECMA Technical Report TR/82
December 2000

ECMA

Standardizing Information and Communication Systems

Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase III

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **URL:**http://www.ecma.ch - **Internet:** helpdesk@ecma.ch

Brief History

This Technical Report provides example call scenarios based upon Phase III of Services for Computer Supported Telecommunications Applications (CSTA). This Technial Report is part of a Suite of Standards and Technical Reports for Phase III of CSTA. All of the Standards and Technical Reports in the Suite are based on practical experience of ECMA member companies and each one represents a pragmatic and widely-based consensus.

The evolution of this Suite began with CSTA Phase I, which included only the CSTA Services and Protocol Standards (ECMA-179 and ECMA-180). In Phase II, Technical Report ECMA TR/68 was added illustrating how CSTA services and events may be used in typical call scenarios.

Phase III of CSTA extends the previous Phase II Standards (ECMA-217 and ECMA-218) in major theme directions as well as numerous details. This incorporates technology based upon the *versit* CTI Encyclopedia (Version 1.0), which was contributed to ECMA by *versit*. Major areas of advancement include:

- New categories of services and events such as capabilities exchange, charging, media attach services, call data recording (CDR), etc.
- Additional services and events for call and device control.
- Enhancement to existing services and events.
- Organization of services and events to reflect a grouping based on function (call control, device control, etc.).
- Use of a consistent template for services and events that includes initial/final connection state, connection state transitions, event monitoring sequences, etc.

Adopted as ECMA Technical Report TR/82 by the General Assembly of December 2000.

Table of Contents

1 Scope **1**

2 References **1**

3 Definitions and Abbreviations **1**

4 Call Origination Scenarios **3**

4.1 Make Call service - calling device is prompted to go off-hook 3

4.2 Make Call service - calling device is in hands free mode 5

4.3 Make Call service - calling device is already off-hook 6

4.4 Manually dialled call 7

4.5 Manually dialled call showing individual digits dialled 8

4.6 Dialling using Dial Digits service 10

4.7 Multi-stage dialling 11

4.8 Make Call service - called device is busy 13

4.9 Make Call service - call attempted to a busy calling device (negative acknowledgement) 15

4.10 Make Call service - called number is an invalid number (negative acknowledgement) 16

4.11 Manually dialled call - dialled number is invalid 17

5 Answering Call Scenarios **19**

5.1 Answer Call service 19

5.2 Manually answering a call 20

6 Call and Connection Termination Scenarios **21**

6.1 Device disconnects from a call by going on-hook (remaining device is cleared from the call) 21

6.2 Device disconnects from a call by going on-hook (remaining device goes blocked) 22

6.3 Device disconnects from a call using the Clear Connection service (remaining device is cleared) 23

6.4 Device disconnects from a call using the Clear Connection service (remaining device goes blocked) 24

6.5 Device disconnects from a conference call using the Clear Connection service 25

6.6 Clearing a two-party call using the Clear Call service 26

6.7 Clearing a conference call using the Clear Call service 27

6.8 Call is cleared after an alerting time-out 28

7 External Outgoing Call Scenarios **29**

7.1 Make Call service - called device is outside the CSTA sub-domain 29

7.2 Manually dialled call to a device outside the CSTA sub-domain 31

7.3 Make Call service - busy called device is outside the CSTA sub-domain 33

8 External Incoming Call Scenarios **35**

8.1 External incoming call (no network information) 35

8.2 External incoming call (with network information) 36
8.3 External incoming call to a busy device (with network information) 37

9 Forwarding Call Scenarios 39
9.1 Call forward - no answer 39
9.2 Call forward - immediate 40
9.3 Call forward - immediate (with Diverted events) 41
9.4 Call forward - busy 42
9.5 Call forward - busy (with Diverted events) 43

10 Call Movement Scenarios 45
10.1 Deflect Call service 45
10.2 Directed Pickup Call service 46
10.3 Group Pickup Call service 47
10.4 Manual group pick up 48
10.5 Park Call service 50

11 Holding/Retrieving Call Scenarios 51
11.1 Hold Call service 51
11.2 Retrieve Call service 52

12 Consultation Call Scenarios 53
12.1 Consultation Call service 53
12.2 Manual consultation call 55
12.3 Consultation Call service (negative acknowledgement) 56
12.4 Consultation Call service - consulted party is busy 57
12.5 Consulted party disconnects using the Clear Connection service 59
12.6 Held party disconnects using the Clear Connection service 60
12.7 Reconnect Call service 61
12.8 Alternate Call service 63
12.9 Manual alternate call 64

13 Transfer Call Scenarios 65
13.1 Transfer Call service - screened transfer (with fixed view in Transferred event) 65
13.2 Transfer Call service - screened transfer (with local view in Transferred event) 66
13.3 Transfer Call Call service - blind transfer (with local view in Transferred event) 67
13.4 Single Step Transfer Call service 68

14 Conference Call Scenarios 71
14.1 Conference Call service 71
14.2 Single Step Conference Call service 72

15 Call Completion Scenarios **73**

15.1 Call Back Call-Related service - called device is busy 73

15.2 Camp On Call service 75

16 Distributing Call Scenarios **77**

16.1 Incoming Call to ACD with no available agents 77

1 Scope

This Technical Report illustrates call scenarios for Services for Computer Supported Telecommunications Applications (CSTA) Phase III (ECMA-269).

The scenarios are only for information and as such ECMA-269 (Services) and ECMA-285 (Protocol) Standards may define additional options or parameters. The purpose of this Technical Report is to provide examples of some CSTA Service invocations and illustrate associated call event reports. It is not an exhaustive document and some implementations may not perform as illustrated within this document, while still conforming to the Standard.

Each scenario includes a textual description and an illustration. Illustrations use the same key as described within ECMA-269. For each scenario, message sequences are listed for all device type monitored devices - call type monitors have not been illustrated. All devices have device type monitors set with no events masked. The columns in each scenario represent the following:

- The Activity column includes a brief description of the telephony activity. The activity can either be initiated by a service invocation or manually.
- The Monitored Device(s) columns list events generated for the specified device-type monitor or a service request and service response.
- The Comments column describes additional information on the activity.

All mandatory parameters is CSTA messages are provided. In addition, all conditional parameters that are required in the context of the scenario are provided. Optional parameters are generally not included unless they are useful in the context of illustrating a specific scenario. The mandatory, conditional, and optional classification of parameters in CSTA messages are specified in ECMA-269.

The monitorCrossRefID parameter in events is not shown.

DeviceIDs are illustrated by Dn and ConnectionIDs in the form DnCn. All Device IDs are within the same switching sub-domain unless otherwise indicated or stated. Any exception comments are made in the final column Comments.

2 References

ECMA-269	Services for Computer Supported Telecommunications Applications (CSTA) Phase III, 4th edition (June 2000)
ECMA-285	Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III, 2nd edition (June 2000)
ECMA TR/72	Glossary of definitions and terminology for Computer Supported Telecommunications Applications (CSTA) Phase III, 3rd edition (June 2000)

3 Definitions and Abbreviations

The definitions and abbreviations used in this Technical Report are defined in ECMA TR/72.

4 Call Origination Scenarios

This clause includes examples of how calls can be initiated, either by using the Make Call service or by manual operation.

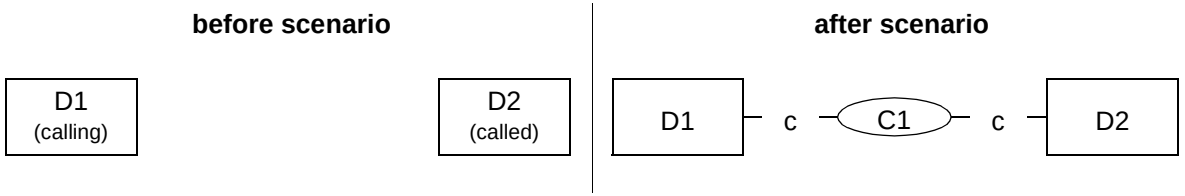
The first flow illustrates how a call is originated using the Make Call service. In this flow the calling device is prompted to go off-hook, and then the call is established between two devices.

Additional call origination flows are provided that illustrate how to originate a call using the Make Call service with hands free dialling, manual dialling, multi-stage dialling, and scenarios that show calls that fail, etc.

4.1 Make Call service - calling device is prompted to go off-hook

This scenario illustrates a successful Make Call from device D1 to device D2. In this scenario both devices are available and valid, device D1 is permitted to make the call and the call is answered by device D2.

In this scenario the Make Call service specifies that device D1 should be prompted to go off-hook (via the autoOriginate parameter) before D2 device is called.



Activity	Monitored Device D1		Monitored Device D2		Comments
A Make Call service to a valid device is invoked on behalf of device D1.	MakeCallRequest • callingDevice • calledDirectoryNumber • autoOriginate	 D1 D2 Prompt			The Make Call service specifies that device D1 should be prompted to go off-hook.
Acknowl-edgement.	MakeCallResult • initiatedCall	 D1C1			
Indication that the service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection • initiatedDevice • localConnectionState • cause	 D1C1 D1 Initiated makeCall			The generation of this event is switch specific. The MakeCall cause indicates that the device D1 is being prompted (via ringing, for example) to go off-hook.

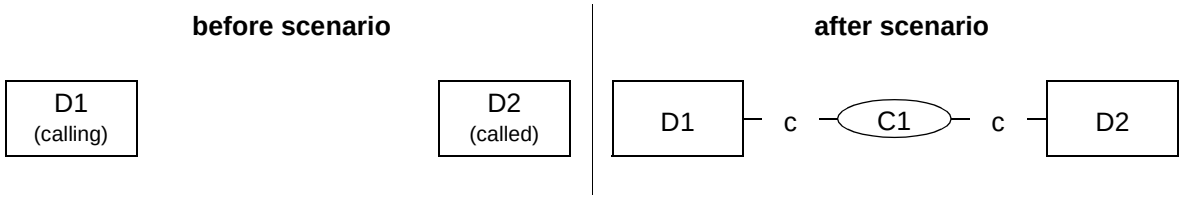
Activity	Monitored Device D1		Monitored Device D2			Comments
Device D1 goes off hook and is connected in the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall				
Device D2 begins to ring and D1 receives ringing tone.	DeliveredEvent • connection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D2C1 D2 D1 D2 NR Connected newCall	DeliveredEvent • connection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D2C1 D2 D1 D2 NR Alerting newCall		
Device D2 answers the call by manually going off-hook.	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D2C1 D2 D1 D2 NR Connected newCall	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D2C1 D2 D1 D2 NR Connected newCall		

4.2 Make Call service - calling device is in hands free mode

This scenario illustrates the case when the *calling device* is requested to automatically connect to the call ("hands free" mode).

This scenario differs from the first scenario in the following ways:

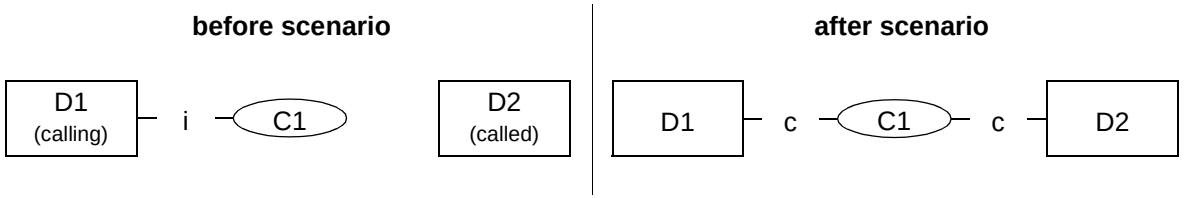
- The Make Call service request (via the autoOriginate parameter) specifies that the calling device should be automatically connected to the call ("hands free") mode.
- The NewCall cause on the Service Initiated event indicates that the device is not being prompted to go off-hook.



Activity	Monitored Device D1	Monitored Device D2		Comments
A MakeCall service to a valid device is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDirectoryNumber D2 • autoOriginate DoNotPrompt			The autoOriginate parameter specifies that the calling device should be automatically connected to the call (not prompted).
Acknowl-edgement.	MakeCallResult • initiatedCall D1C1			
Indication that the service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection D1C1 • initiatedDevice D1 • localConnectionState Initiated • cause NewCall			The generation of this event is switch specific. The Service Initiated event with the NewCall cause indicates that there is no prompting at the device.
Device D1 is automatically connected to the call.	OriginatedEvent • originatedConnection D1C1 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause newCall			Since the autoOriginate parameter indicates "DoNot-Prompt", device D1 is connected to the call without manual inter-vention (hands free mode).
	...scenario proceeds as shown in 4.1.			

4.3 Make Call service - calling device is already off-hook

This scenario illustrates the invoking of a call that already was initiated by a user going off-hook on a telephone. The call is invoked from device D1 to device D2.



Activity	Monitored Device D1	Monitored Device D2		Comments
Device D1 manually initiates a call by going off-hook.	ServiceInitiatedEvent • initiatedConnection D1C1 • initiatedDevice D1 • localConnectionState Initiated • cause newCall			
MakeCall service is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDevice D2			
Acknowl-edgement.	MakeCallResult • initiatedCall D1C1			
Call proceeds from device D1.	OriginatedEvent • originatedConnection D1C1 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause newCall			
	...scenario proceeds as shown in 4.1.			

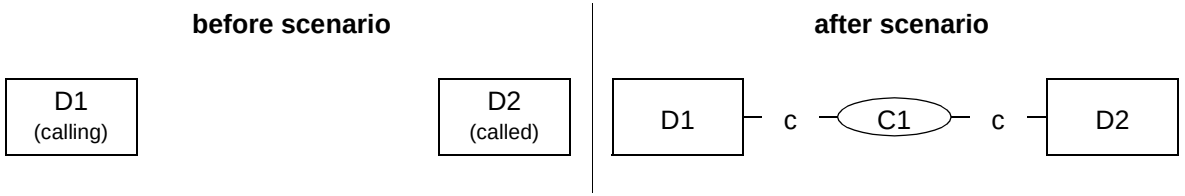
4.4 Manually dialled call

This scenario illustrates a call originated through manual device activity.

The scenario differs from the first scenario in the following ways:

- The Make Call service is not included.
- The cause on the Service Initiated event does indicate prompting.

Note that in this scenario the implementation buffers the dialled digits until the complete dialling sequence has been dialled and provides the complete dialled digits in the Originated event (i.e., no Digits Dialled event(s) are provided in this scenario).

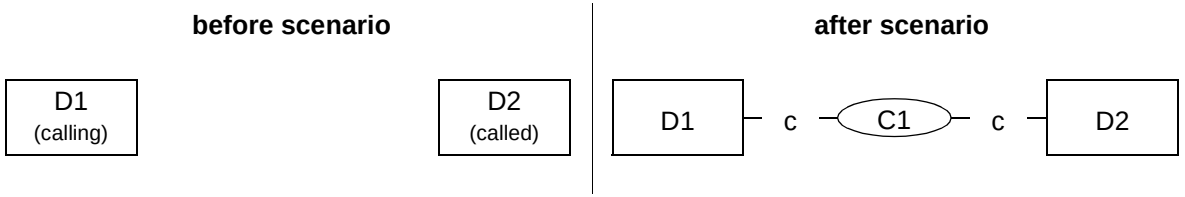


Activity	Monitored Device D1		Monitored Device D2		Comments
User at Device D1 goes off-hook and receives dial tone.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall			
Device D1 completes dialling device D2 and is connected to the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			
	...scenario proceeds as shown in 4.1.				

4.5 Manually dialled call showing individual digits dialled

This scenario differs from the previous scenarios because it illustrates how an individual Digits Dialled event is generated for each digit dialled. After all digits are dialled the Originated event provides the complete dialled sequence.

Note that it is implementation specific how many digits are buffered before they are sent in a Digits Dialled event, or if the digits are buffered until the complete sequence of digits is dialled (i.e., no Digits Dialled events prior to an Originated event).

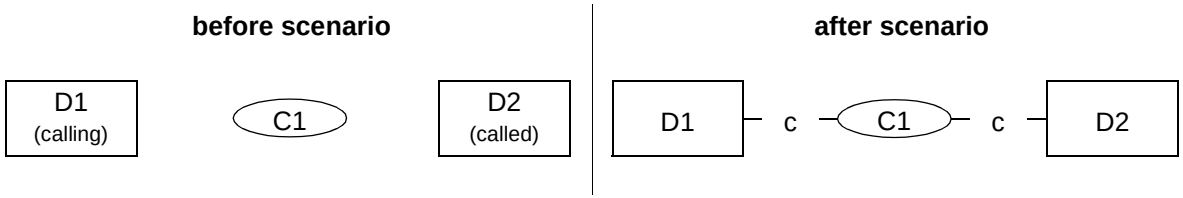


Activity	Monitored Device D1		Monitored Device D2		Comments
User at Device D1 goes off-hook and receives dial tone.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall			
Digit “2” is dialled.	DigitsDialledEvent • diallingConnection • diallingDevice • diallingSequence • localConnectionState • cause	 D1C1 D1 “2” Initiated normal			Digit “2” is dialled.
Digit “3” is dialled.	DigitsDialledEvent • diallingConnection • diallingDevice • diallingSequence • localConnectionState • cause	 D1C1 D1 “3” Initiated normal			Digit “3” is dialled.
Digit “4” is dialled.	DigitsDialledEvent • diallingConnection • diallingDevice • diallingSequence • localConnectionState • cause	 D1C1 D1 “4” Initiated normal			Digit “4” is dialled.
Digit “3” is dialled.	DigitsDialledEvent • diallingConnection • diallingDevice • diallingSequence • localConnectionState • cause	 D1C1 D1 “3” Initiated normal			Digit “3” is dialled.

Activity	Monitored Device D1	Monitored Device D2		Comments
Device D1 has completed dialling and is connected to the call.	OriginatedEvent • originatedConnection D1C1 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause newCall			All of the digits for D2 (“2343”) have been dialled. The Originated event contains the complete dialling sequence.
	...scenario proceeds as shown in 4.1.			

4.6 Dialling using Dial Digits service

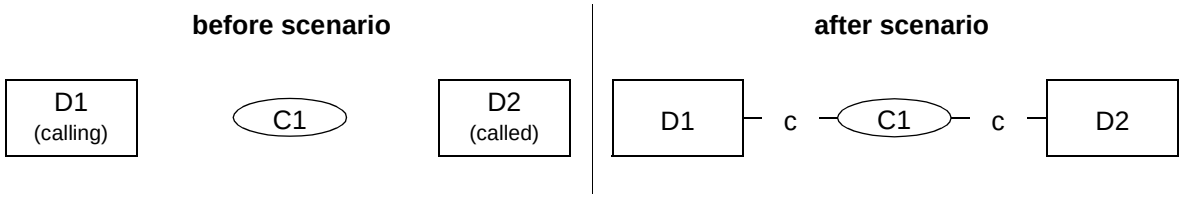
This scenario illustrates the use of the Dial Digits service for a call that has already been established by the user manually going off-hook.



Activity	Monitored Device D1	Monitored Device D2		Comments
Device D1 manually initiates a call by going off-hook.	ServiceInitiatedEvent • initiatedConnection D1C1 • initiatedDevice D1 • localConnectionState Initiated • cause newCall			
The Dial Digits service with the complete dialling sequence is provided.	DialDigitsService • diallingConnection D1C1 • diallingSequencee D2			
Acknowl-edgement.	DialDigitsResult			
The event indicates that the requested digits were dialled.	DialDigitsEvent • diallingConnection D1C1 • diallingDevice D1 • diallingSequence D2 • localConnectionState Initiated • cause normal			
The dialling sequence is complete and device D1 is connected in the call.	OriginatedEvent • originatedConnection D1C1 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause newCall			
	...scenario proceeds as shown in 4.1.			

4.7 Multi-stage dialling

This scenario illustrates the use of the Dial Digits service to complete dialling a call that was established via a Make Call service.



Activity	Monitored Device D1	Monitored Device D2		Comments
A MakeCall service to a valid device is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDirectoryNumber "2;" • autoOriginate Prompt			The Make Call service specifies a partial dialling string that begins with a ("2") and the partial dialling indicator (";").
Acknowl-edgement.	MakeCallResult • initiatedCall D1C1			
Indication that the service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection D1C1 • initiatedDevice D1 • localConnectionState Initiated • cause makeCall			The generation of this event is switch specific. The MakeCall cause indicates that the device D1 is being prompted (via ringing, for example) to go off-hook.
The event indicates that a partial dialling sequence was received.	DialDigitsEvent • diallingConnection D1C1 • diallingDevice D1 • diallingSequence "2;" • localConnectionState Initiated • cause normal			A ";" character indicates that there is an incomplete dialling string.
The Dial Digits service with the remainder of the dialling sequence is provided.	DialDigitsService • diallingConnection D1C1 • diallingSequencee "3456"			A ";" is not provided in the dialling string since there are no more digits to be dialled.
Acknowl-edgement.	DialDigitsResult			

Activity	Monitored Device D1		Monitored Device D2		Comments
The event indicates that the requested digits were dialled.	DialDigitsEvent • diallingConnection • diallingDevice • diallingSequence • localConnectionState • cause	 D1C1 D1 “3456” Initiated normal			A complete dialling sequence has been received (no “;” character).
The dialling sequence is complete and device D1 is connected in the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			D2 is the called device. It contains the digits “23456” in this scenario.
	...scenario proceeds as shown in 4.1.				

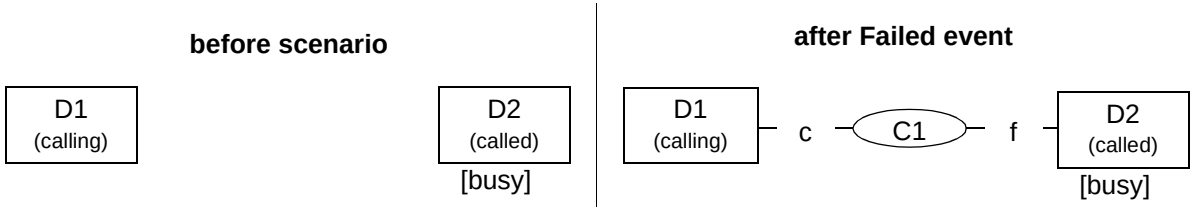
4.8 Make Call service - called device is busy

This scenario illustrates a Make Call from device D1 to device D2, where device D2 is busy and is not set to forward busy calls. The call fails because the called party is busy.

This scenario differs from the first scenario in the following ways:

- The Failed event is generated to indicate that the call has encountered a busy device.

Note that in this example the Make Call service is successful (positive acknowledgement) and the events indicate that device D2 is busy. Another possible scenario is where the Make Call service is negatively acknowledged with an error code indicating that device D2 is in an invalid state.



Activity	Monitored Device D1		Monitored Device D2		Comments
MakeCall service is invoked on behalf of device D1.	MakeCallRequest • callingDevice • calledDirectoryNumber • autoOriginate	 D1 D2 Prompt			
Acknowl-edgement.	MakeCallResult • initiatedCall	 D1C1			
Device D1 notified of initi-ating call.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated MakeCall			The generation of this event is switch specific.
The call is being attempted from device D1.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			

Activity	**Monitored Device D1**	**Monitored Device D2**		**Comments**
Device D2 is busy - the call cannot be completed. Device D1 hears busy tone.	FailedEvent • failedConnection D2C1 • failingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause busy	FailedEvent • failedConnection D2C1 • failingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Failed • cause busy		This illustrates connection failures that report the Failed event for all devices involved with the call and that will provide a complete connectionID for the failed connection. See ECMA-269, clause 2.8.2, item 2.
Device D1 replaces handset.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Failed • cause normalClearing		
Failed connection D2C1 also clears.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing		

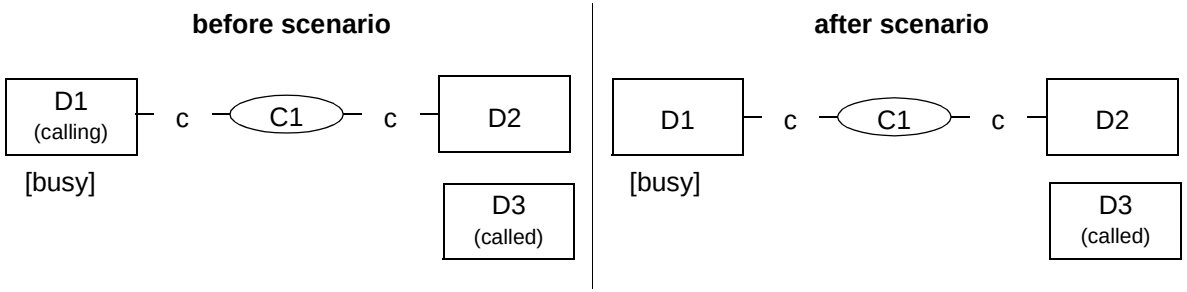
4.9 Make Call service - call attempted to a busy calling device (negative acknowledgement)

This scenario illustrates a Make Call from device D1 to device D3, where device D1 is busy. The call fails because the calling device is busy.

This scenario differs from the first scenario in the following ways:

- The (negative) response to the Make Call service request indicates that the call attempt has failed. No subsequent events are generated.

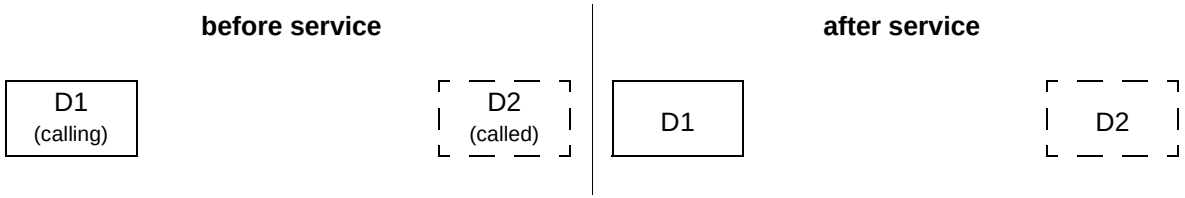
The Make Call request is negatively acknowledged because the calling party, device D1, is busy when a Make Call request is issued.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A MakeCall service is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDevice D3			
Negative Acknowl-edgement.	MakeCallError • stateIncompatibility invalid calling device state			Make Call service fails because calling party is busy.

4.10 Make Call service - called number is an invalid number (negative acknowledgement)

This scenario illustrates a Make Call from device D1 to device D2. In this scenario device D1 is available, valid and permitted to make the call. Device D2 (illustrated by a box with a dotted line around it) is actually an invalid number (e.g., the number is correctly formatted but it is not part of the dialling plan) and the call fails.

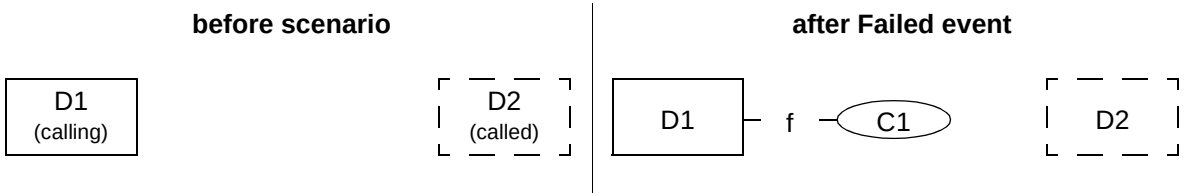


Activity	Monitored Device D1	Monitored Device D2		Comments
MakeCall service is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDevice D2			
Negative Acknowl-edgement.	MakeCallError • operationalError InvalidCalled-Device			

4.11 Manually dialled call - dialled number is invalid

This scenario illustrates a manually dialled call from device D1 to device D2. In this scenario device D1 is available, valid and permitted to make the call. Device D2 (illustrated by a box with a dotted line around it) is actually an invalid number (e.g., the number is correctly formatted but it is not part of the dialling plan) and the call fails.

Note that in this scenario the dialled digits are buffered in the switching function until the complete dialling sequence has been dialled and is providing the complete dialled digits in the Originated event (i.e., no Digits Dialled event(s) are provided in this scenario).



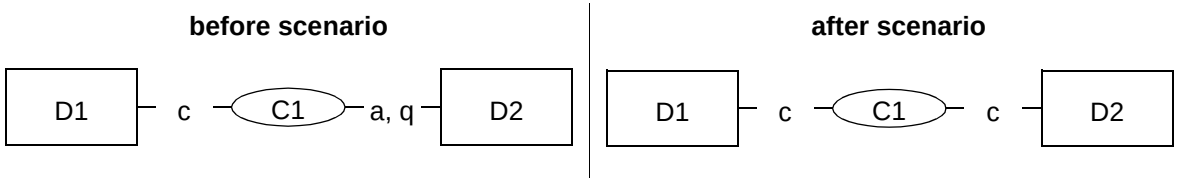
Activity	Monitored Device D1		Monitored Device D2		Comments
Device D1 goes off-hook and receives dial tone.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall			
Device D1 completes dialling device D2 and is connected to the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			
Device D2 is an invalid number the call cannot be completed. Device D1 receives reorder tone.	FailedEvent • failedConnection • failingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D1C1 D1 D1 D2 NR Failed reorderTone			
Device D1 goes on-hook.	ConnectionClearedEvent • droppedConnection • releasingDevice • localConnectionState • cause	 D1C1 D1 Null normalClearing			

5 Answering Call Scenarios

This clause illustrates how calls are answered, manually and by CSTA services.

5.1 Answer Call service

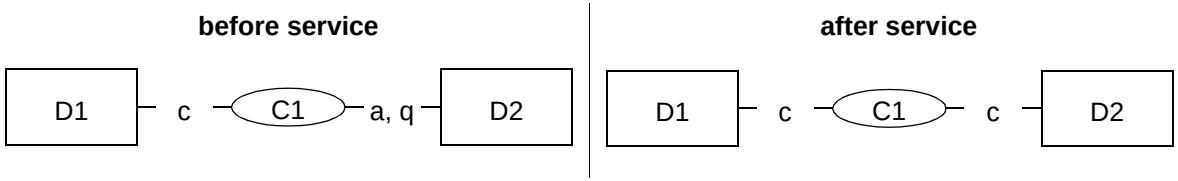
This scenario illustrates the successful use of the Answer Call service invoked on behalf of device D2.



Activity	Monitored Device D1	Monitored Device D2		Comments
Answer Call service is invoked on behalf of device D2.		AnswerCallRequest • callToBeAnswered	D2C1	An error will be returned if the device is not able to answer the call without manual intervention.
Acknowl-edgement.		AnswerCallResult		
Device D2 has been answered.	EstablishedEvent • establishedConnection D2C1 • answeringDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause newCall	EstablishedEvent • establishedConnection D2C1 • answeringDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause newCall		

5.2 Manually answering a call

This scenario illustrates the event sequence when alerting device D2 goes off-hook to answer the call.



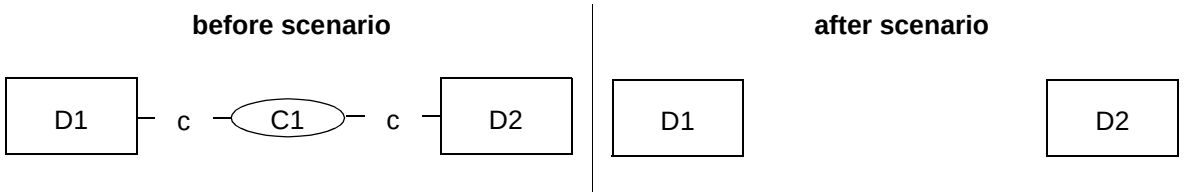
Activity	Monitored Device D1		Monitored Device D2			Comments
Device D2 answers the call by going off-hook.	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D2C1 D2 D1 D2 NR Connected newCall	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 D2C1 D2 D1 D2 NR Connected newCall		

6 Call and Connection Termination Scenarios

This clause illustrates how calls and connections are ended.

6.1 Device disconnects from a call by going on-hook (remaining device is cleared from the call)

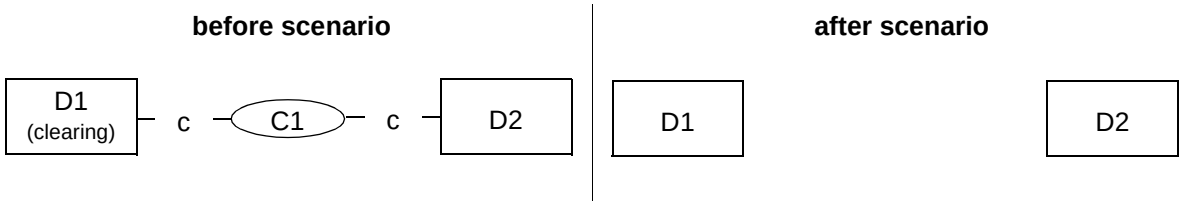
The user at device D1, while connected to device D2, replaces the handset.



Activity	Monitored Device D1	Monitored Device D2		Comments
D1 goes on-hook.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing		
Since D2 is the only device in the call, it is cleared as the result of D1 being cleared.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing		

6.2 Device disconnects from a call by going on-hook (remaining device goes blocked)

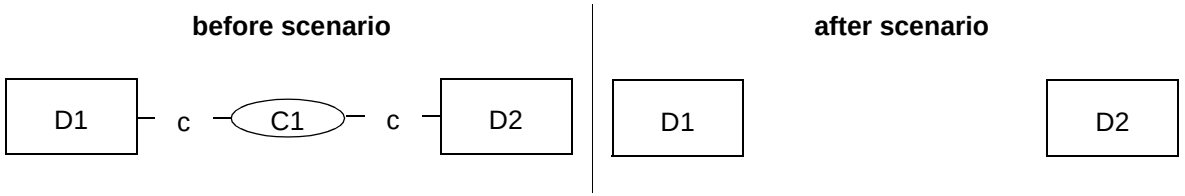
In this scenario device D1 is manually put on-hook to release itself from the call. The remaining device goes blocked, until the device goes on-hook.



Activity	Monitored Device D1	Monitored Device D2		Comments
Device D1 goes on-hook.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing		
As a result of the “far end disconnect”, the remaining connection D2C1 goes blocked.		FailedEvent • failedConnection D2C1 • failingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Failed • cause blocked		While D2C1 is blocked, the user at the device may be hearing error tone.
The remaining device goes on-hook.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing		

6.3 Device disconnects from a call using the Clear Connection service (remaining device is cleared)

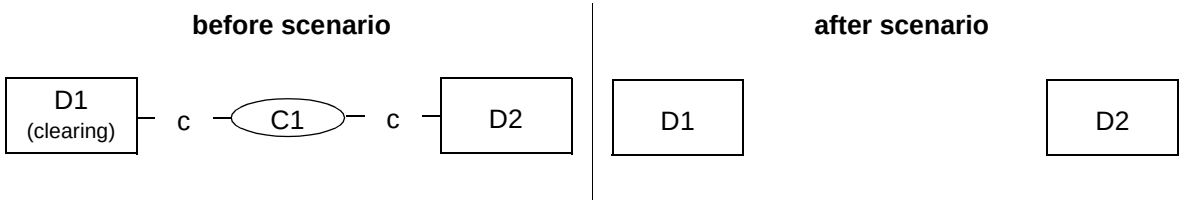
The Clear Connection service is used to disconnect device D1 from the call. In this example, the remaining device in the call, D2, is cleared after D1 has been removed from the call.



Activity	Monitored Device D1	Monitored Device D2		Comments
A Clear Connection service is invoked on D1s behalf.	ClearConnectionRequest • connectionToBeCleared D1C1			
Acknowl-edgement.	ClearConnectionResult			
Event indicates that connection D1C1 has been removed from the call.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing		
Since D2 is the only device in the call, it is cleared as the result of D1 being cleared.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing		

6.4 Device disconnects from a call using the Clear Connection service (remaining device goes blocked)

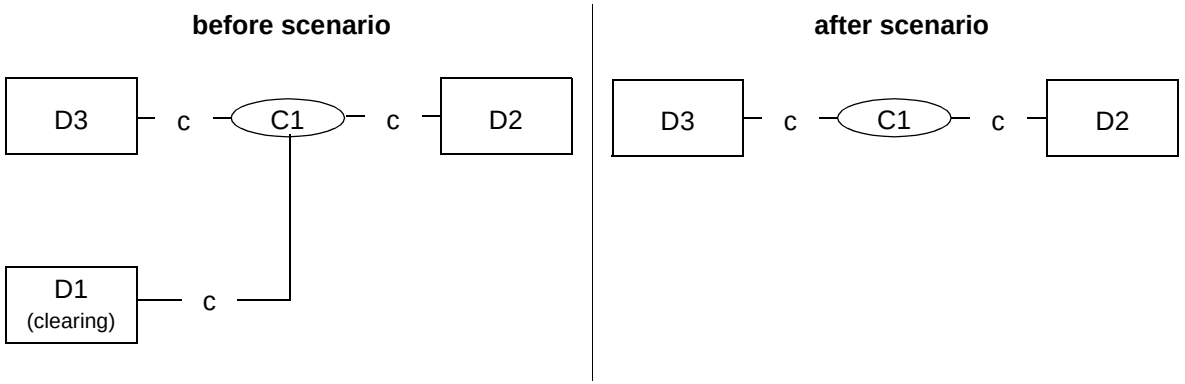
The Clear Connection service is used to disconnect device D1 from the call. In this example, the remaining device in the call, D2, is goes blocked, until it is cleared from the call by manually going on-hook.



Activity	Monitored Device D1	Monitored Device D2		Comments
A Clear Connection service is invoked on D1s behalf.	ClearConnectionRequest • connectionToBeCleared D1C1			
Acknowl-edgement.	ClearConnectionResult			
Event indicates that connection D1C1 has been removed from the call.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing		
As a result of the “far end disconnect”, the remaining connection D2C1 goes blocked.		FailedEvent • failedConnection D2C1 • failingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Failed • cause blocked		While blocked, the user at the device may be hearing error tone.
The remaining device goes on-hook.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing		

6.5 Device disconnects from a conference call using the Clear Connection service

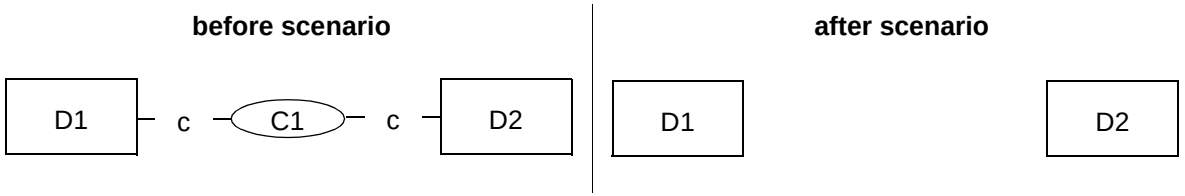
This service releases a specific device from a call. In the case of a Conference Call this results in the specific party being removed from this conference.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A Clear Connection service is invoked.	ClearConnectionRequest • connectionToBeCleared D1C1			
Acknowl-edgement.	ClearConnectionResult			
Event indicates that connection D1C1 has been removed from the call.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing	Devices D2 and D3 remain connected in the call.

6.6 Clearing a two-party call using the Clear Call service

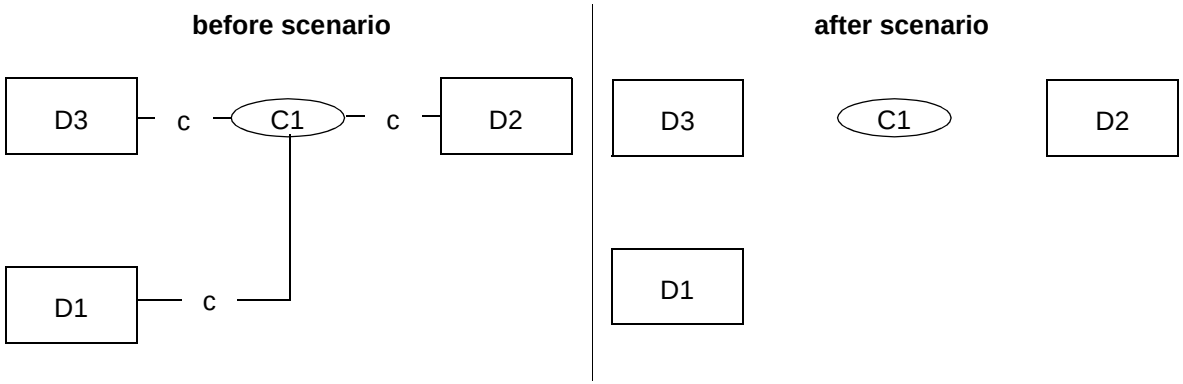
This scenario illustrates the use of the Clear Call service. This service releases all devices from an existing call.



Activity	Monitored Device D1		Monitored Device D2			Comments
A Clear Call service is invoked.	ClearCallRequest • callToClear	D1C1				
Acknowl-edgement.	ClearCallResult					
The events indicate that D1 has discon-nected from the call.	ConnectionClearedEvent • droppedConnection • releasingDevice • localConnectionState • cause	 D1C1 D1 Null normalClearing	ConnectionClearedEvent • droppedConnection • releasingDevice • localConnectionState • cause	 D1C1 D1 Connected normalClearing		
The events indicate that D2 has discon-nected from the call.			ConnectionClearedEvent • droppedConnection • releasingDevice • localConnectionState • cause	 D2C1 D2 Null normalClearing		

6.7 Clearing a conference call using the Clear Call service

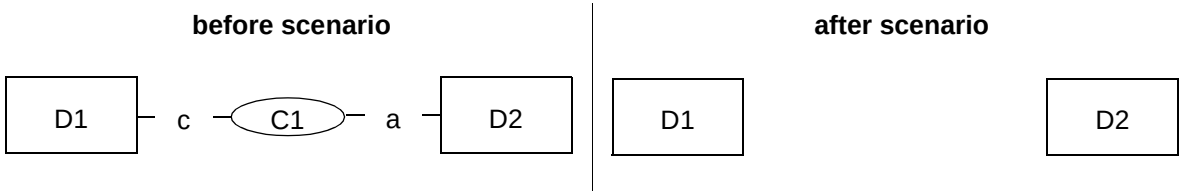
This scenario illustrates the use of the Clear Call service. This service releases all devices from an existing conference call.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A Clear Call service is invoked.	ClearCallRequest • callToClear D1C1			
Acknowl-edgement.	ClearCallResult			
D1s connection to the call is cleared.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Connected • cause normalClearing	D1C1s Connection Cleared event is reported for all device-type monitors in the call.
D2s connection to the call is cleared.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause normalClearing	ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Connected • cause normalClearing	D2C1s Connection Cleared event is reported for all devices remaining in the call.
The remaining connection is cleared.			ConnectionClearedEvent • droppedConnection D3C1 • releasingDevice D3 • localConnectionState Null • cause normalClearing	

6.8 Call is cleared after an alerting time-out

In this scenario, the call is cleared as the result of an alerting timer expiry.



Activity	Monitored Device D1	Monitored Device D2		Comments
D2C1 is cleared as the result of an alert timer expiry.	ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Connected • cause callNotAn-swered	ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • cause callNotAn-swered		The cause of “callNotAnswered” indicates that the call was cleared as the result of a timer expiry.
Connection D1C1 clears as a result of D1 clearing.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • cause normalClearing			

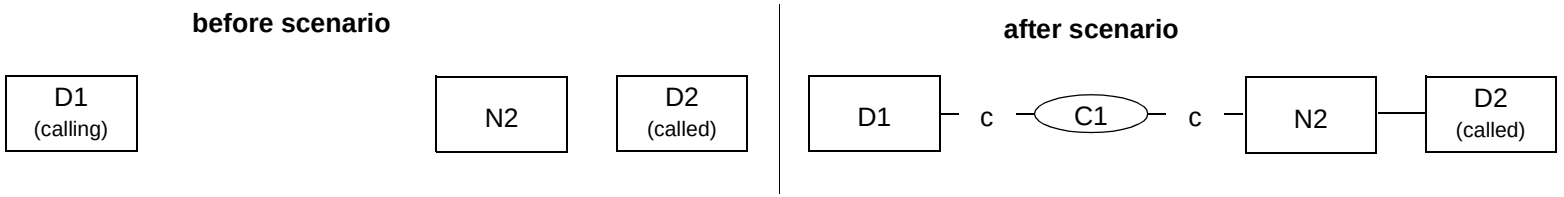
7 External Outgoing Call Scenarios

This section includes examples of successful external outgoing calls, initiated manually and by CSTA services.

7.1 Make Call service - called device is outside the CSTA sub-domain

This scenario illustrates a Make Call service request on behalf of device D1 to the device D2 which is outside the CSTA sub-domain.

Since device D2 is located outside this CSTA sub-domain, it can not be directly monitored through this CSTA interface and therefore no events will be seen for that device. However, device N2, which is a network interface device (NID) (e.g., trunk interface), acts as a proxy for device D2, and depending upon the type of signalling available via the external network, some signalling information can be made available through the connection of the NID.

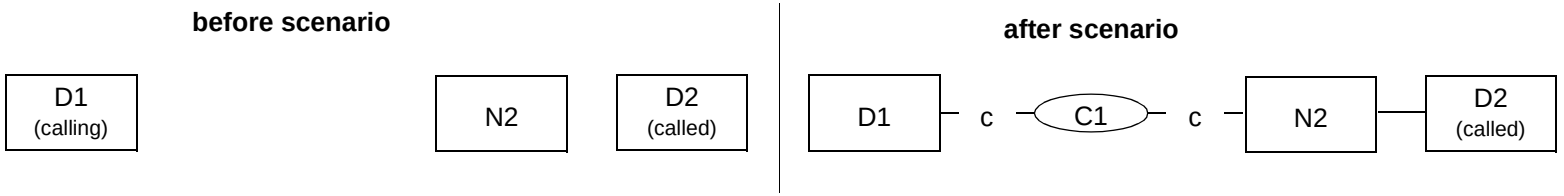


Activity	Monitored Device D1		Monitored Device N2		Comments
A Make Call service to a valid device outside the CSTA sub-domain is invoked on behalf of device D1.	MakeCallRequest • callingDevice • calledDevice • autoOriginate	 D1 D2 doNotPrompt			The service request specifies “hands free” mode (See 4.2).
Acknowl-edgement.	MakeCallResult • initiatedCall	 D1C1			
Indication that service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall			The generation of this event is switch specific.
D1 is connected to the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			

Activity	Monitored Device D1		Monitored Device N2			Comments
The call leaves the CSTA sub-domain.	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 D2 NR Connected newCall	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 D2 NR Connected newCall		
Device D2 is alerted.	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2		Receiving this event depends on the type of network interface. The cause of NetworkSignal indicates that the event is due to activity at the device located outside of the CSTA switching sub-domain (D2), not the NID (N2).
Device D2 answers the call.	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 D2C1 D2 D1 D2 NR Connected networkSignal N2	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 D2C1 D2 D1 D2 NR Connected networkSignal N2		Answer supervision is received from the network (this depends upon the type of signalling supported by the network). The cause of NetworkSignal indicates that the event is due to activity at the device located outside of the CSTA switching sub-domain (D2), not the NID (N2).

7.2 Manually dialled call to a device outside the CSTA sub-domain

In this scenario the device D1 is manually lifted to initiate a call, and the call is routed out of the CSTA sub-domain. The user dials a trunk access code and the NID is selected. Then the user completes dialling the external number.



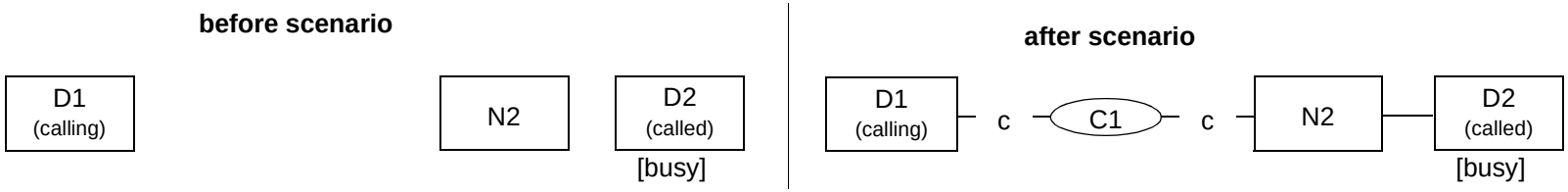
Activity	Monitored Device D1		Monitored Device N2			Comments
User at Device D1 goes Off-Hook and dials the trunk access code.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall				
The call leaves the CSTA sub-domain.	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 NK NR Initiated newCall	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 NK NR Connected newCall		
User at Device D1 completes dialling device D2 and D1 is connected to the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause • assoc.CalledDevice	 D1C1 D1 D2 Connected newCall N2	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause • assoc.CalledDevice	 D1C1 D1 D2 Connected newCall N2		
Device D2 is alerted.	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2		Receiving this event depends on the type of network interface. The cause of NetworkSignal indicates that the event is due to activity at the device located outside of the CSTA switching sub-domain (D2), not the NID (N2).

Activity	Monitored Device D1		Monitored Device N2			Comments
Device D2 answers the call.	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 D2C1 D2 D1 D2 NR Connected networkSignal N2	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 D2C1 D2 D1 D2 NR Connected networkSignal N2		Answer supervision is received from the network (this depends upon the type of signalling supported by the network). The cause of NetworkSignal indicates that the event is due to activity at the device located outside of the CSTA switching sub-domain (D2), not the NID (N2).

7.3 Make Call service - busy called device is outside the CSTA sub-domain

This scenario illustrates a Make Call service request on behalf of device D1 to a busy device D2 outside the CSTA sub-domain.

Event information after the Network Reached event depends on the type of the network interface.



Activity	Monitored Device D1		Monitored Device N2			Comments
A Make Call service to a valid device outside the CSTA sub-domain is invoked on behalf of device D1.	MakeCallRequest • callingDevice • calledDevice • autoOriginate	 D1 D2 doNotPrompt				
Acknowl-edgement.	MakeCallResult • initiatedCall	 D1C1				
Indication that service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • cause	 D1C1 D1 Initiated newCall				
D1 is connected to the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall				
The call leaves the CSTA sub-domain.	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 D2 NR Connected newCall	NetworkReachedEvent • outboundConnection • networkInterfaceUsed • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause	 N2C1 N2 D1 D2 NR Connected newCall		

Activity	Monitored Device D1		Monitored Device N2			Comments
Device D2 is busy - the call cannot be completed. Device D1 receives busy tone.	FailedEvent • failedConnection • failingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2	FailedEvent • failedConnection • failingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • cause • assoc.CalledDevice	 N2C1 D2 D1 D2 NR Connected networkSignal N2		Receiving this event depends on the type of network interface. The cause of NetworkSignal indicates that the event is due to activity at the device located outside of the CSTA switching sub-domain (D2), not the NID (N2). A cause code of Busy may also be used.

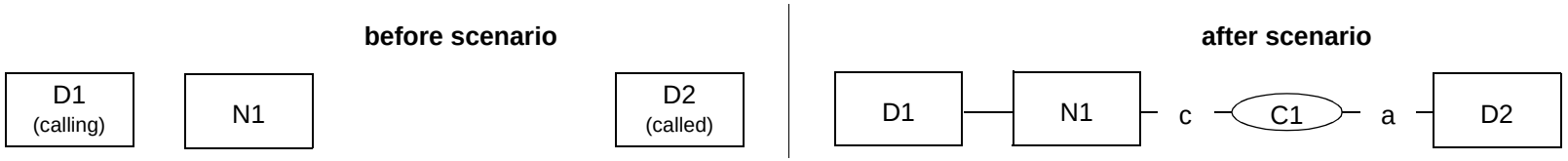
8 External Incoming Call Scenarios

This section includes examples of successful external incoming calls.

8.1 External incoming call (no network information)

This scenario illustrates the successful incoming call from device D1. Because device D1 is located outside the CSTA sub-domain, it can not be monitored and therefore events will be seen only for the devices N1 (NID - Network Interface Device, e.g., trunk) and D2.

In this scenario, no calling party or called party information is passed over the NID. The called device is determined via a dedicated trunk device.

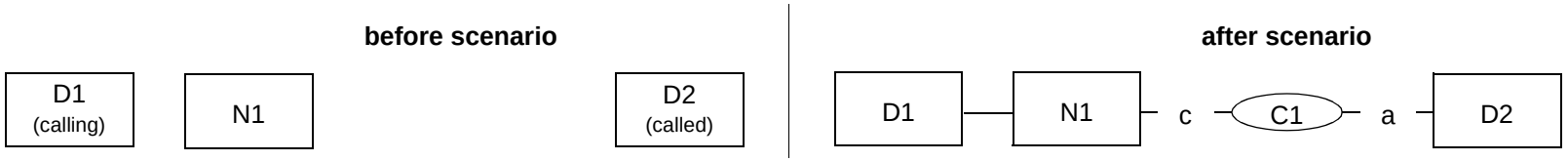


Activity	Monitored Device N1	Monitored Device D2		Comments
Indicates an external incoming call on the NID (e.g. trunk).	ServiceInitiatedEvent • initiatedConnection N1C1 • initiatingDevice N1 • localConnectionState Initiated • eventCause newCall • assoc.CallingDevice N1			
The NID has connected in the call.	OriginatedEvent • originatedConnection N1C1 • callingDevice NK • calledDevice D2 • localConnectionState Connected • cause newCall • assoc.CallingDevice N1			
Device D2 is available and is alerted.	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice NK • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • eventCause newCall • assoc.CallingDevice N1	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice NK • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause newCall • assoc.CallingDevice N1		In this scenario, the network does not provide calling and called device information.

8.2 External incoming call (with network information)

This scenario illustrates the successful incoming call from device D1.

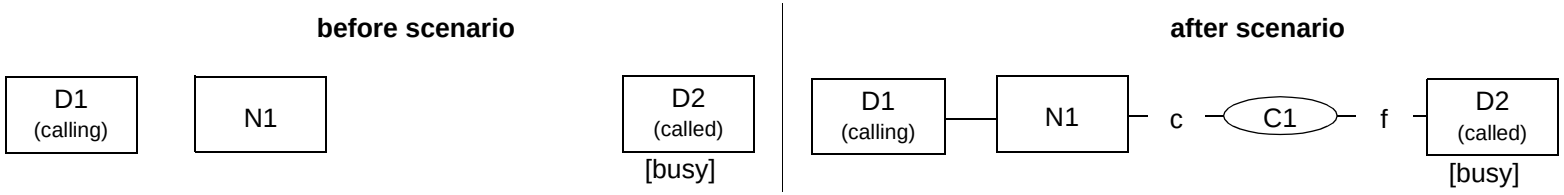
In this scenario, the network provides the calling and called party information over the NID.



Activity	Monitored Device N1	Monitored Device D2		Comments
Indicates an external incoming call on the NID (e.g. trunk).	ServiceInitiatedEvent • initiatedConnection N1C1 • initiatingDevice N1 • localConnectionState Initiated • eventCause newCall • assoc.CallingDevice N1			
The network has provided the calling and called device information over the NID.	OriginatedEvent • originatedConnection N1C1 • callingDevice D1 • calledDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • localConnectionState Connected • cause newCall • assoc.CallingDevice N1			The networkCallingDevice and the networkCalledDevice parameters will not change as long as N1 is involved with the call. The callingDevice and the calledDevice parameters may change as the result of features.
Device D2 is available and begins to ring.	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • eventCause newCall • assoc.CallingDevice N1	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause newCall • assoc.CallingDevice N1		

8.3 External incoming call to a busy device (with network information)

This scenario illustrates an external incoming call service to a busy device. The calling device D1 is located outside the CSTA sub-domain.



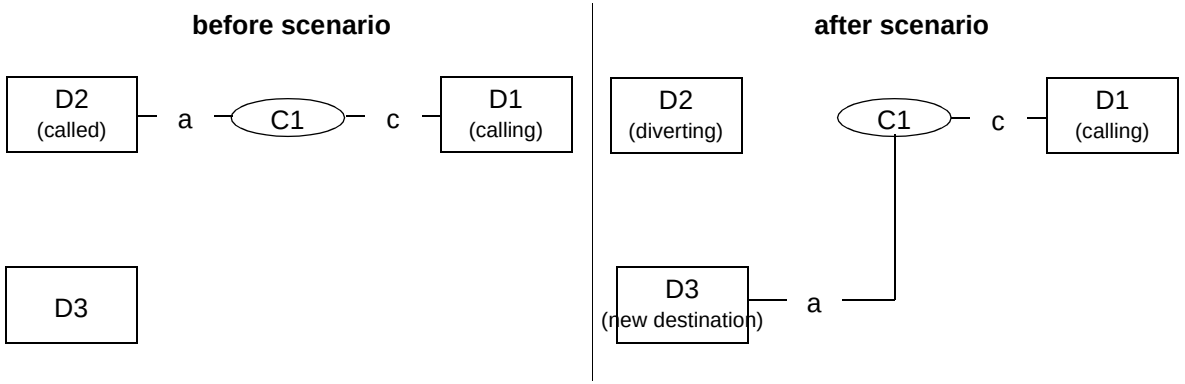
Activity	Monitored Device N1		Monitored Device D2			Comments
Indicates an external incoming call on the NID (e.g. trunk).	ServiceInitiatedEvent • initiatedConnection • initiatingDevice • localConnectionState • eventCause • assoc.CallingDevice	 N1C1 N1 Initiated newCall N1				
The NID is connected in the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • networkCallingDevice • networkCalledDevice • cause • assoc.CallingDevice	 N1C1 D1 D2 Connected D1 D2 newCall N1				The network has provided the calling and called devices.
Device D2 is busy - the call cannot be completed.	FailedEvent • failedConnection • failingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause • networkCallingDevice • networkCalledDevice • assoc.CallingDevice	 D2C1 D2 D1 D2 NR Connected busy D1 D2 N1	FailedEvent • failedConnection • failingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause • networkCallingDevice • networkCalledDevice • assoc.CallingDevice	 D2C1 D2 D1 D2 NR Failed busy D1 D2 N1		Called device D2 is busy.

9 Forwarding Call Scenarios

This section includes examples of successful forwarding calls, forwarding calls on busy, no answer and immediate.

9.1 Call forward - no answer

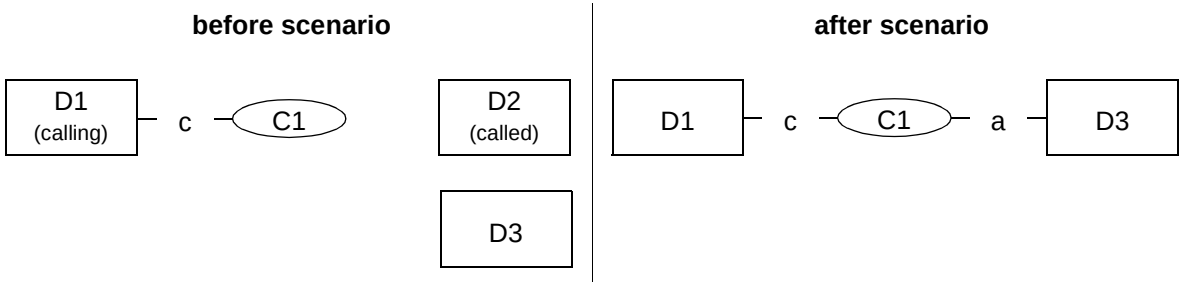
This scenario illustrates the flow for a basic call forward no answer. A call comes to a device which is set to forward calls to a predefined device after a specified number of rings.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D2 is alerted for a specified number of rings and then forwards the call to device D3.	DivertedEvent • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState connected • eventCause callForward-NoAnswer	DivertedEvent • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause callForward-NoAnswer		Device D3 is the device predefined by device D2 to forward its call. This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D2) monitor.
The Call is forwarded to device D3.	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Connected • eventCause callForward-NoAnswer		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Alerting • eventCause callForward-NoAnswer	

9.2 Call forward - immediate

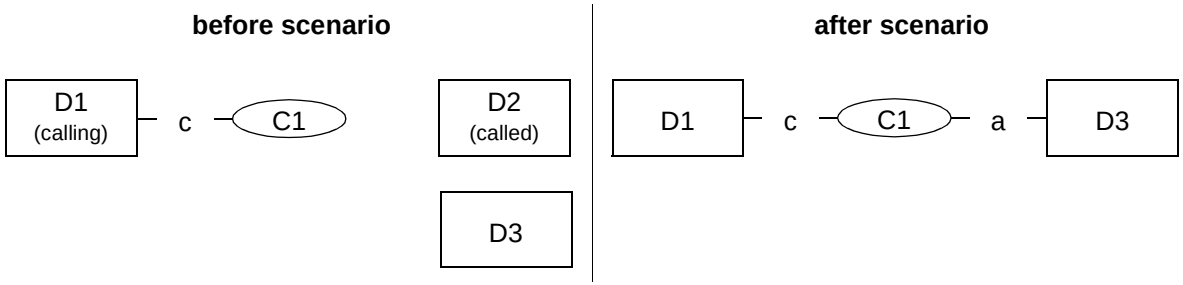
This scenario illustrates the flow for a basic call forward immediate. A call comes to a device which is set to forward calls immediately to a predefined device.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A call is initiated by device D1 to device D2. Device D1 is already connected to the call. D2 is set to forwarded calls immediately to device D3.	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Connected • eventCause forward-Immediate		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Alerting • eventCause forward-Immediate	In this example, the monitor for device D2 never receives an event with respect to this call.

9.3 Call forward - immediate (with Diverted events)

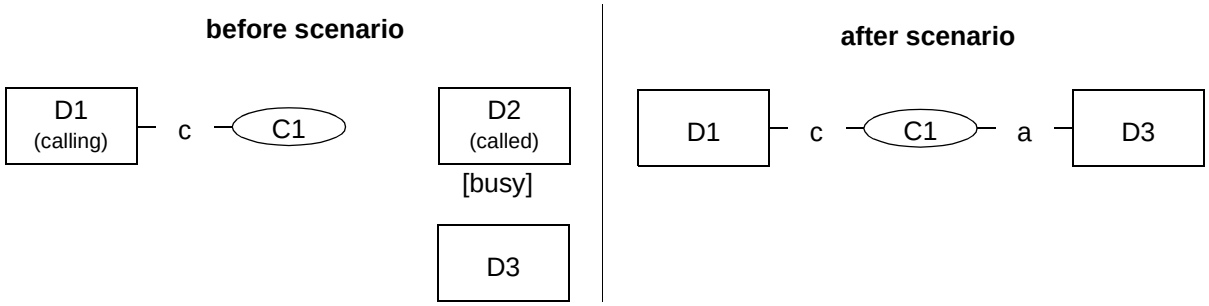
This scenario illustrates the flow for a basic call forward immediate. A call comes to a device which is set to forward calls immediately to a predefined device.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A call is initiated by device D1 to device D2. Device D1 is already connected to the call. D2 is set to forwarded calls immediately to device D3.	DivertedEvent (See 2nd bullet in the comments column) • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause forward-Immediate	DivertedEvent (See 1st bullet in the comments column) • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause forward-Immediate		This illustrates two different CSTA modeling options (as specified via the capability exchange services): • the forwarding model where forwarding is processed after a call arrives at the device. In this case the D2 monitor flows a Diverted event representing a Null/ Null connection state transition. • the Diverted event option where the Diverted event is being sent to all devices in the call, not just for the diverting device monitor. In this case the Diverted event flows on the D1 monitor.
The call is forwarded to device D3.	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Connected • eventCause forward-Immediate		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Alerting • eventCause forward-Immediate	In this example, the monitor for device D2 never receives an event with respect to this call.

9.4 Call forward - busy

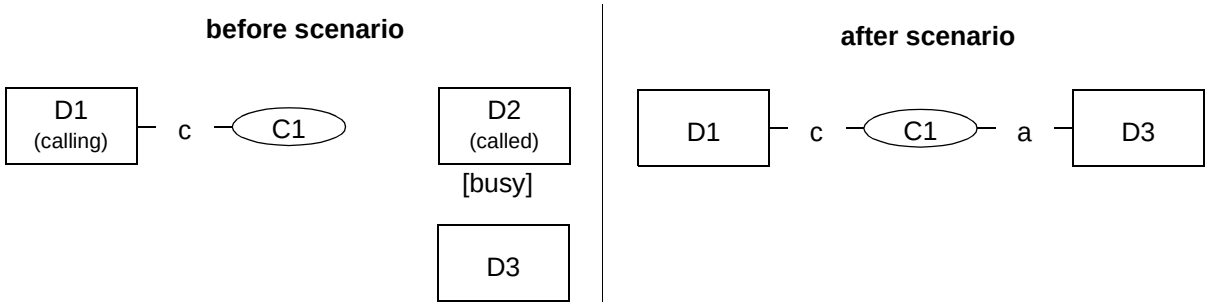
This scenario illustrates the flow for a basic call forward busy. A call comes to a device which is set to forward calls immediately to a predefined device if the called device is busy.



Activity	Monitored Device D1		Monitored Device D2	Monitored Device D3		Comments
Since Device D2 is busy on another call, when D1 calls D2 the call is forwarded to device D3.	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D3C1 D3 D1 D2 D2 Connected forwardBusy		DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D3C1 D3 D1 D2 D2 Alerting forwardBusy	In this example, the monitor for device D2 never receives an event with respect to this call.

9.5 Call forward - busy (with Diverted events)

This scenario illustrates the flow for a basic call forward busy. A call comes to a device which is set to forward calls immediately to a predefined device if the called device is busy.



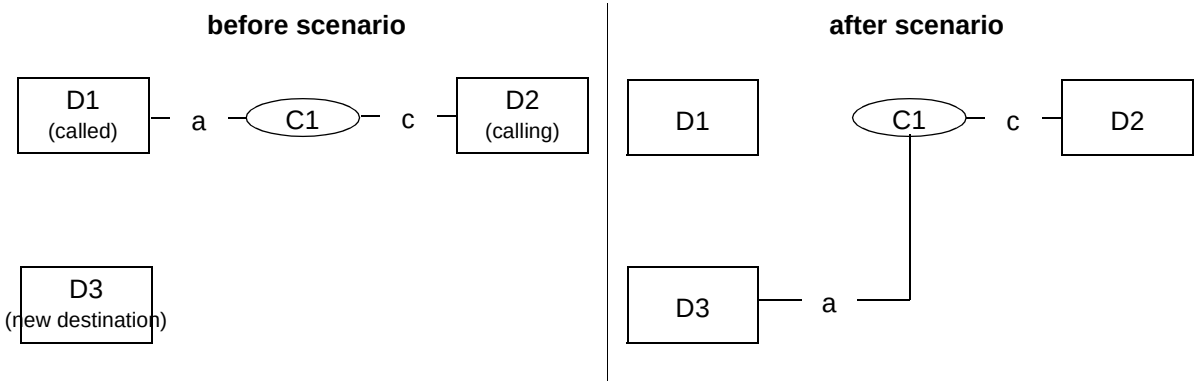
Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A call is initiated by device D1 to device D2. Device D2 is busy with another call and has busy forwarding set to forward calls to D3.	DivertedEvent (See 2nd bullet in the comments column) • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause forward-busy	DivertedEvent (See 1st bullet in the comments column) • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause forward-busy		This illustrates two different CSTA modeling options (as specified via the capability exchange services): • the forwarding model where forwarding is processed after a call arrives at the device. In this case the D2 monitor flows a Diverted event representing a Null/ Null connection state transition. • the Diverted event option where the Diverted event is being sent to all devices in the call, not just for the diverting device monitor. In this case the Diverted event flows on the D1 monitor.
The call is forwarded to device D3.	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Connected • eventCause forwardBusy		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice D2 • localConnectionState Alerting • eventCause forwardBusy	

10 Call Movement Scenarios

This section includes examples of moving calls from one device to another, initiated manually and by CSTA services.

10.1 Deflect Call service

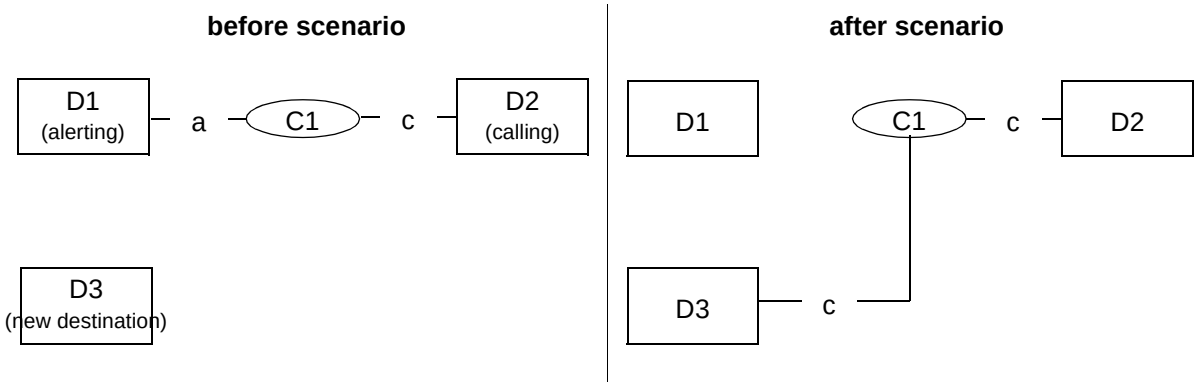
This scenario illustrates how an alerting call is diverted to another destination.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Deflect Call service is invoked on behalf of device D1.	DeflectCallRequest • callToBeDiverted D1C1 • newDestination D3			
Acknowl-edgement.	DeflectCallResult			
The event indicates that the call has been diverted from D1.	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause redirected	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause redirected		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D1) monitor.
The call is alerting D3.		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause redirected	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Alerting • eventCause redirected	

10.2 Directed Pickup Call service

This service illustrates how an alerting connection is moved from one device and connected to another device via the Directed Pickup service.

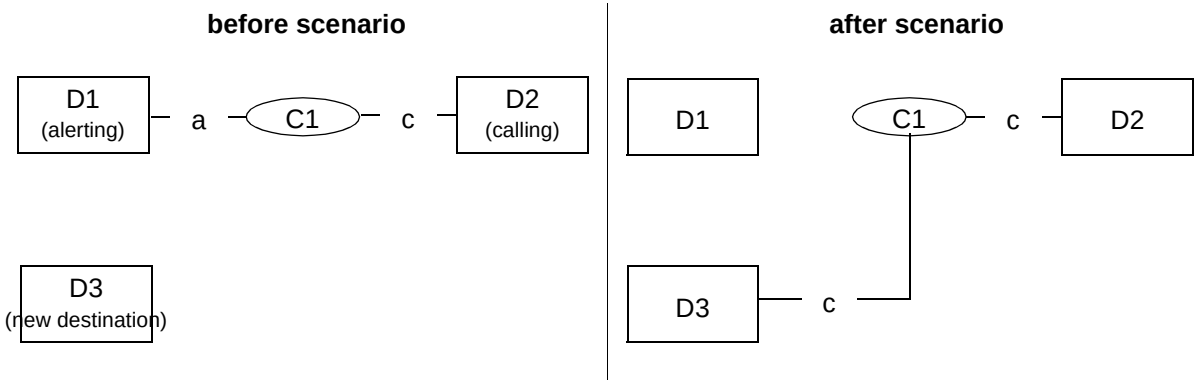


Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A Directed Pickup Call service is invoked.			DirectedPickupCallRequest • callToBePickedUp D1C1 • newDestination D3	
Acknowl-edgement.			DirectedPickupCallResult D3C1 • pickedCall	
The call is diverted from D1.	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause callPickup	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause callPickup		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D1) monitor.
The call is connected to D3.		EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	

10.3 Group Pickup Call service

This scenario illustrates a pickup of a call that is alerting at a device as a member of a specified or default pickup group.

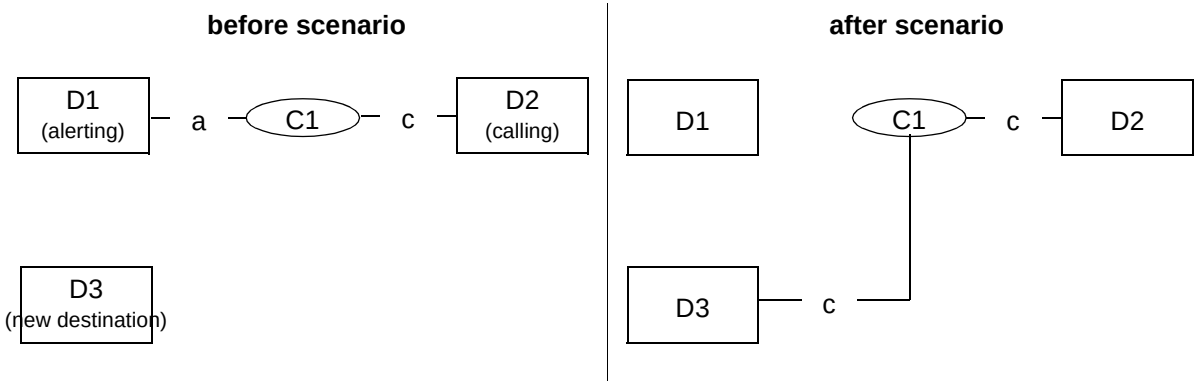
The call is moved and connected at the new specified destination.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
A Group Pickup Call service is invoked.			GroupPickupCallRequest • newDestination D3	Device D3 is in the same pickup group as device D1.
Acknowl-edgement.			GroupPickupCallResult	
The call has been diverted from device D1.	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause callPickup	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause callPickup		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D1) monitor.
The call is connected to device D3.		EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	

10.4 Manual group pick up

Device D2 has called device D1. Device D3 and device D1 are in the same pickup group. Device D3 is going Off-Hook and dialing a specific code number issuing the Group Pickup Call to answer the call.

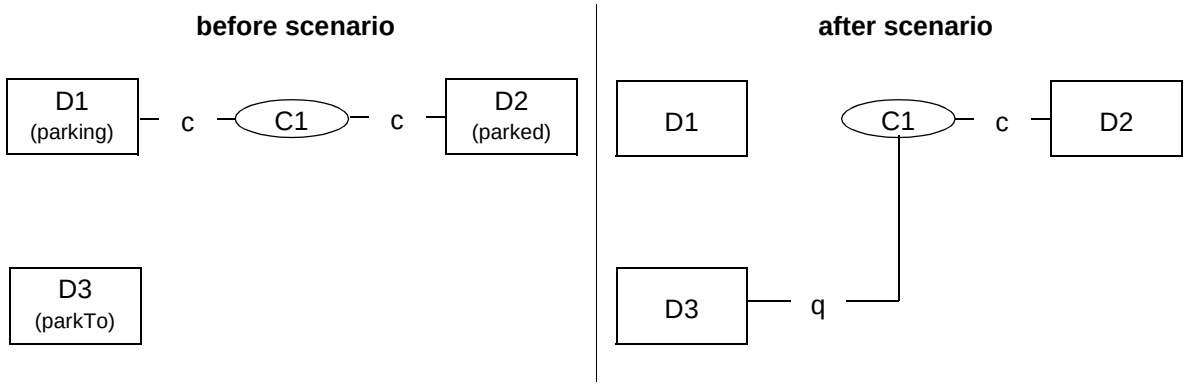


Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D3 goes Off-Hook in order to invoke a feature.			ServiceInitiatedEvent • initiatedConnection D3C2 • initiatingDevice D3 • localConnectionState Initiated • eventCause newCall	Device D3 is in the same pickup group as device D1. A manual pickup could also be invoked via a button on the phone.
The connection was cleared after the feature access code was entered.			ConnectionClearedEvent • droppedConnection D3C2 • releasingDevice D3 • localConnectionState Null • eventCause normalClearing	The feature could also have been invoked via a button on the phone.
The call is diverted from device D1.	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause callPickup	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause callPickup		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D1) monitor.

Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
The call is connected at device D3.		EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	EstablishedEvent • establishedConnection D3C1 • answeringDevice D3 • callingDevice D2 • calledDevice D1 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause callPickup	

10.5 Park Call service

This scenario illustrates how a connected call is parked at another device.



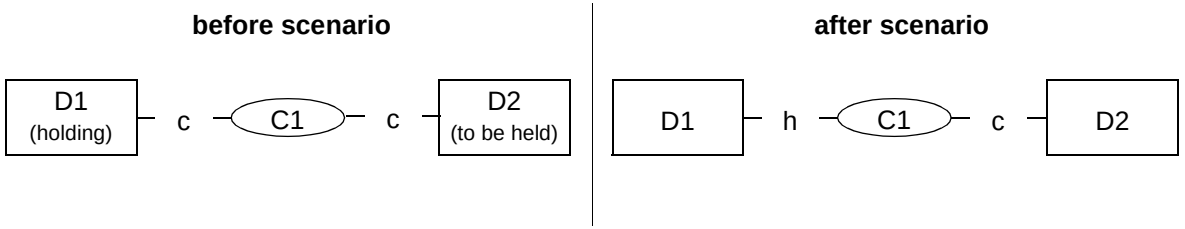
Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Park Call service is invoked on behalf of device D1.	ParkCallRequest • parking D1C1 • parkTo D3			
Acknowl-edgement.	ParkCallResult			
The event indicates that the call has been diverted from D1.	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause park	DivertedEvent • divertingConnection D1C1 • divertingDevice D1 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause park		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D1) monitor.
The call is parked at D3.		QueuedEvent • queuedConnection D3C1 • queue D3 • callingDevice D2 • calledDevice D3 • lastRedirectionDevice D1 • localConnectionState Connected • eventCause park	QueuedEvent • queuedConnection D3C1 • queue D3 • callingDevice D2 • calledDevice D3 • lastRedirectionDevice D1 • localConnectionState Queued • eventCause park	

11 Holding/Retrieving Call Scenarios

This section includes examples of successful Hold and Retrieve Call scenarios.

11.1 Hold Call service

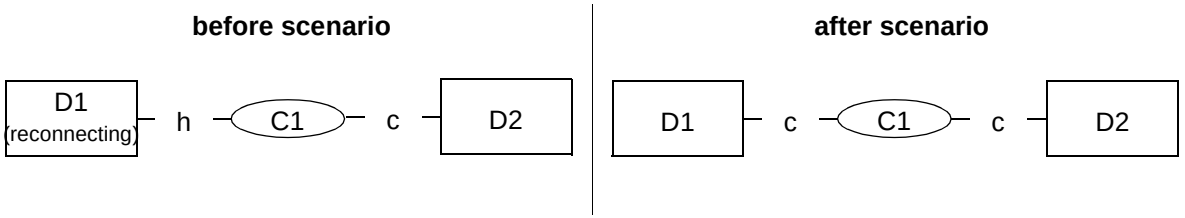
This scenario illustrates the successful use of a Hold Call service. The service places an existing connection on hold.



Activity	Monitored Device D1		Monitored Device D2			Comments
A Hold Call service is issued on behalf of D1.	HoldCallRequest • callToBeHeld	D1C1				
Acknowl-edgement.	HoldCallResult					
Connection placed on hold.	HeldEvent • heldConnection • holdingDevice • localConnectionState • eventCause	 D1C1 D1 Held normal	HeldEvent • heldConnection • holdingDevice • localConnectionState • eventCause	 D1C1 D1 Connected normal		

11.2 Retrieve Call service

This service reconnects to a call that has previously been placed on hold.



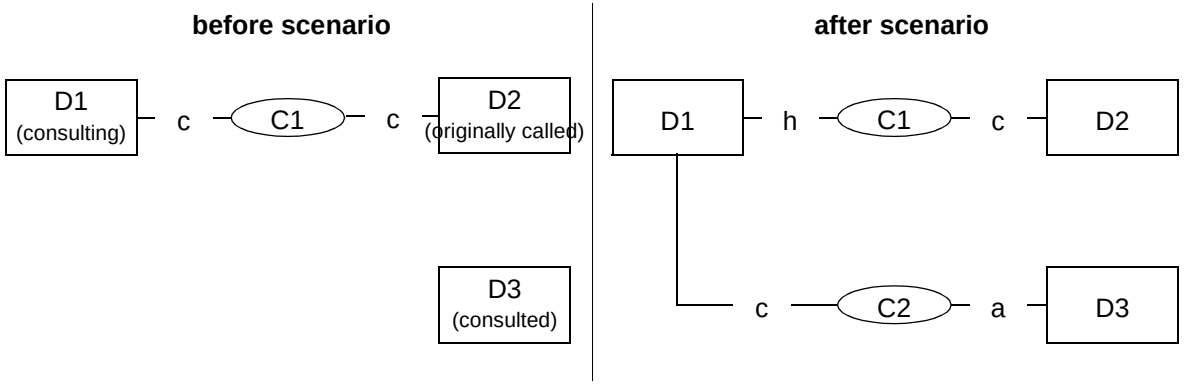
Activity	Monitored Device D1		Monitored Device D2			Comments
Device D1 issues the Retrieve Call service to retrieve the held call.	RetrieveCallRequest • heldCall	 D1C1				
Acknowl-edgement.	RetrievedCallResult					
Device D2 is connected back into the previ-ously held call.	RetrievedEvent • retrievedConnection • retrievingDevice • localConnectionState • eventCause	 D1C1 D1 Connected normal	RetrievedEvent • retrievedConnection • retrievingDevice • localConnectionState • eventCause	 D1C1 D1 Connected normal		

12 Consultation Call Scenarios

This section illustrates examples of successful Consultation Calls, Reconnect Calls and Alternate Calls initiated manually and by CSTA services.

12.1 Consultation Call service

This service places an existing active call at a device on hold and initiates a new call from the same device.

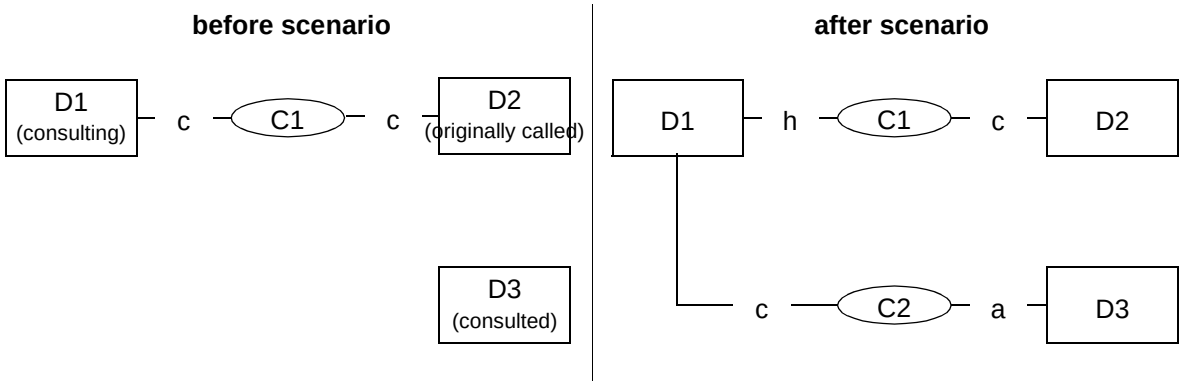


Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Consultation Call service to device D3 is invoked.	ConsultationCallRequest • existingCall D1C1 • consultedDevice D3			
Acknowl-edgement.	ConsultationCallResult • initiatedCall D1C2			
Connection placed on hold.	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Held • eventCause consultation	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Connected • eventCause consultation		
Consultation call is initiated.	ServiceInitiatedEvent • initiatedConnection D1C2 • initiatingDevice D1 • localConnectionState Initiated • eventCause consultation			The generation of this event is switch specific.

Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D1 is connected in the call.	OriginatedEvent • originatedConnection D1C2 • callingDevice D1 • calledDevice D3 • localConnectionState Connected • eventCause consultation			
Device D3 begins to ring.	DeliveredEvent • connection D3C2 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause newCall		DeliveredEvent • connection D3C2 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause newCall	

12.2 Manual consultation call

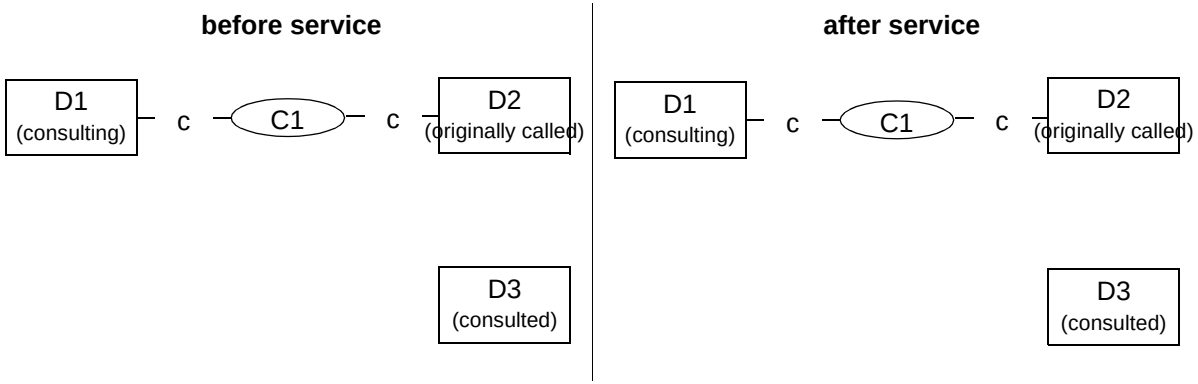
Device D1 is connected with device D2. Device D1 manually places device D2 on hold and creates a new call to device D3.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D1 manually places device D2 on hold.	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Held • eventCause normal	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Connected • eventCause normal		
A new connection is created at device D1.	ServiceInitiatedEvent • initiatedConnection D1C2 • initiatingDevice D1 • localConnectionState Initiated • eventCause newCall			The generation of this event is switch specific.
Device D1 is connected in the call.	OriginatedEvent • originatedConnection D1C2 • callingDevice D1 • calledDevice D3 • localConnectionState Connected • eventCause newCall			
Device D2 is available and begins to ring.	DeliveredEvent • alertingConnection D3C2 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause newCall		DeliveredEvent • alertingConnection D3C2 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause newCall	

12.3 Consultation Call service (negative acknowledgement)

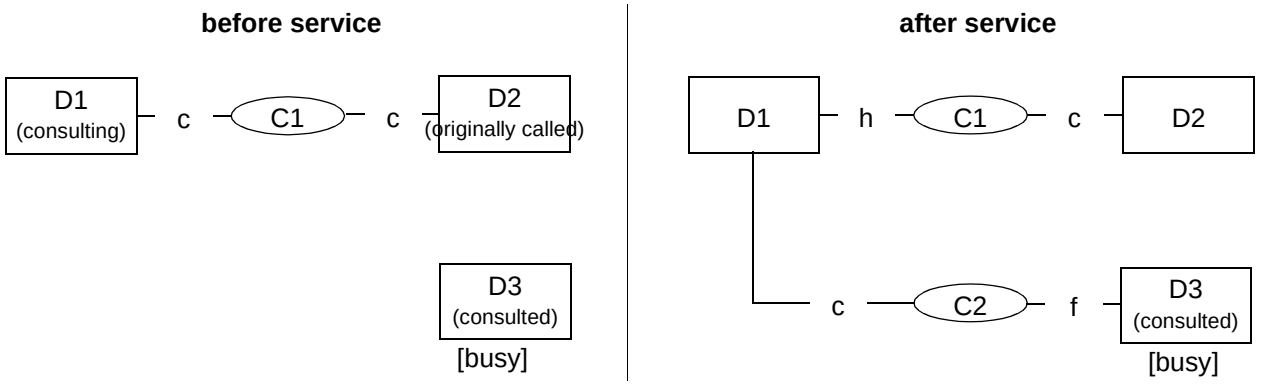
Device D1, which already has an active call, invokes the Consultation Call service which provides the compound action of the Hold Call service and the Make Call service. The Hold Call request may be rejected due to service feature restrictions.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Consultation Call service to device D3 is invoked on behalf of device D1.	ConsultationCallRequest • existingCall D1C1 • consultedDevice D3			
Negative Acknowl-edgement.	ConsultingCallError • operationalError invalidFeature			The possible error categories and error values are described in ECMA-269.

12.4 Consultation Call service - consulted party is busy

This scenario illustrates a Consultation Call service request invoked on behalf of device D1 to the device D3 that is busy. The original call is successfully placed on hold.

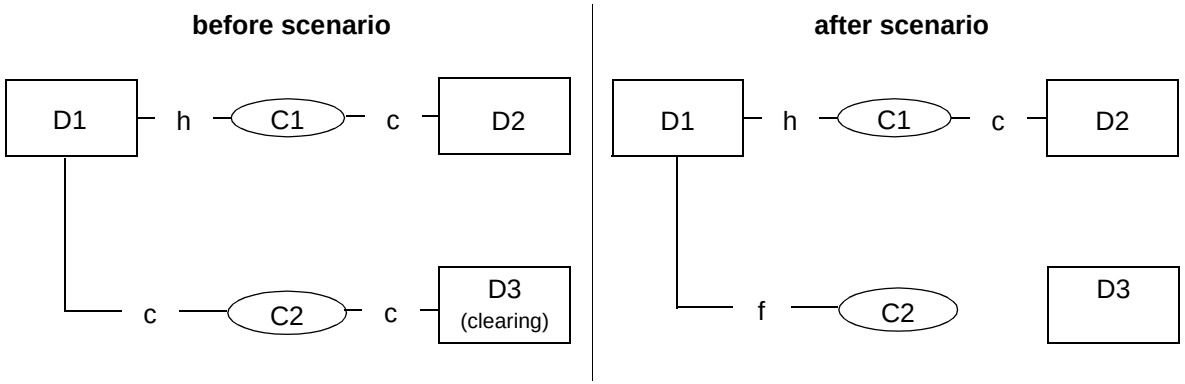


Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Consultation Call service to device D3 is invoked.	ConsultationCallRequest • existingCall D1C1 • consultedDevice D3			
Acknowl-edgement.	ConsultationCallResult • initiatedCall D1C2			
The D1C1 connection is placed on hold.	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Held • eventCause consultation	HeldEvent • heldConnection D1C1 • holdingDevice D1 • localConnectionState Connected • eventCause consultation		
A new call is initiated.	ServiceInitiatedEvent • initiatedConnection D1C2 • initiatingDevice D1 • localConnectionState Initiated • eventCause consultation			The generation of this event is switch specific.
Device D1 is connected in the call.	OriginatedEvent • originatedConnection D1C2 • callingDevice D1 • calledDevice D3 • localConnectionState Connected • eventCause consultation			

Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
The new call will not be delivered, because device D3 is busy in another call.	FailedEvent • failedConnection D3C2 • failingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause busy		FailedEvent • failedConnection D3C2 • failingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Failed • eventCause busy	This scenario illustrates the CSTA modeling option where the Failed event is generated for the busy device, with a complete connectionID.

12.5 Consulted party disconnects using the Clear Connection service

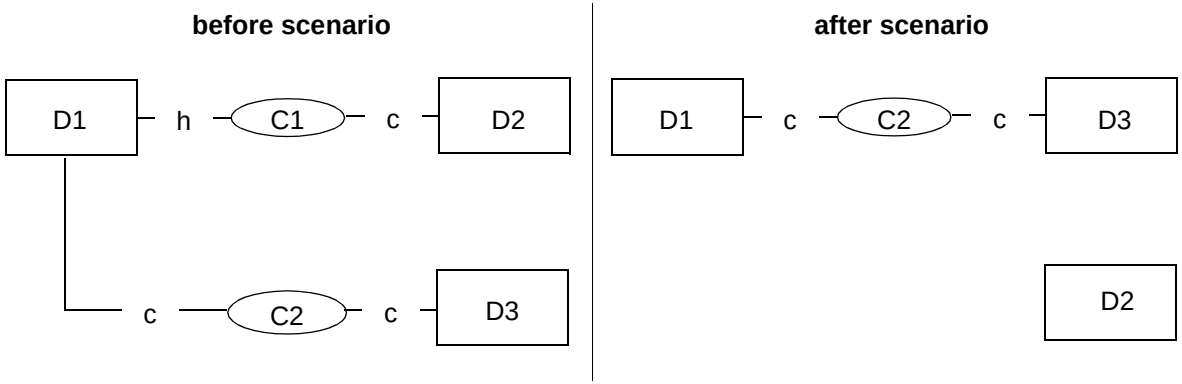
This scenario illustrates the successful use of the Clear Connection service. The Clear Connection service request is issued for the consulted party after the consulted device D3 is connected into the new call.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D3 wants to clear from the call and a Clear Connection service is invoked.			ClearConnectionRequest • connectionToBeCleared D3C2	
Acknowl-edgement.			ClearConnectionResult	
Connection D3C2 is cleared.	ConnectionClearedEvent • droppedConnection D3C2 • releasingDevice D3 • localConnectionState Connected • eventCause normalClearing		ConnectionClearedEvent • droppedConnection D3C2 • releasingDevice D3 • localConnectionState Null • eventCause normalClearing	
Connection D1C2 fails as a result of the clearing of D3.	FailedEvent • failedConnection D1C2 • failingDevice D1 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Failed • eventCause blocked			In this example, the user at D1 hears, as a result of the far end disconnect, error tone, until D1 goes on-hook.

12.6 Held party disconnects using the Clear Connection service

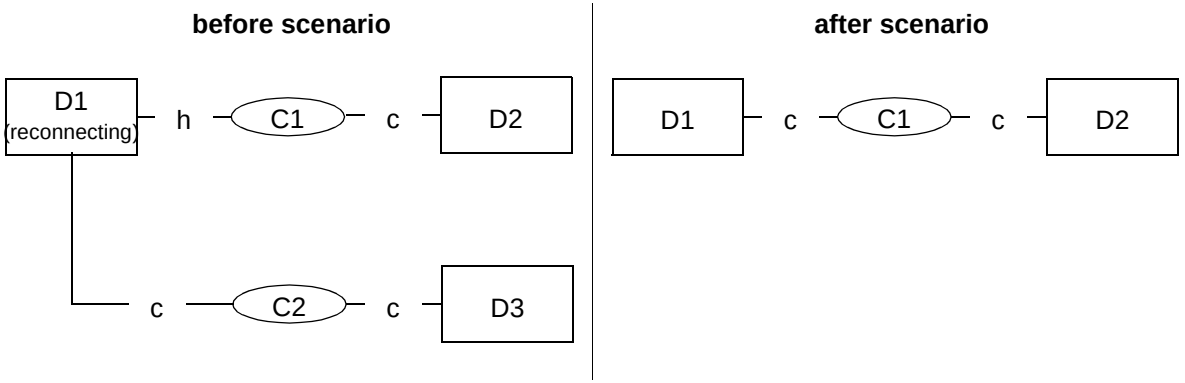
This scenario illustrates the successful use of the Clear Connection service. The Clear Connection service request is issued for the held party after the consulted device D3 is connected into the new call.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D2 wants to clear from the call and a Clear Connection service is invoked.		ClearConnectionRequest • connectionToBeCleared D2C1		
Acknowl-edgement.		ClearConnectionResult		
Connection D2C1 is cleared.	ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Held • eventCause normalClearing	ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • eventCause normalClearing		
As a result of the D2C1 clearing, D1C1 is also cleared.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • eventCause normalClearing			

12.7 Reconnect Call service

This service clears an existing connection and then retrieves a previously held connection at the same device.

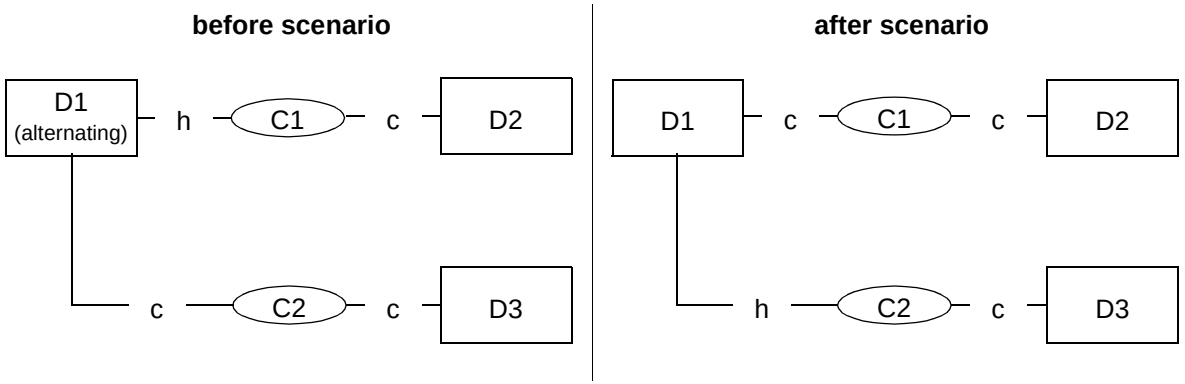


Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Device D1 has finished consultation and issues the Reconnect Call service to drop from the active call and retrieve the held call.	ReconnectCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	ReconnectCallResult			
Device D1 is cleared from the active call.	ConnectionClearedEvent • droppedConnection D1C2 • releasingDevice D1 • localConnectionState Null • eventCause normalClearing		ConnectionClearedEvent • droppedConnection D1C2 • releasingDevice D1 • localConnectionState Connected • eventCause normalClearing	
Device D2 is connected back into the previ-ously held call.	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause normal	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause normal		

Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
As the result of D1C2 clearing from the call, D3C2 is also cleared.			ConnectionClearedEvent • droppedConnection D3C2 • releasingDevice D3 • localConnectionState Null • cause normalClearing	

12.8 Alternate Call service

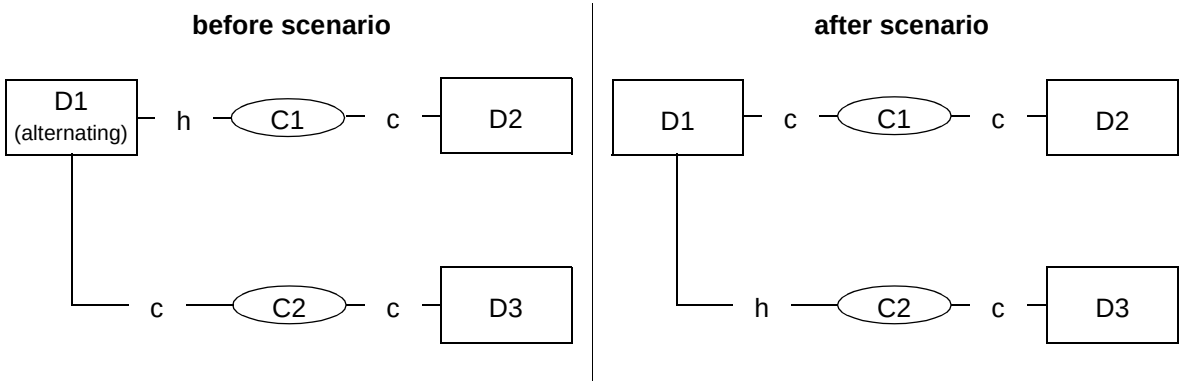
This service places an existing active call on hold and then retrieves a previously held call at the same device. The effect of this service is to swap the device's active and held calls.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
The Alternate Call service is invoked.	AlternateCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	AlternateCallResult			
Connection D1C2 is placed on hold in the active call.	HeldEvent • heldConnection D1C2 • holdingDevice D1 • localConnectionState Held • eventCause alternate		HeldEvent • heldConnection D1C2 • holdingDevice D1 • localConnectionState Connected • eventCause alternate	
Device D1 is connected into the previously held call.	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause alternate	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause alternate		

12.9 Manual alternate call

This scenario illustrates the successful use of the manually alternate feature to place an existing active call on hold and then to retrieve a previously held call at the same device. The effect of this scenario is to swap the devices active and held calls.



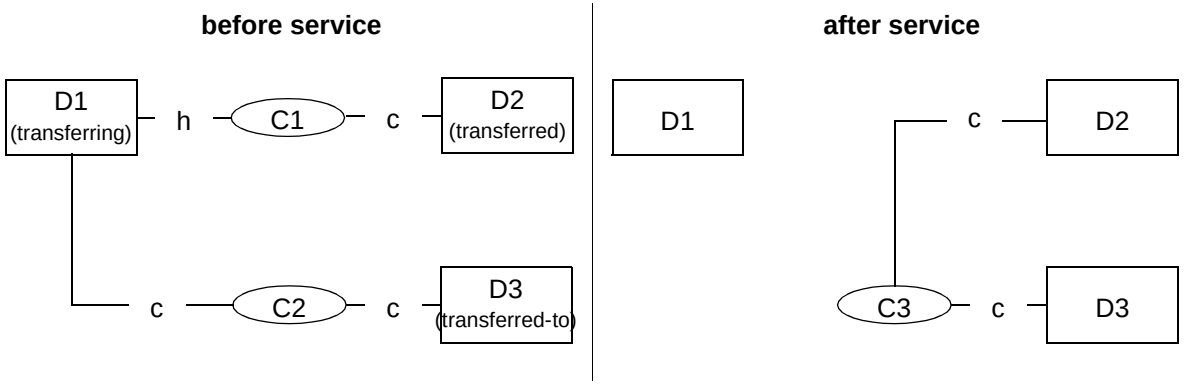
Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Connection D1C2 is placed on hold in the active call.	HeldEvent • heldConnection D1C2 • holdingDevice D1 • localConnectionState Held • eventCause normal		HeldEvent • heldConnection D1C2 • holdingDevice D1 • localConnectionState Connected • eventCause normal	Device D1 wishes to place device D3 on hold and to connect to device D2. It issues the alternate feature manually to do this.
Device D1 is connected into the previously held call.	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause normal	RetrievedEvent • retrievedConnection D1C1 • retrievingDevice D1 • localConnectionState Connected • eventCause normal		

13 Transfer Call Scenarios

This section includes examples of successful call transfers.

13.1 Transfer Call service - screened transfer (with fixed view in Transferred event)

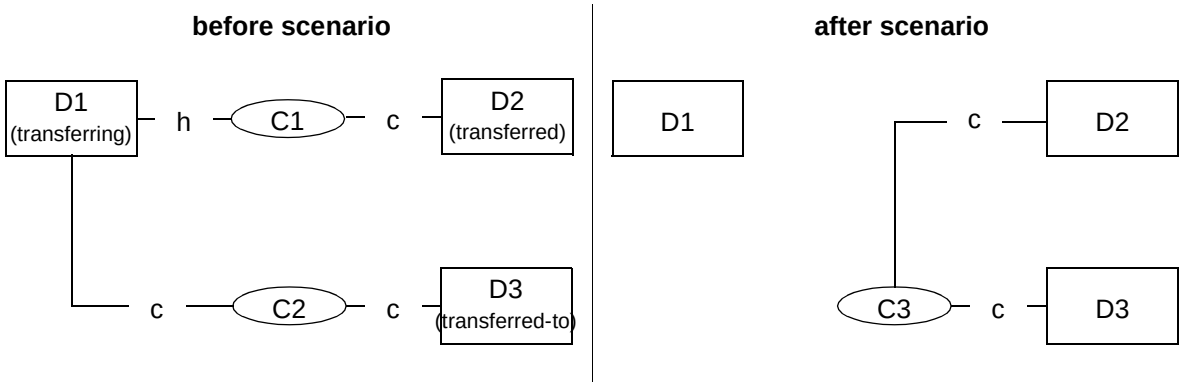
This service transfers a held party to a consulted party.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Transfer Call service is invoked on behalf of device D1.	TransferCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	TransferCallResult • transferredCall D3C3			
Calls between D1,D2 and D1,D3 are released. The connections between D2, D1 and D3,D1 are replaced with a single connection between D2 and D3.	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Null • eventCause transfer	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause transfer	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause transfer	The CSTA Transferred event Fixed View modeling option is illustrated in this scenario. This means that the primary old call parameters in the Transferred event represent a fixed view in contrast to a device oriented view.

13.2 Transfer Call service - screened transfer (with local view in Transferred event)

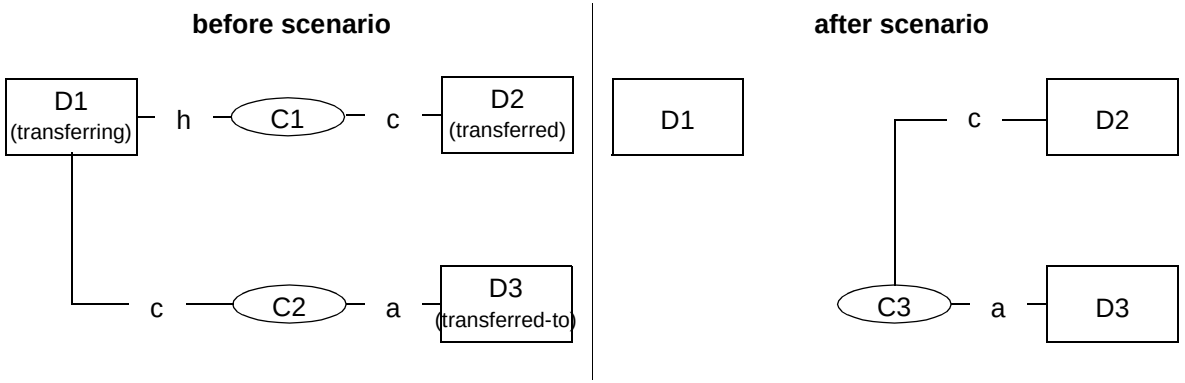
This service transfers a held party to a consulted party.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Transfer Call service is invoked on behalf of device D1.	TransferCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	TransferCallResult • transferredCall D3C3			
Calls between D1,D2 and D1,D3 are released. The connections between D2, D1 and D3,D1 are replaced with a single connection between D2 and D3.	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Null • eventCause Transfer	TransferredEvent • primaryOldCall D2C1 • secondaryOldCall NR • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause Transfer	TransferredEvent • primaryOldCall D3C2 • secondaryOldCall NR • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause Transfer	The CSTA Transferred event Local View modeling option is illustrated in this scenario. This means that the primary old call parameters in the Transferred event represent a device oriented view.

13.3 Transfer Call Call service - blind transfer (with local view in Transferred event)

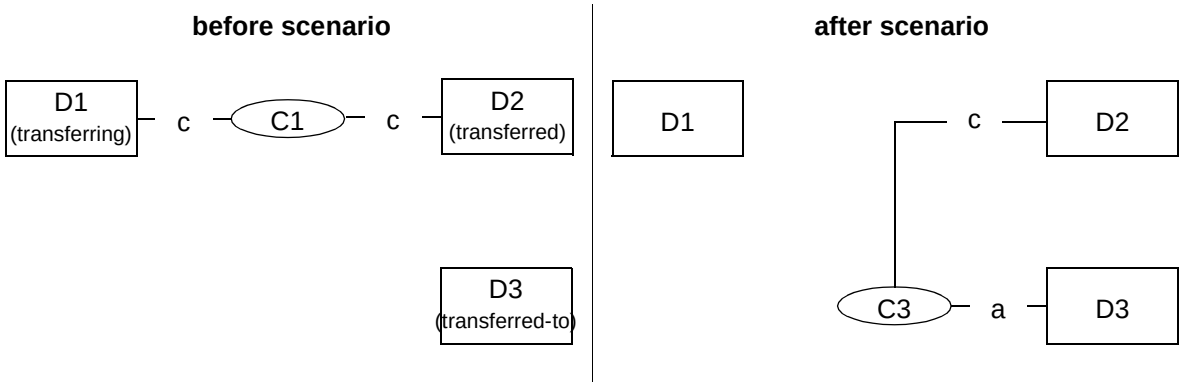
This service transfers a held party to a consulted party. The transfer service request is issued before the consulted device connects into the new call.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Transfer Call service is invoked on behalf of device D1.	TransferCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	TransferCallResult • transferredCall D3C3			
Calls between D1,D2 and D1,D3 are released. The connections between D2, D1 and D3,D1 are replaced with a single alerting connection between D2 and D3.	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Null • eventCause transfer	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause transfer	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Alerting • eventCause transfer	The CSTA Transferred event Fixed View modeling option is illustrated in this scenario. This means that the primary old call parameters in the Transferred event represent a fixed view in contrast to a device oriented view.

13.4 Single Step Transfer Call service

This service transfers a device in one step (i.e., without putting the device on hold).



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Single Step Transfer Call service is invoked on behalf of device D1.	SingleStepTransferRequest • activeCall D1C1 • deviceToTransferTo D3			
Acknowl-edgement.	SingleStepTransferResult • transferredCall D3C3			
The call between D1 and D2 is replaced with an alerting call between D2 and D3.	TransferredEvent • primaryOldCall D1C1 • secondaryOldCall NR • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Null • eventCause singleStep-Transfer	TransferredEvent • primaryOldCall D2C1 • secondaryOldCall NR • transferringDevice D1 • transferredToDevice D3 • transferredConnection D2C3 • transferredConnection D3C3 • localConnectionState Connected • eventCause singleStep-Transfer		The CSTA Transferred event Local View modeling option is illustrated in this scenario. This means that the primary old call parameters in the Transferred event represent a device oriented view. Since there is no connection at D3 at the time of the transfer, there is no Transferred event generated on D3's monitor.

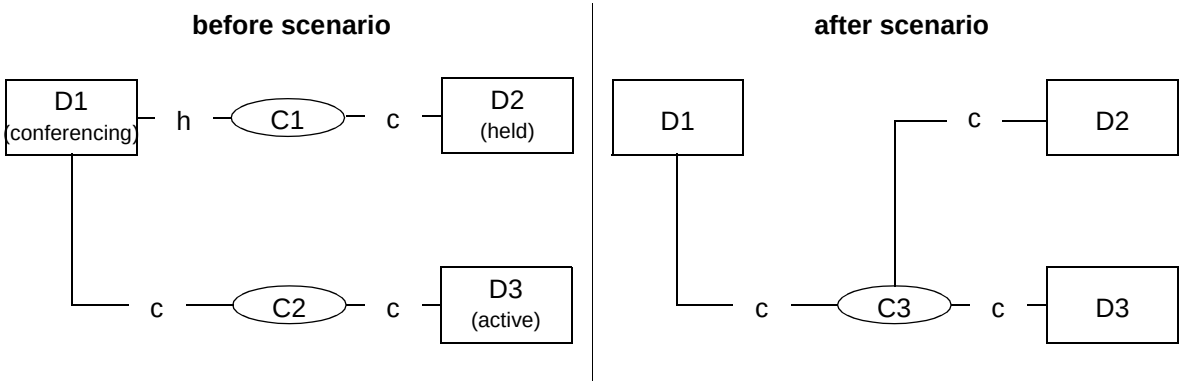
Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
The call alerts device D3.		DeliveredEvent • connection D3C3 • alertingDevice D3 • callingDevice D2 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Connected • cause singleStep-Transfer	DeliveredEvent • connection D3C3 • alertingDevice D3 • callingDevice D2 • calledDevice D3 • lastRedirectionDevice NR • localConnectionState Alerting • cause singleStep-Transfer	This event reflects the connection state change at D3C3.

14 Conference Call Scenarios

This section includes examples of successful conference calls.

14.1 Conference Call service

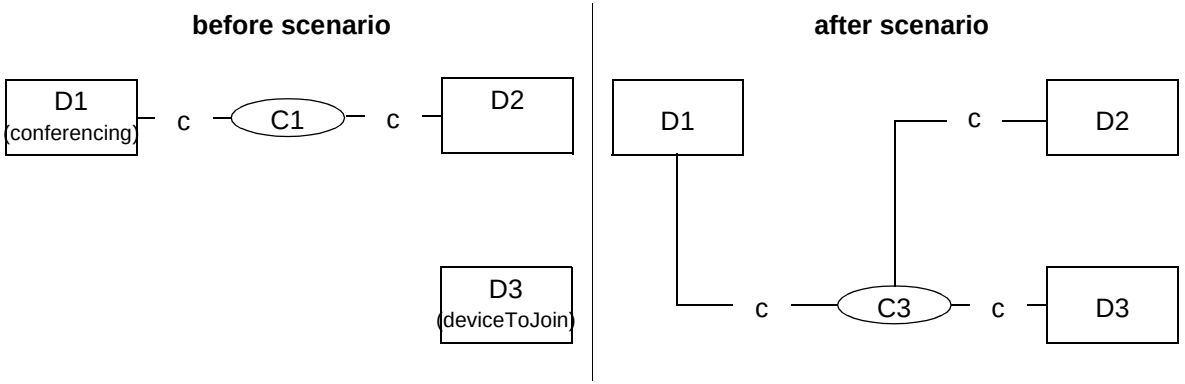
This service provides a conference of an existing held call and another active call at a conferencing device. The two calls are merged into a single call at the conferencing device.



Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Conference Call Service is requested on behalf of device D1.	ConferenceCallRequest • heldCall D1C1 • activeCall D1C2			
Acknowl-edgement.	ConferenceCallResult • conferenceCall D1C3			
Conference established.	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause conference	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause conference	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall D1C2 • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause conference	The added device is the device representing the person who has just joined the call from the perspective of the participants. Note that the primaryOldCall and the secondaryOldCall parameters illustrate the “fixed view” modeling option.

14.2 Single Step Conference Call service

This service provides a conference of an existing call in one step (i.e., without first having to put the existing call on hold). The Single Step Conference Call service is invoked on behalf of device D1 which wishes to silently join the call. The three devices D1, D2, and D3 are then involved in a single call C1. The scenario begins at the point where device D1 begins to join the call. This scenario assumes that device D3 is set to auto-answer the call so device D1 is not shown being prompted to lift the handset.



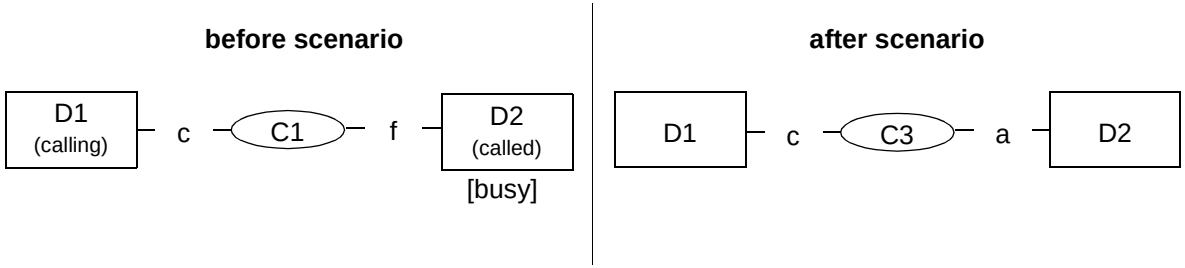
Activity	Monitored Device D1	Monitored Device D2	Monitored Device D3	Comments
Conference Call Service is requested on behalf of device D1.	SingleStepConfCallRequest • activeCall D1C1 • deviceToJoin D3			
Acknowl-edgement.	SingleStepConfCallResult • conferenceCall D1C3			
Conference is established.	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall NR • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause singleStepCon-ference	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall NR • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause singleStepCon-ference	ConferencedEvent • primaryOldCall D1C1 • secondaryOldCall NR • conferencingDevice D1 • addedDevice D3 • conferenceConnection D1C3 • conferenceConnection D2C3 • conferenceConnection D3C3 • localConnectionState Connected • eventCause singleStepCon-ference	The added device is the device representing the person who has just joined the call from the perspective of the participants.

15 Call Completion Scenarios

This clause includes examples of call completion scenarios including call back and camp on.

15.1 Call Back Call-Related service - called device is busy

This scenario illustrates the use of the Call Back Call-Related service where the called device is busy.

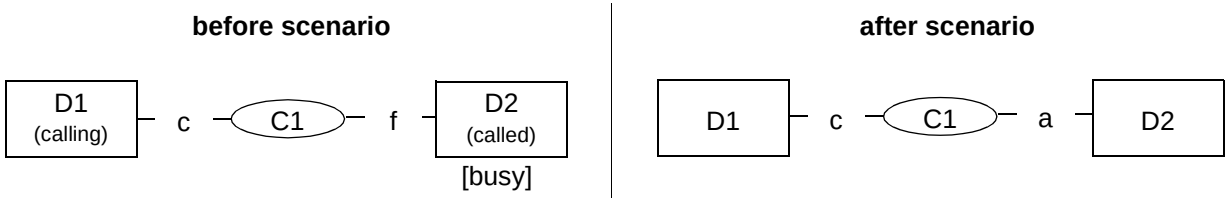


Activity	Monitored Device D1	Monitored Device D2		Comments
The Call Back Call-Related service is invoked on behalf of device D1.	CallBackCallRelatedRequest • callBack D1C1			
Acknowl-edgement.	CallBackCallRelatedResult			
The connection is cleared as a result of the call back request.	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Null • eventCause callback	ConnectionClearedEvent • droppedConnection D1C1 • releasingDevice D1 • localConnectionState Failed • eventCause callback		
Failed connection D2C1 also clears.		ConnectionClearedEvent • droppedConnection D2C1 • releasingDevice D2 • localConnectionState Null • eventCause normalClearing		
Device D2 sometime later clears from its active call.		ConnectionClearedEvent • droppedConnection D2C2 • releasingDevice D2 • localConnectionState Null • eventCause normalClearing		

Activity	Monitored Device D1	Monitored Device D2		Comments
Since device D2 is now available, the callback is initiated from device D1. D1 is being prompted to go off-hook.	ServiceInitiatedEvent • initiatedConnection D1C3 • initiatingDevice D1 • localConnectionState Initiated • eventCause callback			The cause code of callback indicates that the device is being prompted to go off-hook.
Device D1 goes off hook and is connected in the call.	OriginatedEvent • originatedConnection D1C3 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause callback			
Device D2 is alerted.	DeliveredEvent • alertingConnection D2C3 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • eventCause callback	DeliveredEvent • alertingConnection D2C3 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause callback		

15.2 Camp On Call service

This service queues a call for a device (that typically is busy) until that device becomes available.



Activity	Monitored Device D1		Monitored Device D2			Comments
The Camp On Call service is invoked on behalf of device D1.	CampOnCallRequest • campOn	 D1C1				
Acknowl-edgement.	CampOnCallResult					
The call is queued at D2.	QueuedEvent • queuedConnection • queueDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D2C1 D2 D1 D2 NR Connected campOn	QueuedEvent • queuedConnection • queueDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D2C1 D2 D1 D2 NR Queued campOn		Queued events may be provided instead of a Delivered event (for example for a second caller).
Device D2 sometime later clears from its active call.			ConnectionClearedEvent • droppedConnection • releasingDevice • localConnectionState • eventCause	 D2C2 D2 Null normalClearing		
Since D2 is available, the call alerts D2.	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D2C1 D2 D1 D2 NR Connected campOn	DeliveredEvent • alertingConnection • alertingDevice • callingDevice • calledDevice • lastRedirectionDevice • localConnectionState • eventCause	 D2C1 D2 D1 D2 NR Alerting campOn		

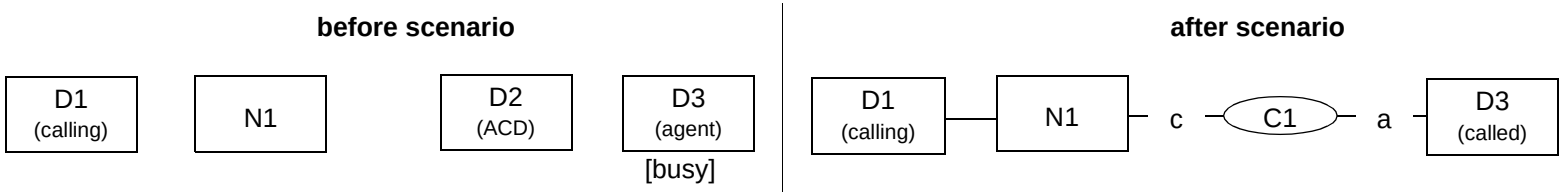
16 Distributing Call Scenarios

This clause illustrates calls that are distributed by a CSTA distribution device (ACD) to an available device.

16.1 Incoming Call to ACD with no available agents

This scenario illustrates an incoming call to an ACD distribution mechanism (ACD). Since there is no available device (ACD agent) the call is queued at the distribution device until an agent becomes available.

The calling device D1 is located outside the CSTA sub-domain and therefore no events are shown for this device. Device N1 in this scenario represents the Network Interface Device (trunk) associated with the calling device.



Activity	Monitored Device N1 (trunk)	Monitored Device D2 (ACD)	Monitored Device D3 (agent).	Comments
Indicates an external incoming call on the NID (e.g. trunk).	ServiceInitiatedEvent • initiatedConnection N1C1 • initiatingDevice N1 • localConnectionState Initiated • eventCause newCall • assoc.CallingDevice N1			
The NID has connected in the call.	OriginatedEvent • originatedConnection N1C1 • callingDevice D1 • calledDevice D2 • localConnectionState Connected • cause newCall • assoc.CallingDevice N1			

Activity	**Monitored Device N1 (trunk)**	**Monitored Device D2 (ACD)**	**Monitored Device D3 (agent).**	**Comments**
The call arrives at a distribution device (ACD).	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • eventCause enteringDistri-bution • assoc.CallingDevice N1	DeliveredEvent • alertingConnection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause enteringDistri-bution • assoc.CallingDevice N1		
There are no available devices (agents) associated with the distribution device (D2) - the call will be queued.	QueuedEvent • queuedConnection D2C1 • queue D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • eventCause noAvailable-Agents • networkCallingDevice D1 • networkCalledDevice D2 • assoc.CallingDevice N1	QueuedEvent • queuedConnection D2C1 • queue D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Queued • eventCause noAvailable-Agents • networkCallingDevice D1 • networkCalledDevice D2 • assoc.CallingDevice N1		Device D2 may identify an ACD group or the ACD queueing mechanism. A call may also be queued for multiple devices (not shown).
Agent D3 becomes available.			AgentReadyEvent • agentDevice D3	
The call leaves the ACD Distri-bution device.	DivertedEvent • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Connected • eventCause distributed • networkCallingDevice D1 • networkCalledDevice D2 • assoc.CallingDevice N1	DivertedEvent • divertingConnection D2C1 • divertingDevice D2 • newDestination D3 • lastRedirectionDevice NR • localConnectionState Null • eventCause distributed • networkCallingDevice D1 • networkCalledDevice D2 • assoc.CallingDevice N1		This illustrates the CSTA modeling option (as specified via the capability exchange services) where the Diverted event is being sent to all devices in the call, not just for the diverting device (D2) monitor. An announcement can be provided prior to this event for which events may be generated (not shown).

Activity	**Monitored Device N1 (trunk)**	**Monitored Device D2 (ACD)**	**Monitored Device D3 (agent).**	**Comments**
The call is delivered to the available agent device (D3).	DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • localConnectionState Connected • eventCause distributed • assoc.CallingDevice N1		DeliveredEvent • alertingConnection D3C1 • alertingDevice D3 • callingDevice D1 • calledDevice D3 • lastRedirectionDevice D2 • networkCallingDevice D1 • networkCalledDevice D2 • localConnectionState Alerting • eventCause distributed • assoc.CallingDevice N1	

Free printed copies can be ordered from:
ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

Fax: +41 22 849.60.01
Internet: documents@ecma.ch

Files of this Technical Report can be freely downloaded from the ECMA web site (www.ecma.ch). This site gives full information on ECMA, ECMA activities, ECMA Standards and Technical Reports.

ECMA
114 Rue du Rhône
CH-1204 Geneva
Switzerland

See inside cover page for obtaining further soft or hard copies.