#### ECMA Technical Report TR/68

December 1994



Standardizing Information and Communication Systems

#### Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase II

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/68

December 1994



Standardizing Information and Communication Systems

#### Scenarios for Computer Supported Telecommunications Applications (CSTA) Phase II

#### **Brief History**

This Technical Report is based upon Phase II of Services for Computer Supported Telecommunications Applications (CSTA) and introduces CSTA Scenarios. This Technical Report uses Standards ECMA-217 Services for Computer Supported Telecommunications Applications Phase II and ECMA-218 Protocol for Computer Supported Telecommunications Applications Phase II to illustrate how CSTA services and events may be used in typical call scenarios. It reflects a common understanding of ECMA member companies. Additional phases of this Technical Report are anticipated. This Technical Report is based on the practical experience of ECMA member companies and represents a pragmatic and widely-based consensus.

The Phase II CSTA standards are not fully backwards compatible with the Phase I standards. Although backwards compatibility is an important consideration and has been maintained whenever possible, the addition of new parameters in certain services and events, as well as the deletion of unused Phase I services and the addition of entirely new Phase II services and events, did not allow complete backwards compatibility.

This Technical Report is dedicated to the memory of Terry Wuerfel

#### Table of contents

	Page
1 Scope	1
2 References	1
3 Make Call Scenarios	3
3.1 Successful Make Call	3
3.2 Successful manual Call (Off-hook dialling)	5
3.3 Make Call meets Called Party Busy	6
3.4 Make Call meets Calling Party Busy	8
3.5 Make Call meets no answer	9
3.6 Make Call to an invalid number	11
3.6.1 Make Call to an invalid number - User already off-hook	12
3.7 Make Call encounters Privilege Violation on Calling Device	13
3.7.1 Make Call encounters Privilege Violation on Calling Device - User off-hook	14
3.7.2 Make Call encounters Privilege Violation on Called Device	15
3.7.3 Make Call encounters Privilege Violation on Called Device - User off-hook	16
3.8 Offnode Make Call (Networked reached)	17
3.9 Make Call while listening to Dial tone	18
3.10 Make Call Hands-free operation	19
3.11 Successful manual Offnode Call ("Overlap" off-hook dialling)	20
4 Successful ISDN ("En-bloc") Call	21
5 Call Release Scenarios	23
5.1 Call Release	23
5.2 The CSTA Blocked Call State	24
6 Consultation Calls	25
6.1 Successful Consultation Call	25
6.2 Consultation Call Rejected	27
6.3 Consulted party busy	28
6.4 Consulted party clears - consulting party receives dial tone	30
6.5 Held party clears	31
7 Successful Alternate Call	32
8 Transfers	33
8.1 Successful Supervised Transfer	33
8.2 Successful Unsupervised Transfer	34
8.3 Single Step Transfer	35

39
39
40
42
43
44
46
47
47
49
51
53
53
55

#### 1 Scope

Services for Computer Supported Telecommunications Applications are defined by Standard ECMA-217 and the Protocol for those services by Standard ECMA-218. Our intention here is to illustrate how, for a range of call scenarios, the services offered by ECMA CSTA are intended to be used by application developers and switch manufacturers.

Each scenario includes a textual description and an illustration. Illustrations use the same key as described within ECMA-217 section 9.3. For each scenario, message sequences are listed for all monitored devices - call type monitors have not been illustrated. All devices have device type monitors set with no events masked. The *Activity* column includes a brief description of the event or service invocation, the *Monitored Device* column lists events and service invocations with the associated parameter list. Optional parameters within the *Monitored Device* column are shown in italics. 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*.

CSTA provides an event reporting service which contains three parameters. These parameters, crossRefIdentifier, eventSpecificInfo and extensions are fully defined by ECMA-218. Within these scenarios only the content of the eventSpecificInfo parameter within the CSTA Event Reporting Service is shown.

These scenarios show a strict order of messaging (i.e. a service request is always followed by a service acknowledgement). However, in a true implementation an acknowledgement to a service request may be received in an asynchronous mode by the computing function. Events are generated throughout the life of a call and as such represent in these scenarios the passing of time.

The precise moment in relation to the switching function activity at which a response is generated, shall be implementation- and Service- dependent. For example, for the Hold Call Service some implementations may generate the Service Response after validating the correctness of the request and when they initiate the requested Service. Other implementations may delay the response until the Hold has completed (or is guaranteed to complete). In this case, a failure of the requested switching Service is reflected in the Service Response.

The scenarios are only for information and as such ECMA-217 (Services) and ECMA-218 (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. A method of providing these scenarios as a collection of smaller building blocks which may then be connected together to form more complex scenarios is under study.

SubjectDeviceID is used in some event reports to specify which device the report refers to. If the SubjectDeviceID has had a monitor invoked upon it then this data is not required and so the implicit NULL encoding for notRequired should be returned, as defined by ECMA-218. In these scenarios, as all devices have active device monitors, the SubjectDeviceID is shown as /NR to indicate Not Required. NK is used to indicate notKnown where appropriate. Within these Scenarios, () indicates that the content is not specified by these scenarios and that anything defined within the protocol is valid.

#### 2 References

ECMA TR/52	Computer Supported Telecommunications Applications (CSTA) (1990)
ECMA-217	Services for Computer Supported Telecommunications Applications (CSTA) Phase II (1994)
ECMA-218	Protocol for Computer Supported Telecommunications Applications (CSTA) Phase II (1994)

#### Make Call Scenarios 3

call are illustrated. A Make Call Response, positive or negative, may indicate the result of the Make Call Request. For some switch implementations the result of the request is determined from the CSTA events which follow the Make Call Response. answered. There are four examples of calls rejected because of privilege violations. Calls made to devices outside of the CSTA domain and an ISDN en-bloc This section includes examples of successful Make Calls, initiated manually and by applications, Make Calls by and to busy parties and calls which do not get

#### Successful Make Call 3.1

tted to make the call and the				Comments									The generation of this event	is switch-specific.																		
This scenario initiates a call from device D1 to device D2. In this scenario both devices are free and valid, device D1 is permitted to make the call and the call reaches establishment.				MONITORED DEVICE D3																												
oth devices are																								D2C1	D2/NR	D1	D2	NR S	<b>C</b>	Alerting		newcall
ce D2. In this scenario b				MONITORED DEVICE D2																			DeliveredEvent	• connection	• alerting Device	• calling Device	• calledDevice	• lastRedirectionDevice	<ul> <li>originatingConnection</li> </ul>	• localConnectionInfo	• correlatorData	• cause
ice D1 to devi	AFTER		ノコ		D1	D2	20	$\sim$	<b>-</b>		DICI			DICI	Initiated makeCall		DICI	D1/NR	D2		Connected	newCall		D2C1	D2	D1	D2	NR S		Connected	( )	newcau
This scenario initiates a call from dev call reaches establishment.		[2	77	MONITORED DEVICE D1	MakeCallRequest • callingDevice	• calledDirectoryNumber	• accountCode	• authCode	• correlatorData • extensions	MoboCollDomlt	MakeCalinesum   • initiatedCall	• extensions	ServiceInitiatedEvent	• initiatedConnection	localConnectionInfo     cause	OriginatedEvent	originatedConnection	• callingDevice	• calledDevice	• originatingDevice	• tocat Connectioningo	• cause	DeliveredEvent	• connection	alertingDevice	callingDevice	• calledDevice	lastRedirectionDevice	• originating Connection	• localConnectionInfo	• correlatorData	• cause
This scenaric call reaches e	BEFORE		2	Activity	A MakeCall to a valid device is invoked on	behalf of device D1.				A ofm one lad a sust	Acknowleagement.		Indication that service has	been initiated from this	device.	Device DI lifts handset	ready for call.	•					Device D2 begins to ring	and DI listens to ringing	tone.							

Device D2 answers the EstablishedEvent	EstablishedEvent		EstablishedEvent		Device D2 manually
call.	establishedConnection	D2C1	<ul> <li>establishedConnection</li> </ul>	D2C1	answers the call by lifting
	answeringDevice	D2	<ul> <li>answeringDevice</li> </ul>	D2/NR	handset.
	callingDevice	D1	<ul> <li>callingDevice</li> </ul>	DI	
	calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2	
	lastRedirectionDevice	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	originatingConnection	$\sim$	<ul> <li>originatingConnection</li> </ul>		
	localConnectionInfo	Connected	ullet local Connection Info	Connected	
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\overline{}$	
	• cause	newCall	• cause	newCall	

### Successful manual Call (Off-hook dialling) 3.2

In this scenario the instrument associated with device D1 is manually lifted to initiate a call from device D1 to device D2. Both devices are initially free and valid, device D1 is permitted to make the call and the call reaches establishment.

AFTER D1 D2 BEFORE D1

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Device DI goes off-hook	ServiceInitiatedEvent					Device is authorised to make
and receives dial tone.	initiatedConnection	D1C1				calls. This event is always
	• localConnectionInfo	Initiated				generated for a manual
	• cause	newCall				Make Call.
Dialling completed - call	OriginatedEvent	, i				
set-up taking place.	originatedConnection	DICI				
	• callingDevice	D1/NR				
	calledDevice	D2				
	• originating Device	$\sim$				
	• localConnectionInfo	Connected				
	• correlatorData	$\mathcal{C}$				
	• cause	newCall	:			;
Device D2 is free and	DeliveredEvent		DeliveredEvent			Assuming that no diverts are
begins to ring.	connection	D2C1	<ul> <li>connection</li> </ul>	D2C1		in operation.
	alertingDevice	D2	<ul> <li>alertingDevice</li> </ul>	D2/NR		
	callingDevice	D1	<ul> <li>callingDevice</li> </ul>	DI		
	calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2		
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
	<ul> <li>originating Connection</li> </ul>	$\sim$	<ul> <li>originating Connection</li> </ul>	$\sim$		
	• localConnectionInfo	Connected	ullet local Connection Info	Alerting		
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	newCall	• cause	newCall		
AnswerCall service is			AnswerCallRequest			An error will be sent if the
invoked on behalf of			<ul> <li>callToBeAnswered</li> </ul>	D2C1		device is not able to answer
device D2.			• extensions	$\Box$		the call without manual
						intervention.
Acknowledgement.			AnswerCallResult	(		
,	;		• extensions	()		
Device D2 answers the	EstablishedEvent	5	EstablishedEvent			
call.	• establishedConnection	D2C1	• established Connection	DZCI		
	<ul> <li>answeringDevice</li> </ul>	D2	<ul> <li>answeringDevice</li> </ul>	D2/NR		
	callingDevice	DI	<ul> <li>callingDevice</li> </ul>	DI		
	calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2		
	lastRedirectionDevice	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
	originatingConnection	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\sim$		
	• localConnectionInfo	Connected	ullet local Connection Info	Connected		
	<ul> <li>correlatorData</li> </ul>	$\sim$	ullet correlator $Data$	$\sim$		
	• cause	newCall	• cause	newCall		

### Make Call meets Called Party Busy 3.3

ot St																																					
s not invoked a divert. That device D2 is busy is no			Comments	Comments									<u> </u>	The generation of this event	is switch-specific.																						
The application initiates a call from device D1 to device D2. Device D1 is prompted to lift the handset. Device D2 is busy and has not invoked a divert. The call fails because the called party is busy. Device D1 clears without invoking a supplementary service. In this scenario the fact that device D2 is busy is not indicated in the MakeCallResult but by a subsequence of event reports.			MONITORED DEVICE D3	MONITONED DEVICE DS																																	
ompted to lift tl ; a supplementa																										וייינים	D2/NR	D2.	Failed	$\sim$	Busy		DICI	D1	r antea	normal	Clearing
vice D2. Device D1 is pr clears without invoking e of event reports.		$\begin{array}{c} 1 \\ 1 \end{array} $ f- $\begin{array}{c} 1 \\ 1 \end{array}$	MONITORED DEVICE D2	MONITONED DEVICE DZ																						FalledEvent	• failingDevice	• called Device	• localConnectionInfo	• correlatorData	• cause	ConnectionClearedEvent	<ul> <li>droppedConnection</li> </ul>	• releasing Device	• correlatorData	esting constant	
evice D1 to dev asy. Device D1 y a subsequenc	AFTER	D1 c(C1	)		D1	D2	23			Ö		D1C1	()		D1C1	Initiated	makeCall	,	DICI	D1/NR	D2		Connected	()	пемсан	וייירו	D2	D2	Connected		Busy	·	DICI	D1/NR	lvutt	normal	Clearing
The application initiates a call from device D1 to device D2. Device D call fails because the called party is busy. Device D1 clears without in indicated in the MakeCallResult but by a subsequence of event reports.		D2	MONITORED DEVICE DI	MONITONED DEVICE DI	MakeCallRequest • callingDevice	• calledDirectoryNumber	• devicer rojue • ggggggtfoda	• authCode	• correlatorData	<ul> <li>extensions</li> </ul>	MakeCallResult	• initiatedCall	<ul> <li>extensions</li> </ul>	ServiceInitiatedEvent	<ul> <li>initiatedConnection</li> </ul>	ullet local Connection Info	• cause	OriginatedEvent	<ul> <li>originatedConnection</li> </ul>	• callingDevice	<ul> <li>calledDevice</li> </ul>	<ul> <li>originatingDevice</li> </ul>	<ul> <li>localConnectionInfo</li> </ul>	• correlatorData	• cause	Falled Event  • failed Connection	failing Device	• calledDevice	• localConnectionInfo	• correlatorData	• cause	ConnectionClearedEvent	<ul> <li>droppedConnection</li> </ul>	• releasingDevice	• correlator Data	· Canse	
The applicati call fails beco indicated in tl	BEFORE	DI	Activity	Acuvity	MakeCall service is invoked on behalf of	device DI.					Acknowledgement.			Device DI notified of	initiating call.			Device DI lifts handset	and establishes the calling	connection of the call.						Device D2 is busy - the	Device DI hears husy	tone				Device DI now replaces	handset.				

Failed connection D2CI	ConnectionClearedEvent		Failed connections must also
also clears.	droppedConnection	D2C1	clear and therefore report
	<ul> <li>releasingDevice</li> </ul>	D2	clear connection events.
	localConnectionInfo	Null	
	• correlatorData	$\overline{}$	
	• cause	normal	
		Clearing	

# 3.4 Make Call meets Calling Party Busy

CSTA Make Call fails because the calling party, device D1, is busy on a call to device D2 when a MakeCallRequest is issued on behalf of device D1. On certain types of device the switch may be able to launch the call on a separate line appearance. This operation is implementation-dependent and is therefore not illustrated here.

D3

D3

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
User initiates Make Call.   MakeCallRequest	MakeCallRequest				
	callingDevice	D1			
	called Directory Number	D3			
	• deviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Make Call fails because	MakeCallError				
calling party is busy.	• stateErrors	invalid object			
		state			

### Make Call meets no answer 3.5

The application initiates a call from device D1 to device D2 and prompts the user of device D1 to lift the handset. Device D1 and device D2 are free but

The generation of this event is switch-specific. Comments MONITORED DEVICE D3 newCall () Alerting Alerting D2C1 D2/NR D1 D2 NR D1C1 device D2 doesn't answer the call. Device D1 replaces the handset to clear the call. MONITORED DEVICE D2 originating Connection ConnectionClearedEvent lastRedirectionDevice localConnectionInfo localConnectionInfo droppedConnection releasingDevice correlatorData correlatorData alertingDevice callingDevice D2 DeliveredEvent • calledDevice () Connected Connected makeCall Initiated newCall newCall D1C1 D1/NR D2 D1/NR AFTER D1C1 D2C1 D2 D1 D2 NR D1C1 D1C1 Null D D1 D2 2222 MONITORED DEVICE D1 • originating Connection • local ConnectionInfo calledDirectoryNumber ConnectionClearedEvent lastRedirectionDevice localConnectionInfo originatedConnection localConnectionInfo localConnectionInfo ServiceInitiatedEvent initiatedConnection droppedConnection ullet originating DeviceMakeCallRequest releasingDevice · correlatorData correlatorData correlatorData correlatorData D2 OriginatedEvent MakeCallResult alertingDevice callingDevice DeliveredEvent callingDevice deviceProfile • accountCode callingDevice calledDevice • calledDevice initiatedCall extensions extensions connection authCode • cause • cause Device DI lifts handset to User initiates Make Call. BEFORE Device D2 is free and begins to alert. Device DI notified of initiating call. originate calling end because of failure to D1 replaces handset  $\overline{D}$ Acknowledgement. connection. connect. Activity

Clearing

Clearing

normal

normal

Connection D2C1 clears	ConnectionClearedEvent		The cause value
as a result of D1 clearing.	<ul> <li>droppedConnection</li> </ul>	D2C1	"callNotAnswered" only
	<ul> <li>releasingDevice</li> </ul>	D2/NR	applies if a call is cleared by
	ullet local Connection Info	Null	a switch as a result of a
	• correlatorData		timer rather than by the
	• cause	normal	caller replacing their
		Clearing	handset.

### Make Call to an invalid number 3.6

This scenarion invalid numb	This scenario initiates a call from device D1 to device D2. In this scenari invalid number (not part of any known numbering plan) and the call fails.	ce D1 to dev numbering	device D2. In this scenario device D1 is free, valid and permitted to make the call. Device D2 is actually an ing plan) and the call fails.	ee, valid and permitted to make the call	. Device D2 is actually an
BEFORE	7	AFTER			
D1		DI			
	_				
Activity	MONITORED DEVICE DI		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
MakeCall service to an	MakeCallRequest				
invalid number is initiated • callingDevice	• callingDevice	D1			
on behalf of device D1.	calledDirectoryNumber	D2			
	• deviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Acknowledgement.	UniversalFailure				
	<ul> <li>operationalErrors</li> </ul>	invalid			
		Called			
		Device			

# 3.6.1 Make Call to an invalid number - User already off-hook

This scenario invokes a call that was already initiated by a user going off-hook on a telephone. The call is from device D1 to D2. In this scenario device D1 is free, valid and permitted to make the call. Device D2 is actually an invalid number (not part of any known numbering plan) and the call fails.

BEFORE AFTER  $D1 \longrightarrow i \leftarrow C1$   $D1 \longrightarrow D1$ 

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
Device D1 manually	ServiceInitiatedEvent				
initiates a call by going	initiatedConnection	D1C1			
off-hook.	<ul> <li>localConnectionInfo</li> </ul>	Initiated			
	• cause	newCall			
MakeCall service to an	MakeCallRequest				
invalid device invoked on	<ul> <li>callingDevice</li> </ul>	D1			
behalf of device D1.	<ul> <li>calledDirectoryNumber</li> </ul>	D2			
,	• deviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Acknowledgement.	UniversalFailure				D2 is a number that is
	operational Errors	invalid			unknown to the switching
	•	Called			function.
		Device			
Device DI gets a "failed"	FailedEvent				D2 is a number that is
tone.	failedConnection	D1C1			unknown to the switching
	failing Device	D1/NR			function.
	• calledDevice	D2			)
	<ul> <li>localConnectionInfo</li> </ul>	Failed			
	• correlatorData	$\sim$			
	• cause	destNot			
		Obtainable			
Connection cleared as DI	ConnectionClearedEvent				
replaces handset.	<ul> <li>droppedConnection</li> </ul>	D1C1			
	releasingDevice	D1/NR			
	• localConnectionInfo	Null			
	• correlatorData	$\sim$			
	• cause	normal			
		Clearing			

## Make Call encounters Privilege Violation on Calling Device 3.7

This scenari call fails.	o initiates a call from de	vice D1 to dev	ice D2. In this case device D1 is not p	This scenario initiates a call from device D1 to device D2. In this case device D1 is not permitted to attempt calls resulting in a privilege violation and the call fails.	orivilege violation and the
BEFORE		AFTER			
DI	D2	DI	D2		
Activity	MONITORED DEVICE D1	]	MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
MakeCall service to	MakeCallRequest				
device D2 invoked on	callingDevice	D1			
behalf of device D1.	calledDirectoryNumber	D2			
	• deviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Acknowledgement.	UniversalFailure				
	operationalErrors	Privilege			
		ViolationOn			
		Calling			
		Device			

# Make Call encounters Privilege Violation on Calling Device - User off-hook 3.7.1

This scen device D1 may not b	This scenario dials a call that was already initiated by a user going device D1 is not permitted to make calls - so calling device D2 cause may not be a privilege violation as it is also a way to activate features.	already initiate calls - so call t is also a way	ed by a user going off-hook on a telling device D2 causes a privilege vio to activate features.	This scenario dials a call that was already initiated by a user going off-hook on a telephone. The call is from device D1 to device D2. In this scenario device D1 is not permitted to make calls - so calling device D2 causes a privilege violation, at device D1, and the call fails. Note that off-hook in itself may not be a privilege violation as it is also a way to activate features.	evice D2. In this scenario lote that off-hook in itself
BEFORE		AFTER			
I I	-(C1) D2	DI	D2		
Activity	MONITORED DEVICE D1		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
Device DI manually	ServiceInitiatedEvent				
initiates a call by going	• initiatedConnection	D1C1 Initiated			
oy-nook.	• cause	newCall			
MakeCall service to an	MakeCallRequest				
invalid device is invoked	• callingDevice	D1			
on behalf of device DI.	calledDirectoryNumber	D2			
	• deviceProfile				
	• accountCode				
	• authCode				
	• correlatorData				
	• extensions	()			
Acknowledgement.	UniversalFailure	:			Outbound calling is not
	operationalErrors	privilege			allowed from this device.
		ViolationOn			
		Calling Device			
Device DI gets a "failed"	FailedEvent				D2 is a number that is not
tone.	<ul> <li>failedConnection</li> </ul>	D1C1			reachable from device D1.
	• failingDevice	D1/NR			
	• caned Device	D2 Eailed			
	• correlatorData	Lanea			
	• content and	dostNot			
	ecuns.	Obtainable			
Connection cleared as DI	ConnectionClearedEvent				Failed connections must also
replaces handset.	droppedConnection	D1C1			be cleared.
	releasingDevice	D1/NR			
	• localConnectionInfo	Null			
	correlatorData	$\sim$			
	• cause	normal			
		Clearing			

## Make Call encounters Privilege Violation on Called Device 3.7.2

This scenario initiates a call from device D1 to device D2. In this

This scen D2, D1 a	This scenario initiates a call from device D1 to device D2. In D2, D1 actually causes a privilege violation and the call fails.	device D1 to c	This scenario initiates a call from device D1 to device D2. In this scenario device D1 is free, valid and permitted to attempt calls - but by calling device D2, D1 actually causes a privilege violation and the call fails.	free, valid and permitted to attempt ca	ills - but by calling device
BEFORE		AFTER			
DI	D2	DI	D2		
Activity	MONITORED DEVICE DI		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
Device DI initiates a MakeCall to an invalid	MakeCallRequest • callingDevice	D1			
device.	• calledDirectoryNumber	D2			
	• deviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	correlatorData	$\Box$			
	• extensions	()			
Acknowledgement.	UniversalFailure				Calling D2 is not allowed for
	operational Errors	privilege			this device.
		ViolationOn			
		Called			
		Device			

# 3.7.3 Make Call encounters Privilege Violation on Called Device - User off-hook

This scenario dials a call that was already initiated by a user going off-hook on a telephone. The call is from device D1 to device D2. In this scenario device D1 is free, valid and permitted to make calls - but by calling device D2, D1 causes a privilege violation and the call fails.

Calling D2 is not allowed for this device. D2 is a number that is not reachable from device DI. Comments MONITORED DEVICE D3 MONITORED DEVICE D2 D2 privilege ViolationOn Called Device ( ) destNot Obtainable AFTER D1C1 Initiated newCall ClearingD1C1 D1/NR Null () normal D1C1 D1/NR D2 Failed D D1 D2 MONITORED DEVICE DI calledDirectoryNumber ConnectionClearedEvent • initiatedConnection • localConnectionInfo localConnectionInfo localConnectionInfo ServiceInitiatedEvent droppedConnection D2 operationalErrors failedConnection MakeCallRequest releasingDevice correlatorData correlatorData correlatorData UniversalFailure callingDevice • deviceProfile accountCode failingDevice calledDevice extensions authCode FailedEvent • cause BEFORE Connection cleared as DI replaces handset. Device DI gets a "failed" tone. invalid device is invoked initiates a call by going MakeCall service to an on behalf of device D1.  $\Box$ Device DI manually Acknowledgement. off-hook.

# 3.8 Offnode Make Call (Networked reached)

A CSTA make call is invoked and the call is routed out of the CSTA sub-domain. Device D1 is prompted to lift the handset. D3, which lies outside the CSTA domain, can not be monitored and therefore no events will be seen for that device. Event information after the NetworkReached event is not available.

D3 CSTA Domain AFTER DI D2 BEFORE D1

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
User initiates a call.	MakeCallRequest					
	callingDevice	D1				
	<ul> <li>calledDirectoryNumber</li> </ul>	D3				
	• deviceProfile	$\sim$				
	• accountCode	$\Box$				
	• authCode	$\sim$				
	• correlatorData	$\Box$				
	• extensions	()				
Acknowledgement.	MakeCallResult					
,	initiatedCall	D1C1				
	• extensions	$\Box$				
Device DI notified of	ServiceInitiatedEvent					The generation of this event
initiating call.	<ul> <li>initiatedConnection</li> </ul>	D1C1				is switch-specific.
	• localConnectionInfo	Initiated				
	• cause	makeCall				
Device DI lifts handset	OriginatedEvent					
ready for call.	<ul> <li>originatedConnection</li> </ul>	D1C1				
	<ul> <li>callingDevice</li> </ul>	D1/NR				
	<ul> <li>calledDevice</li> </ul>	D3				
	<ul> <li>originatingDevice</li> </ul>	D2				
	<ul> <li>localConnectionInfo</li> </ul>	Connected				
	• correlatorData	$\supset$				
	• cause	newCall				
The call leaves the CSTA	NetworkReachedEvent		NetworkReachedEvent			The Called Device
domain.	outboundConnection	D2C1	outboundConnection	D2C1		information will be
	• trunkUsed	D2	• trunkUsed	D2/NR		complete. Within a
	• calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3		networking environment the
	<ul> <li>localConnectionInfo</li> </ul>	Connected	<ul> <li>localConnectionInfo</li> </ul>	Connected		trunk may not be visible.
	• correlatorData	$\Box$	• correlatorData	$\sim$		
	• cause	newCall	• cause	newCall		

### Make Call while listening to Dial tone 3.9

This scenario originates a call from device D1 to device D2. In this scenario device D1 is free, valid and permitted to make the call - and device D2 is a

Call now proceeds. Comments MONITORED DEVICE D3 valid number. The switching function dials the call for which the user of device D1 has already gone off-hook. MONITORED DEVICE D2 D2 DIC1 D1/NR D2 ( ) Connected D1C1 Initiated newCall newCall AFTER D1C1 DI ...sequence proceeds as other Make Call scenarios... MONITORED DEVICE D1 calledDirectoryNumber originatedConnection • initiatedConnection • localConnectionInfo originatingDevice
 localConnectionInfo ServiceInitiatedEvent MakeCallRequest correlatorData correlatorData MakeCallResult
• initiatedCall OriginatedEvent D2 callingDevice callingDevice ullet device Profile accountCode • calledDevice extensions • authCode extensions • cause Call proceeds from device DI. Device DI goes off-hook. BEFORE device D2 invoked on MakeCall service to behalf of device D1. Acknowledgement. D1 Activity

### Make Call Hands-free operation 3.10

This scenario originates a call from device D1 to device D2. In this scenario device D1 is free, valid and permitted to make the call - and device D2 is a

"digital" sets to go off-hook without the user having to participate. With "analogue" sets it is necessary for the switching function to prompt the user to lift the handset.			MONITORED DEVICE D3 Comments									Call now proceeds.										
valid number. The switching function dials the call and automatically forces device D1 to go off-hook. Switching functions often have the capability to force "digital" sets to go off-hook without the user having to participate. With "analogue" sets it is necessary for the switching function to prompt the user to lift the handset.	AFTER	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	MONITORED DEVICE D1 MONITORED DEVICE D2	MakeCallRequest	• callingDevice D1	calledDirectoryNumber D2	• deviceProfile	• accountCode ()	• authCode ( )	• correlatorData ()	• extensions ()	MakeCallResult	• initiatedCall D1C1	• extensions ()	nection	•	• calledDevice D2	• originating Device ()	• localConnectionInfo Connected	• correlatorData ()	• cause newCall	scenario proceeds as for other Make Call scenarios
valid number. "digital" sets t	BEFORE		Activity	MakeCall to device D2	invoked on behalf of	device DI.						Acknowledgement.			D1. The switching function	automatically forced DI	off-hook without the user	participating.				

# 3.11 Successful manual Offnode Call ("Overlap" off-hook dialling)

In this scenario the instrument associated with device D1 is manually lifted to initiate a call, and the call is routed out of the CSTA sub-domain. The device dials a trunk or a trunk group prefix first and then the external number. Device D3 is not within the CSTA domain and therefore cannot be monitored.

	AFTER   AFTER    D1	i dunk group preux instand then the external number. Device D3 is not within the C3.1A domain and dieferor		D3	
	AFTER DI	nen die external n		(1) (1)	
by or a truink group.  D2		ciais a cruin	BEFORE	D1	
	BEFORE  D1				

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Device D1 goes off-hook	ServiceInitiatedEvent					
and receives dial tone.	• initiatedConnection	D1C1				
	<ul> <li>localConnectionInfo</li> </ul>	Initiated				
	• cause	newCall				
Device DI completes	NetworkReachedEvent		NetworkReachedEvent			The Called Device
dialling the trunk access.	outboundConnection	D2C1	<ul> <li>outboundConnection</li> </ul>	D2C1		information will be complete
The call leaves the CSTA	• trunkUsed	D2	• trunkUsed	D2/NR		only when DI completes its
domain.	• calledDevice	NK	<ul> <li>calledDevice</li> </ul>	NK		dialling. Within a networking
	• localConnectionInfo	Initiated	<ul> <li>localConnectionInfo</li> </ul>	Connected		environment the trunk may
	• correlatorData	$\sim$	ullet correlator $Data$	$\sim$		not be visible.
	• cause	newCall	• cause	newCall		
Device DI completes	OriginatedEvent		OriginatedEvent			
dialling.	<ul> <li>originatedConnection</li> </ul>	D1C1	<ul> <li>originatedConnection</li> </ul>	DICI		
	<ul> <li>callingDevice</li> </ul>	D1/NR	<ul> <li>callingDevice</li> </ul>	DI		
	• calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3		
	<ul> <li>originatingDevice</li> </ul>	D2	ullet originating $Device$	D2		
	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Connected		
	• correlatorData	$\sim$	ullet correlator $Data$	$\sim$		
	• cause	newCall	• cause	newCall		
Device DI receives an	EstablishedEvent		EstablishedEvent			Depending on the type of
answer from D3 or the	<ul> <li>establishedConnection</li> </ul>	D2C1	<ul> <li>establishedConnection</li> </ul>	D2C1		signalling employed over the
switch assumes an answer	answeringDevice	D3	<ul> <li>answeringDevice</li> </ul>	D3		trunk a delivered event may
based upon a timer.	callingDevice	D1	<ul> <li>callingDevice</li> </ul>	DI		be seen.
,	calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3		
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
	<ul> <li>originatingConnection</li> </ul>	D2CI	<ul> <li>originatingConnection</li> </ul>	D2CI		
	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Connected		
	• correlatorData	$\sim$	ullet correlator $D$ ata	$\circ$		
	• cause	network	• cause	network		
		Signal		Signal		

# Successful ISDN ("En-bloc") Call

In this scenario the instrument associated with device D1 manually initiates a call, and the call is routed out of the CSTA sub-domain via an ISDN trunk.

D3 CSTA Domain AFTER DI D2 BEFORE D1

Activity	MONITORED DEVICE DI		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Device DI goes off-hook	ServiceInitiatedEvent					
and moonings dial tong	• initiated Connection	D1C1				
and receives and tone.	• localConnectionInfo	Initiated				
	• cause	newCall				
Device DI completes	OriginatedEvent					
dialling.	<ul> <li>originatedConnection</li> </ul>	D1C1				
	<ul> <li>callingDevice</li> </ul>	D1/NR				
	<ul> <li>calledDevice</li> </ul>	D3				
	<ul> <li>originatingDevice</li> </ul>	D2				
	ullet local $Connection Info$	Connected				
	ullet correlator $Data$	$\mathcal{C}$				
	• cause	newCall				
The call leaves the CSTA	NetworkReachedEvent		NetworkReachedEvent			
domain.	<ul> <li>outboundConnection</li> </ul>	D2C1	<ul> <li>outboundConnection</li> </ul>	D2C1		
	<ul> <li>trunkUsed</li> </ul>	D2	<ul> <li>trunkUsed</li> </ul>	D2/NR		
	<ul> <li>calledDevice</li> </ul>	D3	<ul> <li>calledDevice</li> </ul>	D3		
	ullet local $Connection Info$	Connected	ullet local $Connection Info$	Connected		
	ullet correlator $D$ ata	$\mathcal{C}$	ullet correlator $D$ ata	$\sim$		
	• cause	newCall	• cause	newCall		
Device D3 is being alerted DeliveredEvent	DeliveredEvent		DeliveredEvent			Corresponds to the Alert
and DI listens to ringing	<ul> <li>connection</li> </ul>	D2C1	<ul> <li>connection</li> </ul>	D2C1		message in ISDN. Device D2
tone.	<ul> <li>alertingDevice</li> </ul>	D3	<ul> <li>alertingDevice</li> </ul>	D3		acts as a proxy device for
	<ul> <li>callingDevice</li> </ul>	D1	<ul> <li>callingDevice</li> </ul>	DI		device D3.
	<ul> <li>calledDevice</li> </ul>	D3	<ul> <li>calledDevice</li> </ul>	D3		
	<ul> <li>IastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
	<ul> <li>originatingConnection</li> </ul>	D2CI	<ul> <li>originatingConnection</li> </ul>	D2C1		
	ullet local $Connection Info$	Connected	ullet local $Connection Info$	Connected		
	ullet correlator $D$ ata	$\sim$	ullet correlator $D$ ata	$\sim$		
	• cause	network	• cause	network		
		Signal		Signal		

Device D3 answers.	EstablishedEvent		EstablishedEvent		Corresponds to the Answer
	• establishedConnection	D2C1	<ul> <li>establishedConnection</li> </ul>	D2C1	message in ISDN.
	<ul> <li>answeringDevice</li> </ul>	D3	<ul> <li>answeringDevice</li> </ul>	D3	)
	callingDevice	D1	<ul> <li>callingDevice</li> </ul>	DI	
	calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3	
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	<ul> <li>originating Connection</li> </ul>	D2CI	<ul> <li>originating Connection</li> </ul>	D2CI	
	• localConnectionInfo	Connected	ullet local Connection Info	Connected	
	• correlatorData	$\Box$	ullet correlator $Data$	$\overline{}$	
	• cause	network	• cause	network	
		Signal		Signal	

### 5 Call Release Scenarios

This section includes examples of calling and called parties releasing. The latter is an example of how a call can reside in the CSTA blocked call state.

### 5.1 Call Release

Device D1 and D2 are in an established call. Device D1 decides to clear the call - a local end disconnect.

BEFORE AFTER  $D_{1} \leftarrow C_{1} \leftarrow D_{2}$   $D_{1} = D_{2}$   $D_{2}$ 

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Clear Connection service   ClearConnectionRequest	ClearConnectionRequest					
invoked on behalf of	<ul> <li>connectionToBeCleared</li> </ul>	D1C1				
device DI.	• extensions	$\sim$				
Acknowledgement.	ClearConnectionResult					
	<ul> <li>extensions</li> </ul>	$\sim$				
Connection DICI is	ConnectionClearedEvent		ConnectionClearedEvent			A connection cleared event
cleared.	<ul> <li>droppedConnection</li> </ul>	D1C1	<ul> <li>droppedConnection</li> </ul>	DICI		does not necessarily mean
	<ul> <li>releasingDevice</li> </ul>	D1/NR	<ul> <li>releasingDevice</li> </ul>	DI		that a handset has been
	<ul> <li>localConnectionInfo</li> </ul>	Null	<ul> <li>localConnectionInfo</li> </ul>	Connected		replaced. An application will
	<ul> <li>correlatorData</li> </ul>	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$		determine from the
	• cause	normal	• cause	normal		connection cleared event
		Clearing		Clearing		who initiated clearing.
Connection D2C1 cleared			ConnectionClearedEvent			
as a result of DICI			<ul> <li>droppedConnection</li> </ul>	D2C1		
clearing.			<ul> <li>releasingDevice</li> </ul>	D2/NR		
)			<ul> <li>localConnectionInfo</li> </ul>	Null		
			• correlatorData	$\circ$		
			• cause	normal		
				Clearing		

# 5.2 The CSTA Blocked Call State

Device D1 and D2 are in an established call. Device D2 decides to clear the call - a far end disconnect which places the call in the blocked state.

BEFORE

AFTER  $D1 \leftarrow C1 \leftarrow D2$  D1 D2

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Clear Connection service			ClearConnectionRequest			When a call consists of only
invoked on behalf of			<ul> <li>connectionToBeCleared</li> </ul>	D2C1		two devices a
device D2.			• extensions	$\sim$		ClearConnection is
						equivalent to ClearCall.
Acknowledgement.			ClearConnectionResult			
			<ul> <li>extensions</li> </ul>	$\sim$		
Device D2 hangs up first.	ConnectionClearedEvent		ConnectionClearedEvent			Device DI is still off-hook.
	droppedConnection	D2C1	<ul> <li>droppedConnection</li> </ul>	D2C1		
	releasingDevice	D2	<ul> <li>releasing Device</li> </ul>	D2/NR		
	• localConnectionInfo	Connected	<ul> <li>localConnectionInfo</li> </ul>	Null		
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	normal	• cause	normal		
		Clearing		Clearing		
Connection DICI moves	FailedEvent					The CSTA Simple Call state
into the failed state.	failedConnection	D1C1				Blocked applies when the
	failingDevice	D1/NR				local connection, DICI, is
	calledDevice	D2				in the Failed state and the
	• localConnectionInfo	Failed				other connection, D2CI, is
	• correlatorData	$\sim$				in the Null state.
	• cause	blocked				
Device DI hangs up.	ConnectionClearedEvent					
	droppedConnection	D1C1				
	<ul> <li>releasingDevice</li> </ul>	D1/NR				
	• localConnectionInfo	Null				
	• correlatorData	$\sim$				
	• cause	normal				
		Clearing				

### Consultation Calls

This section includes examples of invoking a consultation call which is rejected, making a consultation call where the consulted party clears, making a consultation call where the held party clears, and invoking enquiry calls to non-existent devices.

## 6.1 Successful Consultation Call

A call between device D1 and device D2 is already connected. Device D1 decides to invoke a consultation call to another device, D3. This call is successful.

BEFORE  $D1 \leftarrow C1 \leftarrow D2$   $D1 \rightarrow C \leftarrow D2$   $D1 \rightarrow C \leftarrow D2$   $C2 \rightarrow C \rightarrow D3$ 

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Consultation Call service	ConsultationCallRequest					
to device D3 is invoked on	<ul> <li>existingCall</li> </ul>	DICI				
behalf of device D1.	• consultedDevice	D3				
	<ul> <li>consultedDeviceProfile</li> </ul>	$\sim$				
	• accountCode	$\sim$				
	• authCode	$\sim$				
	• correlatorData	$\sim$				
	<ul> <li>extensions</li> </ul>	()				
Acknowledgement.	ConsultationCallResult					
	initiatedCall	D1C2				
	• extensions	()				
Connection placed on	HeldEvent		HeldEvent			In some implementations the
hold.	<ul> <li>heldConnection</li> </ul>	D1C1	<ul> <li>heldConnection</li> </ul>	DICI		cause consultation may not
	• holdingDevice	D1/NR	<ul> <li>holdingDevice</li> </ul>	D1		be supported.
	<ul> <li>localConnectionInfo</li> </ul>	Hold	ullet local Connection Info	Connected		
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	consultation	• cause	consultation		
Application notified of	ServiceInitiatedEvent					The generation of this event
initiating call.	<ul> <li>initiatedConnection</li> </ul>	D1C2				is switch-specific.
	<ul> <li>localConnectionInfo</li> </ul>	Initiated				
	• cause	newCall				
The Consultation Call is	OriginatedEvent					At this point a consultation
launched.	<ul> <li>originatedConnection</li> </ul>	D1C2				call may follow any of the
	<ul> <li>callingDevice</li> </ul>	D1/NR				MakeCall scenarios
	• calledDevice	D3				produced in this Technical
	<ul> <li>originatingDevice</li> </ul>	D2				Report.
	ullet local Connection Info	Connected				
	• correlatorData	$\sim$				
	• cause	newCall				

										4	. the								
										Assuming that the called	device actually answers the	call.							
	D3C2	D3/NR	DI	D3	NR	$\sim$	Alerting	$\sim$	newCall		D3C2	D3/NR	DI	D3	NR	$\sim$	Connected	$\sim$	newCall
DeliveredEvent	• connection	alertingDevice	callingDevice	calledDevice	<ul> <li>lastRedirectionDevice</li> </ul>	<ul> <li>originatingConnection</li> </ul>	<ul> <li>localConnectionInfo</li> </ul>	• correlatorData	• cause	EstablishedEvent	<ul> <li>establishedConnection</li> </ul>	answeringDevice	callingDevice	calledDevice	• lastRedirectionDevice	<ul> <li>originatingConnection</li> </ul>	<ul> <li>localConnectionInfo</li> </ul>	• correlatorData	osnos.
	D3C2	D3	D1	D3	NR	$\sim$	Connected	$\sim$	newCall		D3C2	D3	D1	D3	NR	$\sim$	Connected	$\sim$	newCall
DeliveredEvent	• connection	alertingDevice	• callingDevice	• calledDevice	<ul> <li>lastRedirectionDevice</li> </ul>	<ul> <li>originating Connection</li> </ul>	<ul> <li>localConnectionInfo</li> </ul>	• correlatorData	• cause	EstablishedEvent	<ul> <li>establishedConnection</li> </ul>	answeringDevice	• callingDevice	calledDevice	• lastRedirectionDevice	<ul> <li>originating Connection</li> </ul>	<ul> <li>localConnectionInfo</li> </ul>	• correlatorData	• cause
Device D3 begins to alert.   DeliveredEvent	1									New call established.									

### Consultation Call Rejected 6.2

and the r where

		$\begin{bmatrix} D_1 \\ -c \end{bmatrix}$ $\begin{bmatrix} c \end{bmatrix}$ $\begin{bmatrix} c \end{bmatrix}$ $\begin{bmatrix} c \end{bmatrix}$	D1 -c- $C1$ -c- $D2$
		AFTER	BEFORE
		perators may not be placed on hold, or a network hold request which is rejected.	operators may not be placed on he
ls which involve an operator w	is for not allowing hold include call	Aake Call service. The Hold Call request fails because a hold is not allowed. Reasons for not allowing hold include calls which involve an operator w	Make Call service. The Hold Call
n of the Hold Call service and	which provides the compound actio	n active call, invokes the Consultation Call service which provides the compound action of the Hold Call service and	Device D1, which already has an active

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2	MONITORED DEVICE D3	Comments
Application issues	ConsultationCallRequest				
Consultation Call which	existingCall	DICI			
invokes a hold call	calledDirectoryNumber	D3			
request.	• calledDeviceProfile	$\sim$			
	• accountCode	$\sim$			
	• authCode	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Hold request rejected.	ConsultationCallError				Example - an operator may
	operationalError	invalid			not be placed on hold.
		Feature			

### **6.3** Consulted party busy

Device D1 is active in a call with device D2 and decides to invoke a consultation call to another device. The consultation call fails because the specified destination device is busy. Device D1 clears the enquiry call.

BEFORE  $D1 \xrightarrow{c} C1 \xrightarrow{c} D2$   $D1 \xrightarrow{h} C1 \xrightarrow{c} D2$   $C2 \xrightarrow{f} D3$ 

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Consultation Call service	ConsultationCallRequest					
to device D3 is invoked on	• existing Call	D1C1				
behalf of device D1.	<ul> <li>calledDirectoryNumber</li> </ul>	D3				
	ullet called Device Profile	$\sim$				
	• accountCode	$\sim$				
	• authCode	$\sim$				
	• correlatorData	$\sim$				
	• extensions	()				
Acknowledgement.	ConsultationCallResult					
	initiatedCall	D1C2				
	• extensions	$\sim$				
Call CI is put on hold.	HeldEvent		HeldEvent			In some implementations the
	heldConnection	D1C1	<ul> <li>heldConnection</li> </ul>	DICI		cause consultation may not
	<ul> <li>holdingDevice</li> </ul>	D1/NR	<ul> <li>holdingDevice</li> </ul>	D1		be supported.
	<ul> <li>localConnectionInfo</li> </ul>	Hold	$\bullet \ local Connection Info$	Connected		
	• correlatorData	$\sim$	ullet correlator $D$ ata	$\sim$		
	• cause	consultation	• cause	consultation		
Application notified of	ServiceInitiatedEvent					
initiating call.	<ul> <li>initiatedConnection</li> </ul>	D1C2				
	• localConnectionInfo	Initiated				
	• cause	newCall				
Device DI completes	OriginatedEvent					At this point a consultation
dialling the consultation	<ul> <li>originatingDevice</li> </ul>	D1/NR				call may follow any of the
party number.	<ul> <li>originatedConnection</li> </ul>	D1C2				MakeCall scenarios
	<ul> <li>callingDevice</li> </ul>	D1/NR				produced in this Technical
	<ul> <li>calledDevice</li> </ul>	D3				Report.
	<ul> <li>originatingDevice</li> </ul>	D2				
	• localConnectionInfo	Connected				
	• correlatorData	$\sim$				
	• cause	newCall				

The connection to device	FailedEvent		FailedEvent		
D3 fails due to busy.	<ul> <li>failedConnection</li> </ul>	D3C2	failedConnection	D3C2	
	<ul> <li>failingDevice</li> </ul>	D3	<ul> <li>failingDevice</li> </ul>	D3/NR	
	• calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3	
	• localConnectionInfo	Connected	<ul> <li>localConnectionInfo</li> </ul>	Failed	
	• correlatorData	$\sim$	ullet correlator $D$ ata	$\Box$	
	• cause	Busy	• cause	Busy	
Originator clears call C2	ConnectionClearedEvent		ConnectionClearedEvent		
and does not invoke the	<ul> <li>droppedConnection</li> </ul>	D1C2	<ul> <li>droppedConnection</li> </ul>	D1C2	
call completion service.	<ul> <li>releasingDevice</li> </ul>	D1/NR	<ul> <li>releasingDevice</li> </ul>	DI	
,	<ul> <li>localConnectionInfo</li> </ul>	Null	<ul> <li>localConnectionInfo</li> </ul>	Failed	
	• correlatorData	$\Box$	ullet correlator $D$ ata	$\Box$	
	• cause	normal	• cause	normal	
		Clearing		Clearing	
Failed connection D3C2			ConnectionClearedEvent		Device D1 may now go on
clears.			<ul> <li>droppedConnection</li> </ul>	D3C2	and reconnect to the call on
			<ul> <li>releasingDevice</li> </ul>		hold. This may be automatic.
			<ul> <li>localConnectionInfo</li> </ul>	Null	
			ullet correlator $D$ ata	$\Box$	
			• cause	normal	
				Clearing	

# 6.4 Consulted party clears - consulting party receives dial tone

Device D1 is active in a call with device D2 and decides to make a Consultation call to another device. The consultation call is successful and when complete the consulted party clears. Device D1 is then provided with dial tone to enable them to either invoke the retrieve call service to reconnect to the held party or make another consultation call.

AFTER	D1 $h$ $C1$ $c$ $D2$	(S)
BEFORE	$D_1 - h C_1 - D_2$	$\begin{bmatrix} c \\ C2 \end{bmatrix} c - \begin{bmatrix} D3 \end{bmatrix}$

Activity	MONITORED DEVICE D1	MONITORED DEVICE D2	MONITORED DEVICE D3		Comments
The Consulted party			ClearConnectionRequest		
clears the call using the			<ul> <li>connectionToBeCleared</li> </ul>	D3C2	
Clear Connection Service.			• extensions	()	
Acknowledgement.			ClearConnectionResult		
			• extensions	()	
Connection D3C2 clears.	ConnectionClearedEvent		ConnectionClearedEvent		
	droppedConnection     D3C2		droppedConnection	D3C2	
	• releasingDevice D3		<ul> <li>releasingDevice</li> </ul>	D3/NR	
	• localConnectionInfo Connected		• localConnectionInfo	Null	
	• correlatorData ()		• correlatorData	$\sim$	
	• cause normal		• cause	normal	
	Clearing			Clearing	
Switch clears connection	ConnectionClearedEvent				
DIC2.	droppedConnection     D1C2				
	• releasingDevice D1/NR				
	• localConnectionInfo Null				
	• correlatorData ()				
	• cause normal				
	Clearing				
The switch provides dial	ServiceInitiatedEvent				
tone for a new call or	initiatedConnection     D1C3				
feature invocation - call	localConnectionInfo     Initiated				
C3.	• cause newCall				

#### Held party clears 6.5

Device D1 is active in a Consultation call with device D3; the connection to device D2 is on hold. During the Consultation call device D2, the held party,

decides to clear. The held call and associated resources are cleared while the active call continues uninterrupted. AFTER D1 BEFORE DI

		,				
Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Clear Connection service			ClearConnectionRequest			
invoked on behalf of			connectionToBeCleared	D2C1		
device D2.			• extensions	$\sim$		
Acknowledgement.			ClearConnectionResult			
			• extensions	$\sim$		
Connection D2CI clears.	ConnectionClearedEvent		ConnectionClearedEvent			
	droppedConnection	D2C1	droppedConnection	D2C1		
	releasingDevice	D2	releasing Device	D2/NR		
	• localConnectionInfo	Hold	• localConnectionInfo	Null		
	• correlatorData	$\sim$	• correlatorData	$\sim$		
	• cause	normal	• cause	normal		
		Clearing		Clearing		
Automatic clearing of	ConnectionClearedEvent					The Call between devices D1
connection DICI.	droppedConnection	D1C1				and D3 continues
	releasingDevice	D1/NR				uninterrupted.
	localConnectionInfo	Null				•
		$\sim$				
	• cause	normal				
		Cloaning		_		

#### Successful Alternate Call

The Alternate Call service provides the compound action of the Hold Call service followed by the Retrieve Call service. An existing active call is placed on hold and a previously held call is re-connected as illustrated by this scenario.

AFTER DI FC D3 BEFORE D1

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
Alternate Call service is	AlternateCallRequest						
invoked on behalf of	• heldCall	D1C1					
device DI.	activeCall	D1C2					
	• extensions	$\sim$					
Acknowledgement.	AlternateCallResult						
	• extensions	$\sim$					
Connection to D3 is put	HeldEvent				HeldEvent		In some implementations the
on hold.	heldConnection	D1C2			<ul> <li>heldConnection</li> </ul>	D1C2	cause value "Alternate" may
	<ul> <li>holdingDevice</li> </ul>	D1/NR			<ul> <li>holdingDevice</li> </ul>	D1	not be supported.
	<ul> <li>localConnectionInfo</li> </ul>	Hold			ullet local $Connection Info$	Connected	
	<ul> <li>correlatorData</li> </ul>	$\sim$			ullet correlator $D$ ata	$\Box$	
	• cause	Alternate			• cause	Alternate	
Switch acknowledges	RetrievedEvent		RetrievedEvent				
retrieval of the held call to • retrievedConnection	<ul> <li>retrievedConnection</li> </ul>	D1C1	<ul> <li>retrievedConnection</li> </ul>	DICI			
device D2.	<ul> <li>retrievingDevice</li> </ul>	D1/NR	<ul> <li>retrievingDevice</li> </ul>	DI			
	<ul> <li>localConnectionInfo</li> </ul>	Connected	<ul> <li>localConnectionInfo</li> </ul>	Connected			
	<ul> <li>correlatorData</li> </ul>	$\sim$	• correlatorData	$\sim$			
	• cause	Alternate	• cause	Alternate			

#### 8 Transfers

This section includes examples of a supervised and unsupervised transfer and a single step transfer.

## 8.1 Successful Supervised Transfer

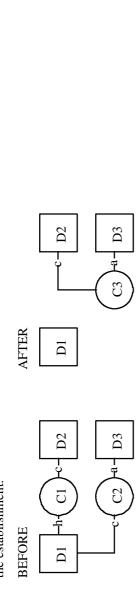
Device D1 has a call to device D3 connected and a call to device D2 which is on hold. The Transfer Call service provides the ability to connect devices D2 and D3. Device D1 drops out of the call. This is a supervised transfer because device D1 established the call to device D3 before transferring the call.

BEFORE  $D1 \rightarrow h \leftarrow C1 \rightarrow C \rightarrow D2$   $D1 \rightarrow h \leftarrow C1 \rightarrow C \rightarrow D2$   $C2 \rightarrow C \rightarrow D3$   $C3 \rightarrow C \rightarrow D3$ 

Activity	MONITORED DEVICE DI		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
Transfer Call service is	TransferCallRequest						
invoked on behalf of	• heldCall	DICI					
device DI.	<ul> <li>activeCall</li> </ul>	D1C2					
	<ul> <li>extensions</li> </ul>	()					
Acknowledgement.	TransferCallResult						
	<ul> <li>transferredCall</li> </ul>	D3C3					
	<ul> <li>new Connection</li> </ul>	D2C3					
	• deviceID	D2					
	<ul> <li>oldConnection</li> </ul>	D2CI					
	<ul> <li>newConnection</li> </ul>	D3C3					
	ullet deviceID	D3					
	<ul> <li>oldConnection</li> </ul>	D3C2					
	<ul> <li>extensions</li> </ul>	$\sim$					
Calls between DI,D2 and	TransferredEvent		TransferredEvent		TransferredEvent		The result is that
OI, D3 are released. The	<ul> <li>primaryOldCall</li> </ul>	D1C1	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D3C2	connections D2C1, D1C1,
connections between D2,	<ul> <li>secondaryOldCall</li> </ul>	D1C2	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	D1C2 and D3C2 are
id D3,D1 are	<ul> <li>transferringDevice</li> </ul>	D1/NR	<ul> <li>transferringDevice</li> </ul>	D1	<ul> <li>transferringDevice</li> </ul>	D1	cleared. SecondaryOldCall
replaced with a single	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3/NR	parameter is only included if
connection between D2	<ul> <li>new Connection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	events have been reported
and D3.	ullet deviceID	D2	ullet deviceID	D2	ullet deviceID	D2	for the connection identifier.
	<ul> <li>oldConnection</li> </ul>	D2CI	<ul> <li>oldConnection</li> </ul>	D2C1	<ul> <li>oldConnection</li> </ul>	D2C1	,
	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	
	• deviceID	D3	• deviceID	D3	ullet deviceID	D3	
	<ul> <li>oldConnection</li> </ul>	D3C2	• oldConnection	D3C2	• oldConnection	D3C2	
	<ul> <li>localConnectionInfo</li> </ul>	Null	<ul> <li>localConnectionInfo</li> </ul>	Connected	<ul> <li>localConnectionInfo</li> </ul>	Connected	
	ullet correlator $Data$	$\Box$	<ul> <li>correlatorData</li> </ul>	$\sim$	ullet correlator $D$ ata	$\sim$	
	• cause	Transfer	dShDJ •	Transfer	• cause	Transfer	

## 8.2 Successful Unsupervised Transfer

devices D2 and D3. Device D1 drops out of the call. This is an unsupervised transfer because device D1 transferred the call to device D3 before completing Device D1 has a call to device D3 which is alerting and a call to device D2 which is on hold. The Transfer Call service provides the ability to connect the establishment.



Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
Transfer Call service is	TransferCallRequest						
invoked on behalf of	• heldCall	D1C1					
device DI.	<ul> <li>activeCall</li> </ul>	D1C2					
	• extensions	()					
Transfer call is accepted.	TransferCallResult						
	<ul> <li>transferredCall</li> </ul>	D3C3					
	<ul> <li>newConnection</li> </ul>	D2C3					
	ullet deviceID	D2					
	<ul> <li>oldConnection</li> </ul>	D2CI					
	<ul> <li>newConnection</li> </ul>	D3C3					
	• deviceID	D3					
	<ul> <li>oldConnection</li> </ul>	D3C2					
	<ul> <li>extensions</li> </ul>	$\Box$					
Calls between DI,D2 and	TransferredEvent		TransferredEvent		TransferredEvent		The result is that
DI,D3 are released.	<ul> <li>primaryOldCall</li> </ul>	D1C1	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D3C2	connections DICI,DIC2,
Connections between	<ul> <li>secondaryOldCall</li> </ul>	D1C2	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	D2C1 and D3C2 are
D2,D1 and D3,D1 are	<ul> <li>transferringDevice</li> </ul>	D1/NR	<ul> <li>transferringDevice</li> </ul>	D1	<ul> <li>transferringDevice</li> </ul>	D1	cleared. SecondaryOldCall
replaced with a single	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3/NR	parameter is only included if
connection between D2,	<ul> <li>newConnection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	events have been reported
<i>D3</i> .	ullet deviceID	D2	ullet deviceID	D2	• deviceID	D2	for the connection identifier.
	<ul> <li>oldConnection</li> </ul>	D2C1	<ul> <li>oldConnection</li> </ul>	D2CI	<ul> <li>oldConnection</li> </ul>	D2CI	
	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	
	ullet deviceID	D3	ullet deviceID	D3	ullet deviceID	D3	
	<ul> <li>oldConnection</li> </ul>	D3C2	<ul> <li>oldConnection</li> </ul>	D3C2	• oldConnection	D2C2	
	ullet local Connection Info	Null	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Alerting	
	ullet correlator $Data$	$\Box$	<ul> <li>correlatorData</li> </ul>	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	
	• cause	Transfer	• cause	Transfer	• cause	Transfer	

#### Single Step Transfer 8.3

Device D1 has an active call to device D2 and invokes the Single Step Transfer service to transfer the call to device D3. Device D1 does not participate in the new call and clears. <del>)-a-</del> D3 D2 AFTER DI D3 BEFORE D1

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Single Step Transfer	SingleStepTransferRequest					
service is invoked on	activeCall	D1C1				
behalf of device D1.	<ul> <li>deviceToTransferTo</li> </ul>	D3				
	ullet transfer $ToDevice Profile$	$\Box$				
	• accountCode	$\sim$				
	• authCode	$\sim$				
	• correlatorData	$\sim$				
	• extensions	()				
Single Step Transfer is	SingleStepTransferResult					
accepted.	• transferredCall	D3C2				
	• new Connection	D2C2				
	• deviceID	D2				
	• oldConnection	D2C1				
	• new Connection	D3C2				
	• deviceID	D3				
	• extensions	()				
The call between DI, D2 is	TransferredEvent		TransferredEvent			As there is no connection to
replaced with an alerting	primaryOldCall	D1C1	<ul> <li>primaryOldCall</li> </ul>	D2C1		D3 at this time it will not see
call between D2,D3.	• secondaryOldCall	$\Box$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$		the transferred event.
	transferringDevice	D1/NR	<ul> <li>transferringDevice</li> </ul>	DI		
	transferredDevice	D3	<ul> <li>transferredDevice</li> </ul>	D3		
	• new Connection	D2C2	<ul> <li>newConnection</li> </ul>	D2C2		
	• deviceID	D2	ullet deviceID	D2		
	• oldConnection	D2C1	<ul> <li>oldConnection</li> </ul>	D2C1		
	• new Connection	D3C2	<ul> <li>newConnection</li> </ul>	D3C2		
	• deviceID	D3	ullet deviceID	D3		
	• localConnectionInfo	Null	ullet local Connection Info	Connected		
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	Single Step	• cause	SingleStep		
		Transfer		Transfer		

Device D3 begins to ring.	DeliveredEvent		DeliveredEvent		This event reflects the state
	• connection	D3C2	• connection	D3C2	change for connection
	alerting Device	D3	<ul> <li>alertingDevice</li> </ul>	D3/NR	D3C2.
	calling Device	D2	<ul> <li>callingDevice</li> </ul>	D2	
	calledDevice	D3	• calledDevice	D3	
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	originating Connection	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\sim$	
	localConnectionInfo	Connected	<ul> <li>localConnectionInfo</li> </ul>	Alerting	
	correlatorData	$\sim$	• correlatorData	$\sim$	
	• cause	newCall	• cause	single Step	
				Transfer	

#### Transfer on Busy 8.4

Device D1 has an active call to device D2 and invokes a Consultation call with device D3. Device D3 is busy but device D1 still transfers the call so that D2

camps-on to device D3. The scenario below illustrates the events once the transfer has been invoked. 년 - D3 AFTER D BEFORE D1

	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Cc	Comments
Transfer Call service is	TransferCallRequest						
	• heldCall	DICI					
device DI.	<ul> <li>activeCall</li> </ul>	D1C2					
	<ul> <li>extensions</li> </ul>	()					
Acknowledgement.	TransferCallResult						
	<ul> <li>transferredCall</li> </ul>	D3C3					
	<ul> <li>new Connection</li> </ul>	D2C3					
	ullet deviceID	D2					
	<ul> <li>oldConnection</li> </ul>	D2CI					
	• new Connection	D3C3					
	• deviceID	D3					
	<ul> <li>oldConnection</li> </ul>	D3C2					
	• extensions	()					
The calls between D1,D2	TransferredEvent		TransferredEvent		TransferredEvent		
	<ul> <li>primaryOldCall</li> </ul>	DICI	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D3C2	
The connections between	<ul> <li>secondaryOldCall</li> </ul>	D1C2	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\Box$	
	<ul> <li>transferringDevice</li> </ul>	D1/NR	<ul> <li>transferringDevice</li> </ul>	DI	<ul> <li>transferringDevice</li> </ul>	D1	
replaced with a single	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3	<ul> <li>transferredDevice</li> </ul>	D3/NR	
queued call between D2,	<ul> <li>new Connection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	
D3.	ullet deviceID	D2	ullet deviceID	D2	ullet deviceID	D2	
	<ul> <li>oldConnection</li> </ul>	D2CI	ullet old $Connection$	D2C1	<ul> <li>oldConnection</li> </ul>	D2CI	
	• new Connection	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	
	$\bullet$ deviceID	D3	ullet deviceID	D3	• deviceID	D3	
	<ul> <li>oldConnection</li> </ul>	D3C2	<ul> <li>oldConnection</li> </ul>	D3C2	• oldConnection	D3C2	
	ullet local Connection Info	Null	ullet local Connection Info	Connected	ullet local Connection Info	Failed	
	<ul> <li>correlatorData</li> </ul>	$\sim$	<ul> <li>correlatorData</li> </ul>	$\Box$	ullet correlator $D$ ata	$\Box$	
	• cause	Transfer	• cause	Transfer	• cause	Transfer	

The call queues at device	QueuedEvent		QueuedEvent		
D3.	queuedConnection	D3C3	queuedConnection	D3C3	
	• dnene	D3	• dnene	D3/NR	
	callingDevice	D2	callingDevice	D2	
	calledDevice	D3	calledDevice	D3	
	lastRedirectionDevice	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	• numberedQueued	$\sim$	• numberedQueued		
	callsInFront	$\sim$	• callsInFront		
	localConnectionInfo	Connected	<ul> <li>localConnectionInfo</li> </ul>	Quened	
	correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\overline{}$	
	• cause	campON	• cause	campON	
Device D3 becomes free	DeliveredEvent		DeliveredEvent		
and begins to ring.	• connection	D3C3	• connection	D3C3	
	alerting Device	D3	<ul> <li>alertingDevice</li> </ul>	D3/NR	
	callingDevice	D2	<ul> <li>callingDevice</li> </ul>	D2	
	calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3	
	lastRedirectionDevice	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	originating Connection	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\overline{}$	
	localConnectionInfo	Connected	<ul> <li>localConnectionInfo</li> </ul>	Alerting	
	correlatorData	$\sim$	• correlatorData		
	• cause	newCall	• cause	newCall	
Device D3 answers the	EstablishedEvent		EstablishedEvent		
call.	establishedConnection	D3C3	<ul> <li>establishedConnection</li> </ul>	D3C3	
	answeringDevice	D3	<ul> <li>answeringDevice</li> </ul>	D3/NR	
	callingDevice	D2	<ul> <li>callingDevice</li> </ul>	D2	
	calledDevice	D3	<ul> <li>calledDevice</li> </ul>	D3	
	lastRedirectionDevice	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	originating Connection	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\overline{}$	
	localConnectionInfo	Connected	ullet $local Connection Info$	Connected	
	correlatorData	$\sim$	• correlatorData	$\overline{}$	
	• cause	newCall	• cause	newCall	

#### Conference Calls

6

Examples of simple Conference calls and more complex multi party Conference calls where individual parties join conferences, including a Single Step Conference.

### 9.1 Conference Call Service

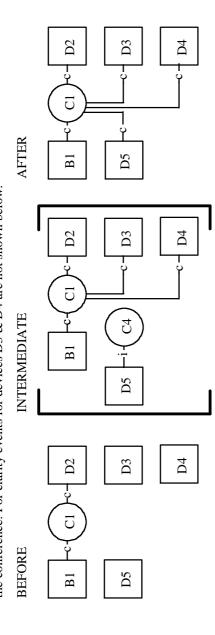
the ability to conference the three devices into a single call. hold This Davine D1 has two

Device D1 has two calls, one active call and the other call on hold. This service provides the ability	BEFORE	$D_1$ $h$ $C_1$ $C_2$ $D_2$ $D_1$ $D_1$ $C_2$ $D_2$	$\begin{bmatrix} c \\ c_2 \end{pmatrix} c - \begin{bmatrix} D_3 \end{bmatrix} \qquad \begin{bmatrix} c \\ c_3 \end{pmatrix} c - \begin{bmatrix} D_3 \end{bmatrix}$
Device	BEFO		

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
Conference Call service is   ConferenceCallRequest	ConferenceCallRequest						
invoked on behalf of	• heldCall	DICI					
device DI.	• extensions	) ()					
Acknowledgement.	ConferenceCallResult						
)	• conferenceCall	D1C3					
	• new Connection	D2C3					
	• deviceID	D2					
	• oldConnection	D2CI					
	• new Connection	D3C3					
	ullet deviceID	D3					
	• oldConnection	D3C2					
	• extensions	$\sim$					
Conference established.	ConferencedEvent		ConferencedEvent		ConferencedEvent		The added party is the device
	primaryOldCall	DICI	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D3C2	representing the person who
	<ul> <li>secondaryOldCall</li> </ul>	D1C2	ullet secondary $Old Call$	$\sim$	ullet secondary Old Call	$\sim$	has just joined the call from
	<ul> <li>confController</li> </ul>	D1/NR	<ul> <li>confController</li> </ul>	D1	<ul> <li>confController</li> </ul>	D1	the perspective of the
	addedParty	D3	<ul> <li>addedParty</li> </ul>	D3	<ul> <li>addedParty</li> </ul>	D3/NR	participants.
	• new Connection	DIC3	<ul> <li>newConnection</li> </ul>	DIC3	<ul> <li>newConnection</li> </ul>	DIC3	
	ullet deviceID	DI	ullet deviceID	DI	ullet deviceID	DI	
	• new Connection	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	<ul> <li>newConnection</li> </ul>	D2C3	
	ullet deviceID	D2	ullet device $ID$	D2	ullet deviceID	D2	
	• oldConnection	D2CI	<ul> <li>oldConnection</li> </ul>	D2CI	<ul> <li>oldConnection</li> </ul>	D2CI	
	• new Connection	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	<ul> <li>newConnection</li> </ul>	D3C3	
	ullet deviceID	D3	ullet device $ID$	D3	• deviceID	D3	
	• oldConnection	D3C2	<ul> <li>oldConnection</li> </ul>	D3C2	<ul> <li>oldConnection</li> </ul>	D3C2	
	• localConnectionInfo	Connected	$\bullet \ local Connection Info$	Connected	$\bullet \ local Connection Info$	Connected	
	• correlatorData	$\sim$	ullet correlator $Data$	$\sim$	ullet correlator $D$ ata	$\sim$	
	• cause	newCall	• cause	newCall	• cause	newCall	

## 9.2 Multiple Party (Meet Me) Conference

the same call ID. This scenario represents the case when device D2 is the first to join the conference bridge B1 followed by another device, D5, joining into Device B1 is a conference bridge. Each device that joins the conference dials the conference access number. All the devices in the conference bridge have the conference. For clarity events for devices D3 & D4 are not shown below.

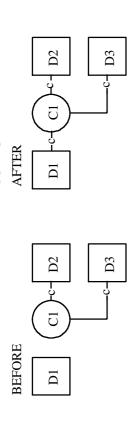


Activity	MONITORED DEVICE B1		MONITORED DEVICE D2		MONITORED DEVICE D5		Comments
Device B1 "answers" the	EstablishedEvent		EstablishedEvent				For a simple "two" party
call from device D2.	<ul> <li>establishedConnection</li> </ul>	D2C1	<ul> <li>establishedConnection</li> </ul>	D2C1			call an established event is
	<ul> <li>answeringDevice</li> </ul>	B1	<ul> <li>answeringDevice</li> </ul>	B1			generated.
	<ul> <li>callingDevice</li> </ul>	D2	<ul> <li>callingDevice</li> </ul>	D2			
	<ul> <li>calledDevice</li> </ul>	B1	<ul> <li>calledDevice</li> </ul>	B1			
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR			
	<ul> <li>originating Connection</li> </ul>	$\sim$	<ul> <li>originating Connection</li> </ul>	$\sim$			
	<ul> <li>localConnectionInfo</li> </ul>	Connected	<ul> <li>localConnectionInfo</li> </ul>	Connected			
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$			
	• cause	newCall	• cause	newCall			
	previous call establishments						
	and conferencing for devices						
	D3 and D4 take place here						
Device D5 dials into the					OriginatedEvent		Call IDs C2 and C3 were
conference bridge.					<ul> <li>originatedConnection</li> </ul>	D5C4	associated with the
					<ul> <li>callingDevice</li> </ul>	D5/NR	originated events for D3 and
					<ul> <li>calledDevice</li> </ul>	B1	D4 originating calls to the
					ullet originating $Device$	$\sim$	conference bridge.
					ullet local $Connection Info$	Connected	)
					ullet correlator $D$ ata	$\sim$	
					• cause	newCall	

Device B1 "answers" the ConferencedEvent	ConferencedEvent		ConferencedEvent		ConferencedEvent		D5 has a cause because
call.		B1C1	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D5C4	there is a new call ID.
	11	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	
		B1/NR	<ul> <li>confController</li> </ul>	B1	<ul> <li>confController</li> </ul>	B1	
		D5	<ul> <li>addedParty</li> </ul>	D5	<ul> <li>addedParty</li> </ul>	D5/NR	
		BICI	<ul> <li>newConnection</li> </ul>	BICI	<ul> <li>newConnection</li> </ul>	BICI	
		BI	ullet deviceID	BI	• deviceID	BI	
	• oldConnection	BICI	<ul> <li>oldConnection</li> </ul>	BICI	<ul> <li>oldConnection</li> </ul>	BICI	
		D2C1	<ul> <li>newConnection</li> </ul>	D2C1	<ul> <li>newConnection</li> </ul>	D2CI	
		D2	ullet deviceID	D2	• deviceID	D2	
		D2C1	<ul> <li>oldConnection</li> </ul>	D2C1	<ul> <li>oldConnection</li> </ul>	D2CI	
		D3CI	<ul> <li>newConnection</li> </ul>	D3CI	<ul> <li>newConnection</li> </ul>	D3CI	
		D3	ullet deviceID	D3	ullet deviceID	D3	
		D3CI	• oldConnection	D3CI	• oldConnection	D3CI	
		D4CI	<ul> <li>newConnection</li> </ul>	D4C1	<ul> <li>newConnection</li> </ul>	D4CI	
		D4	ullet deviceID	D4	ullet deviceID	D4	
		D4C1	<ul> <li>oldConnection</li> </ul>	D4CI	• oldConnection	D4CI	
		D5CI	<ul> <li>newConnection</li> </ul>	D5C1	<ul> <li>newConnection</li> </ul>	D5CI	
		D5	ullet deviceID	D5	ullet deviceID	D5	
		D5C4	<ul> <li>oldConnection</li> </ul>	D5C4	<ul> <li>oldConnection</li> </ul>	D5C4	
	ofulo	Connected	ullet local Connection Info	Connected	ullet local $Connection Info$	Connected	
	• correlatorData	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	ullet correlator $Data$	$\sim$	
	• cause	()	• cause	()	• cause	newCall	

### 9.3 Single Step Conference

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 Device D2 and device D3 are involved in a call C1. The Single Step Conference service is invoked on behalf of device D1 which wishes to silently join the scenario does not show device D1 being prompted to lift the handset or invoking the service from a feature button.



Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
SingleStepConference	SingleStepConfRequest						It is possible that some
ıalf	activeCall	D2C1					devices will be prohibited
	• deviceToJoin	D1					from joining calls.
	ullet participation $Type$	Silent					
	ullet joining Device Profile	$\sim$					
	• accountCode	$\sim$					
	• authCode	$\sim$					
	ullet correlator $D$ ata	$\Box$					
	• extensions	()					
Acknowledgement.	SingleStepConfResult						
	<ul> <li>conferencedCall</li> </ul>	D1C1					
	<ul> <li>extensions</li> </ul>	()					
Conference now	ConferencedEvent		ConferencedEvent		ConferencedEvent		During a Silent
established.	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D2C1	<ul> <li>primaryOldCall</li> </ul>	D3C1	SingleStepConference it is
	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	<ul> <li>secondaryOldCall</li> </ul>	$\sim$	ullet secondary Old Call	$\sim$	an implementation issue
	• confController	D1/NR	<ul> <li>confController</li> </ul>	DI	<ul> <li>confController</li> </ul>	D1	regarding the devices which
	<ul> <li>addedParty</li> </ul>	D1/NR	<ul> <li>addedParty</li> </ul>	DI	<ul> <li>addedParty</li> </ul>	D1	see the ConferencedEvent.
	<ul> <li>new Connection</li> </ul>	DICI	<ul> <li>newConnection</li> </ul>	DICI	<ul> <li>newConnection</li> </ul>	DICI	
	ullet deviceID	DI	ullet deviceID	DI	ullet deviceID	DI	
	<ul> <li>newConnection</li> </ul>	D2CI	<ul> <li>newConnection</li> </ul>	D2C1	<ul> <li>newConnection</li> </ul>	D2CI	
	ullet deviceID	D2	ullet deviceID	D2	ullet deviceID	D2	
	<ul> <li>oldConnection</li> </ul>	D2CI	<ul> <li>oldConnection</li> </ul>	D2CI	<ul> <li>oldConnection</li> </ul>	D2CI	
	<ul> <li>new Connection</li> </ul>	D3CI	<ul> <li>newConnection</li> </ul>	D3CI	<ul> <li>newConnection</li> </ul>	D3CI	
	ullet deviceID	D3	ullet deviceID	D3	ullet deviceID	D3	
	• oldConnection	D3CI	<ul> <li>oldConnection</li> </ul>	D3CI	<ul> <li>oldConnection</li> </ul>	D3CI	
	ullet local $Connection Info$	Connected	ullet local Connection Info	Connected	ullet local Connection Info	Connected	
	ullet correlator $D$ ata	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	
	• cause	silent	• cause	silent	• cause	silent	
		Monitor		Monitor		Monitor	

### **Divert Deflection Service**

10

A call is in the process of being established between device D2 and device D1. When device D1 detects the incoming call it invokes the divert deflection service redirecting the incoming call to a new destination, device D3.

AFTER D1 D3 BEFORE D1 D3

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
Divert deflection service	DivertCallRequest						The state of DICI may be
invoked on behalf of	<ul> <li>callToBeDiverted</li> </ul>	D1C1					connected, queued or
device DI.	<ul> <li>newDestination</li> </ul>	D3					alerting.
	ullet device $Profile$	$\sim$					
	<ul> <li>correlatorData</li> </ul>	$\sim$					
	<ul> <li>extensions</li> </ul>	()					
Acknowledgement.	DivertCallResult						
	• extensions	$\sim$					
Call to DI gets diverted.	DivertedEvent						
1	• connection	D1C1					
	<ul> <li>divertingDevice</li> </ul>	D1/NR					
	<ul> <li>newDestination</li> </ul>	D3					
	<ul> <li>localConnectionInfo</li> </ul>	Null					
	<ul> <li>correlatorData</li> </ul>	$\sim$					
	• cause	Redirected					
Diverted call then alerts			DeliveredEvent		DeliveredEvent		
device D3.			<ul> <li>connection</li> </ul>	D3C1	<ul> <li>connection</li> </ul>	D3C1	
			<ul> <li>alertingDevice</li> </ul>	D3	alertingDevice	D3/NR	
			<ul> <li>callingDevice</li> </ul>	D2/NR	<ul> <li>callingDevice</li> </ul>	D2	
			<ul> <li>calledDevice</li> </ul>	DI	<ul> <li>calledDevice</li> </ul>	D1	
			<ul> <li>lastRedirectionDevice</li> </ul>	DI	<ul> <li>IastRedirectionDevice</li> </ul>	D1	
			<ul> <li>originating Connection</li> </ul>	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\sim$	
			ullet local Connection Info	Connected	<ul> <li>localConnectionInfo</li> </ul>	Alerting	
			<ul> <li>correlatorData</li> </ul>	$\sim$	ullet correlator $D$ ata	$\sim$	
			• cause	Redirected	• cause	Redirected	

#### Call Completion Service

A call attempt between device D1 and device D2 results in a busy connection. The calling party, device D1, invokes the callback Call Completion service. The initial part of the scenario is as a Make Call which meets busy.

D1 +c (c3) +c D2 AFTER BEFORE

Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Commente
Activity	MONITONED DE VICE DI		MONTH ONED DEVICE D2		MONITONED DEVICE D3	Comments
Device D2 is busy.	FailedEvent		FailedEvent			
,	failedConnection	D2C1	<ul> <li>failedConnection</li> </ul>	D2C1		
	<ul> <li>failingDevice</li> </ul>	D2	<ul> <li>failingDevice</li> </ul>	D2/NR		
	<ul> <li>calledDevice</li> </ul>	D2	<ul> <li>calledDevice</li> </ul>	D2		
	$\bullet \ local Connection Info$	Connected	ullet local Connection Info	Failed		
	• correlatorData	$\sim$	• correlatorData	$\sim$		
	• cause	Busy	• cause	Busy		
The Call Completion	CallCompletionRequest					
service is invoked on	<ul> <li>callback</li> </ul>	D1C1				
behalf of device D1, the	• extensions	$\sim$				
calling party.						
Acknowledgement.	CallCompletionResult					
	<ul> <li>extensions</li> </ul>	$\Box$				
Device DI replaces the	ConnectionClearedEvent		ConnectionClearedEvent			
handset.	droppedConnection	D1C1	<ul> <li>droppedConnection</li> </ul>	DICI		
	<ul> <li>releasingDevice</li> </ul>	D1/NR	<ul> <li>releasing Device</li> </ul>	D1		
	<ul> <li>localConnectionInfo</li> </ul>	Null	<ul> <li>localConnectionInfo</li> </ul>	Failed		
	• correlatorData	$\sim$	• correlatorData	$\sim$		
	• cause	callBack	• cause	callBack		
Failed connection D2CI			ConnectionClearedEvent			
also clears.				D2C1		
				D2/NR		
			<ul> <li>localConnectionInfo</li> </ul>	Null		
			• correlatorData	$\sim$		
			• cause	normal		
				Clearing		
Device D2 sometime later			ConnectionClearedEvent			
clears from their active			<ul> <li>droppedConnection</li> </ul>	D2C2		
call C2.			<ul> <li>releasingDevice</li> </ul>	D2/NR		
			<ul> <li>localConnectionInfo</li> </ul>	Null		
			<ul> <li>correlatorData</li> </ul>	$\sim$		
			• cause	normal		
				Clearing		
Device DI prompted to lift	ServiceInitiatedEvent	D1C2				
nanaset Jor cattback.		Luitiatad				
	• tocate onnectioningo	Inilialea				
	• cause	callback				

Device DI lifts handset -	OriginatedEvent				
callback now proceeds.	<ul> <li>originatedConnection</li> </ul>	D1C3			
	<ul> <li>callingDevice</li> </ul>	D1/NR			
	• calledDevice	D2			
	<ul> <li>originatingDevice</li> </ul>	$\sim$			
	<ul> <li>localConnectionInfo</li> </ul>	Connected			
	• correlatorData	$\supset$			
	• cause	callBack			
Device D2 begins to ring.	DeliveredEvent		DeliveredEvent		
	• connection	D2C3	• connection	D2C3	
	<ul> <li>alertingDevice</li> </ul>	D2	<ul> <li>alertingDevice</li> </ul>	D2/NR	
	• callingDevice	DI	<ul> <li>callingDevice</li> </ul>	DI	
	• calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2	
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	<ul> <li>originating Connection</li> </ul>	$\sim$	<ul> <li>originatingConnection</li> </ul>	$\Box$	
	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Alerting	
	<ul> <li>correlatorData</li> </ul>	$\Box$	ullet correlator $Data$	$\sim$	
	• cause	callBack	• cause	callBack	
D2 connects to the call.	EstablishedEvent		EstablishedEvent		
	<ul> <li>establishedConnection</li> </ul>	D2C3	<ul> <li>establishedConnection</li> </ul>	D2C3	
	<ul> <li>answeringDevice</li> </ul>	D2	<ul> <li>answeringDevice</li> </ul>	D2/NR	
	<ul> <li>callingDevice</li> </ul>	D1	<ul> <li>callingDevice</li> </ul>	DI	
	• calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2	
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>IastRedirectionDevice</li> </ul>	NR	
	<ul> <li>originating Connection</li> </ul>	$\Box$	<ul> <li>originatingConnection</li> </ul>	$\Box$	
	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Connected	
	• correlatorData	$\sim$	ullet correlator $D$ ata	$\Box$	
	• cause	callBack	• cause	callBack	

## 12 A Manually Invoked Call Park

In this scenario, D1 and D2 are in an established call. Device D1 decides to put the call in park (illustrated by device D3). This operation can be done both manually or by using a park button on a feature phone.

BEFORE D1 - c - C1 - c - D2 D1 - c - D2 D3 D3

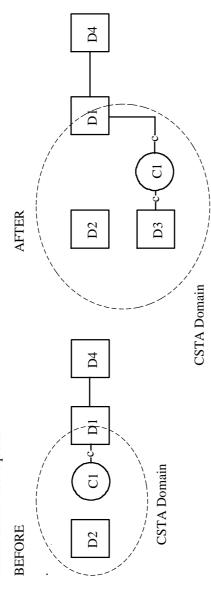
Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3		Comments
(ar. mar.							
A Call Park is initiated by	HeldEvent		HeldEvent				If the phone has a Call Park
device D1. Call to device	<ul> <li>heldConnection</li> </ul>	DICI	<ul> <li>heldConnection</li> </ul>	DICI			feature button then this event
D2 is placed on hold.	<ul> <li>holdingDevice</li> </ul>	D1/NR	<ul> <li>holdingDevice</li> </ul>	DI			can be omitted.
	ullet local Connection Info	Hold	<ul> <li>localConnectionInfo</li> </ul>	Connected			
	ullet correlator $D$ ata	$\Box$	ullet correlator $D$ ata	$\sim$			
	• cause	()	• cause	()			
Indication that device DI	ServiceInitiatedEvent						
has invoked a service.	<ul> <li>initiatedConnection</li> </ul>	D1C2					
	<ul> <li>localConnectionInfo</li> </ul>	Initiated					
	• cause	newCall					
Call is queued at device	QueuedEvent		QueuedEvent		QueuedEvent		
D3.	<ul> <li>queuedConnection</li> </ul>	D3C1	<ul> <li>queuedConnection</li> </ul>	D3C1	<ul> <li>queuedConnection</li> </ul>	D3C1	
	• dnene	D3	• dnene	D3	• dnene	D3/NR	
	<ul> <li>callingDevice</li> </ul>	D2	calling Device	D2	callingDevice	D2	
	• calledDevice	D3	• called Device	D3	calledDevice	D3	
	<ul> <li>IastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	ullet numbered $Q$ ueued	$\sim$	ullet numbered $Q$ ueued	$\sim$	ullet numbered $Q$ ueued	$\sim$	
	ullet calls $InFront$	$\sim$	• callsInFront	$\sim$	• callsInFront	$\sim$	
	ullet local Connection Info	Connected	ullet local Connection Info	Connected	<ul> <li>localConnectionInfo</li> </ul>	Queued	
	ullet correlator $Data$	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	<ul> <li>correlatorData</li> </ul>	$\sim$	
	• cause	Park	• cause	Park	• cause	Park	
Device D2 is connected to	ConnectionClearedEvent		ConnectionClearedEvent				
the Park device.	<ul> <li>droppedConnection</li> </ul>	D1C1	<ul> <li>droppedConnection</li> </ul>	D1C1			
	<ul> <li>releasingDevice</li> </ul>	D1/NR	<ul> <li>releasingDevice</li> </ul>	DI			
	ullet local Connection Info	Null	<ul> <li>localConnectionInfo</li> </ul>	Connected			
	ullet correlator $D$ ata	$\Box$	ullet correlator $D$ ata	$\sim$			
	• cause	Park	• cause	Park			
Device D1 replaces	ConnectionClearedEvent						
handset and the feature	<ul> <li>droppedConnection</li> </ul>	D1C2					
invocation call C2, clears.	<ul> <li>releasingDevice</li> </ul>	DI					
	<ul> <li>localConnectionInfo</li> </ul>	Null					
	• correlatorData	$\sim$					
	• cause	normal					
		Clearing					

#### 13 Route Services

This section of scenarios includes examples where the switch asks the application how to route a call. Routeing at a particular device is enabled by using the Set Feature service as shown below. Only the routeing dialogue is shown - not all the associated call events.

### 13.1 Route Request Service

Switch invokes the Route Request service for an emergency call of high priority to which a reply is received and accepted by the switch. All the following entries are service requests.

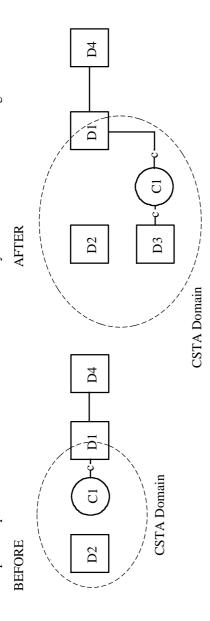


Activity	ROUTEING DEVICE DI		ROUTEING DEVICE D2	ROUTEING DEVICE D3	Comments
The Set Feature service is SetFeatureRequest	SetFeatureRequest				In this example device DI
invoked to enable routeing • device	• device	D1			represents a trunk.
on behalf of device D1.		enable			
		Routing			
	ullet device $Profile$	$\sim$			
	• extensions	$\sim$			
Acknowledgement.	SetFeatureResult				
	<ul> <li>extensions</li> </ul>	()			
Switch initiates a	RouteRequest				
RouteRequest when a call • crossRefIdentifier	<ul> <li>crossRefIdentifier</li> </ul>	Ref1			
involves device D1.	<ul> <li>currentRoute</li> </ul>	D2			
	ullet calling $Device$	D4			
	ullet routing $Device$	D1/NR			
	• routedCall	DICI			
	ullet route $SelAlgorithm$	emergency			
	• priority	TRUE			
	ullet device $Profile$	$\sim$			
	<ul> <li>correlatorData</li> </ul>	$\sim$			
	<ul> <li>extensions</li> </ul>	()			

Computer responds with	RouteSelectRequest			
an alternative route.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1		
	• routeSelected	D3		
	• remainRetry	$\sim$		
	ullet device $Profile$	$\sim$		
	• route UsedReq	$\sim$		
	<ul> <li>correlatorData</li> </ul>	$\sim$		
	<ul> <li>extensions</li> </ul>	()		
Switch ends re-route	RouteEndRequest			
session.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1		
	$\bullet$ error $Value$	$\sim$		

#### 13.2 Route Used Service

Switch invokes the Route Request service for an emergency call of high priority to which a reply is received and accepted by the switch. In this case the computer requests that it is informed of the final route used by the switch. All the following entries are Service Requests. Routeing is already enabled.



Switch invokes the RouteRequest RouteRequest service when a call involves - currentBoute D4 - currentBoute D4 - currentBovice D1/NR - routedCall - routedCall - routedCall - routeSchAlgorithm emergency - priority - priority - trouteSchAlgorithm emergency - priority - trouteSchAlgorithm emergency - priority - trouteSchAlgorithm () - correlatorData () - correlatorData () - extensions  Computer responds with - RouteSchedentifier Ref1 - crossRefictentifier () - trouteSchedentifier () - trouteSchedentifier () - trouteUsedRequest () - crossRefictentifier () - crossReficte	Activity	ROUTEING DEVICE D1		ROUTEING DEVICE D2	ROUTEING DEVICE D3	Comments
curentRoute callingDevice routingDevice routingDevice routeSelAlgorithm priority deviceProfile correlatorData extensions ith RouteSelectRequest crossRefidentifier routeSelected remainRetry deviceProfile routeUsedRequest correlatorData extensions ter RouteUsedRequest crossRefidentifier routeUsedRequest correlatorData extensions ter RouteUsed callingDevice domain correlatorData	Switch invokes the	RouteRequest				In this example device DI
calling Device calling Device routing Device routed Call routeSelAlgorithm priority device Profile correlator Data extensions ith Route Select Request cross Refidentifier route Select de remain Retry device Profile route Used Req correlator Data extensions ter Route Used Request cross Refidentifier route Used RouteUsed calling Device domain correlator Data	RouteRequest service	tifier	Ref1			represents a trunk.
calling Device routing Device routed Call routed Call routed Selected device Profile correlator Data extensions with Route Selected demain	when a call involves		22			
• routingDevice • routedCall • routeSelAlgorithm • priority • deviceProfile • corelatorData • extensions with RouteSelectRequest • crossRefidentifier • routeSelected • remainRetry • deviceProfile • routeUsedRequest • correlatorData • extensions uter RouteUsedRequest • correlatorData • correlatorData • correlatorData • correlatorData • correlatorData	levice DI.		04			
• routedCall • routeSelAlgorithm • priority • deviceProfile • correlatorData • extensions with RouteSelectRequest • crossRefidentifier • routeSelected • remainRery • deviceProfile • routeUsedReq • correlatorData • extensions uter RouteUsedRequest • crossRefidentifier • routeUsedRequest • crossRefidentifier • routeUsed • callingDevice • domain • correlatorData		ice	OI/NR			
• routeSelAlgorithm • priority • deviceProfile • correlatorData • extensions • crossRefidentifier • routeSelectRequest • remainRery • deviceProfile • routeUsedReq • correlatorData • extensions uter RouteUsedRequest • crossRefidentifier • routeUsed • callingDevice • domain • correlatorData			DICI			
• priority • deviceProfile • correlatorData • correlatorData • extensions • crossRefidentifier • routeSelected • remainRery • deviceProfile • routeUsedReq • correlatorData • extensions uter RouteUsedRequest • crossRefidentifier • routeUsed • callingDevice • domain • correlatorData			mergency			
<ul> <li>deviceProfile</li> <li>correlatorData</li> <li>extensions</li> <li>crossRefidentifier</li> <li>crossRefidentifier</li> <li>routeSelected</li> <li>remainRetry</li> <li>deviceProfile</li> <li>routeUsedReaq</li> <li>correlatorData</li> <li>extensions</li> <li>uter</li> <li>RouteUsedRequest</li> <li>correlatorData</li> <li>crossRefidentifier</li> <li>routeUsed</li> <li>domain</li> <li>correlatorData</li> </ul>			TRUE			
• correlatorData • extensions RouteSelectRequest • crossReffdentifier • routeSelected • remainRerry • deviceProfile • routeUsedReq • correlatorData • extensions uter RouteUsedRequest • crossReffdentifier • routeUsed • callingDevice • domain • correlatorData		• deviceProfile (				
with RouteSelectRequest crossReffdentifier routeSelected remainRerry deviceProfile routeUsedReq correlatorData extensions uter RouteUsedRequest crossReffdentifier routeUsed callingDevice domain correlatorData		• correlatorData (				
with RouteSelectRequest  • crossRefIdentifier • routeSelected • remainRetry • deviceProfile • routeUsedReq • correlatorData • extensions uter RouteUsedRequest • crossRefIdentifier • routeUsed • callingDevice • domain • correlatorData		• extensions (	( )			
crossRefidentifier routeSelected remainRetry deviceProfile routeUsedReq correlatorData extensions uter RouteUsedRequest crossRefidentifier routeUsed callingDevice domain correlatorData	Computer responds with	RouteSelectRequest				
• routeSelected • remainRetry • deviceProfile • routeUsedReq • correlatorData • extensions RouteUsedRequest • crossRetIdentifier • routeUsed • callingDevice • domain • correlatorData	in alternative route.		Ref1			
• remainRetry • deviceProfile • routeUsedReq • correlatorData • extensions RouteUsedRequest • crossRefldentifier • routeUsed • callingDevice • domain • correlatorData			53			
• deviceProfile • routeUsedReq • correlatorData • extensions RouteUsedRequest • crossRefidentifier • routeUsed • callingDevice • domain • correlatorData						
• routeUsedReq • correlatorData • extensions RouteUsedRequest • crossRefIdentifier • routeUsed • callingDevice • domain • correlatorData			(			
• correlatorData • extensions RouteUsedRequest • crossRefIdentifier • routeUsed • callingDevice • domain • correlatorData			TRUE			
extensions RouteUsedRequest crossRefIdentifier routeUsed callingDevice domain correlatorData		• correlatorData (				
RouteUsedRequest • crossRefIdentifier • routeUsed • callingDevice • domain • correlatorData		• extensions (	( )			
• crossRefidentifier • routeUsed • callingDevice • domain • correlatorData	Switch notifies computer	RouteUsedRequest				The route used could be
• routeUsed • callingDevice • domain • correlatorData	of the route used.		Refi			different if the switch decides
• callingDevice () • domain () • correlatorData ()			)3			to ignore the route provided
• domain () • correlatorData ()		ullet calling $Device$	(			by the computer.
• correlatorData ()		• domain				
		• correlatorData (				
• extensions ()		• extensions (	)			

Switch ends re-route	RouteEndRequest			
session.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1		
	ullet $error Value$	$\sim$		
	<ul> <li>extensions</li> </ul>	()		

#### Re-Route Service 13.3

Switch invokes the Route Request service for an emergency call of high priority to which a reply is received but the switch requests an alternative. The computer requests that it is informed of the final route used by the switch. All the entries below are Service Requests. Routeing is already enabled.

7 DI AFTER D5 **D**3 D2 CSTA Domain D4 CSTA Domain BEFORE D2

Activity	ROUTEING DEVICE D1		ROUTEING DEVICE D2	ROUTEING DEVICE D3	Comments
Switch initiates	RouteRequest				
RouteRequest.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1			
	• currentRoute	D2			
	• calling Device	D4			
	• routingDevice	D1/NR			
	• routedCall	DICI			
	• routeSelAlgorithm	emergency			
	• priority	TRUE			
	• deviceProfile	$\sim$			
	• correlatorData	$\sim$			
	• extensions	$\sim$			
Computer responds with	RouteSelectRequest				
an alternative route.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1			
	• routeSelected	D3			
	• remainRetry	$\sim$			
	• deviceProfile	$\sim$			
	• route UsedReq	TRUE			
	• correlatorData	$\sim$			
	• extensions	()			
Switch asks for an	Re-routeRequest				
alternative.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1			
	• extensions	$\sim$			

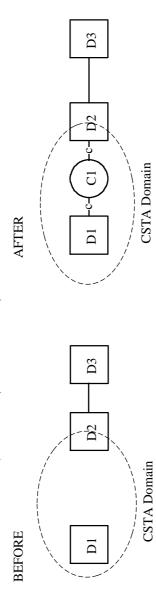
Computer responds with	RouteSelectRequest	-		
	• crossRefIdentifier	Ref1		
	• routeSelected	D5		
	• remainRetry			
	ullet device $Profile$			
	• route UsedReq	TRUE		
	<ul> <li>correlatorData</li> </ul>			
	<ul> <li>extensions</li> </ul>	()		
Switch notifies computer	RouteUsedRequest			The route used could be
of the route used.	• crossRefIdentifier	Ref1		different if the switch decides
	• routeUsed	D5		to ignore the route provided
	ullet calling $Device$			by the computer.
	• domain	$\sim$		
	<ul> <li>correlatorData</li> </ul>			
	<ul> <li>extensions</li> </ul>	()		
Switch ends re-route	RouteEndRequest			
session.	<ul> <li>crossRefIdentifier</li> </ul>	Ref1		
	ullet $error Value$			
	• extensions	()		

#### 14 Incoming Calls

This section includes an example of an incoming call from outside of the CSTA domain handled by a trunk within the CSTA domain and an example of an incoming call to an ACD queue.

### 14.1 Successful incoming Call

In this scenario an incoming trunk, device D2 (the originating device), is initiating a call on behalf of device D3 (the calling device) to a destination within the CSTA sub-domain, device D1 (the called device).

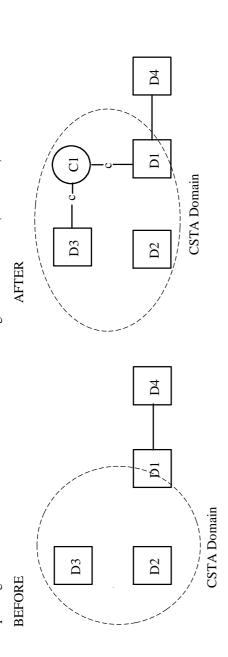


Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Device D2 goes "off-			ServiceInitiatedEvent			This event can be generated
hook".			<ul> <li>initiatedConnection</li> </ul>	D2C1		for all types of trunks
			ullet local Connection Info	Initiated		excluding en-bloc dialling
			• cause	newCall		(i.e by an ISDN trunk).
Device D2 completes			OriginatedEvent			This event can be generated
dialling.			<ul> <li>originatedConnection</li> </ul>	D2C1		for ISDN en-bloc dialling.
			<ul> <li>callingDevice</li> </ul>	D3/NK		Information about D3 may
			<ul> <li>calledDevice</li> </ul>	DI		exist in some types of trunks
			ullet originating $Device$	D2		like ISDN or TI with support
			$\bullet \ local Connection Info$	Connected		of ANI.
			<ul> <li>correlatorData</li> </ul>	$\sim$		
			• cause	newCall		
Device DI begins to ring	DeliveredEvent		DeliveredEvent			Information about D3 may
and D2/D3 listens to		D1C1	<ul> <li>connection</li> </ul>	DICI		exist in some types of trunks
ringing tone.	•	D1/NR	<ul> <li>alertingDevice</li> </ul>	DI		like ISDN or TI with support
		D3/NK	<ul> <li>callingDevice</li> </ul>	D3/NK		of ANI.
		D1	<ul> <li>calledDevice</li> </ul>	DI		
	1Device	Y.R.	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
		D2C1	<ul> <li>originating Connection</li> </ul>	D2CI		
	localConnectionInfo	Alerting	$\bullet \ local Connection Info$	Connected		
	• correlatorData (		<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	newCall	• cause	newCall		

Device DI answers the EstablishedEvent	EstablishedEvent		EstablishedEvent		Information about D3 may
call.	<ul> <li>establishedConnection</li> </ul>	D1C1	<ul> <li>establishedConnection</li> </ul>	DICI	exist in some types of trunks
	<ul> <li>answeringDevice</li> </ul>	D1/NR	<ul> <li>answeringDevice</li> </ul>	D1	like ISDN or TI.
	<ul> <li>callingDevice</li> </ul>	D3/NK	<ul> <li>callingDevice</li> </ul>	D3/NK	
	• calledDevice	DI	<ul> <li>calledDevice</li> </ul>	D1	
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR	
	<ul> <li>originating Connection</li> </ul>	D2CI	<ul> <li>originating Connection</li> </ul>	D2CI	
	<ul> <li>localConnectionInfo</li> </ul>	Connected	ullet local Connection Info	Connected	
	<ul> <li>correlatorData</li> </ul>	$\Box$	<ul> <li>correlatorData</li> </ul>		
	• cause	newCall	• cause	newCall	

## 14.2 Incoming call to ACD with no available agents

Incoming call to an ACD distribution mechanism (device D2) where the call awaits distribution to a free agent. While queuing, the trunk (device D1) may be connected with a recorded announcement (events not shown). The recording may repeat itself or change to a different one, several times while the call is queuing at the ACD distribution mechanism. When an agent becomes free (device D3) the call is delivered to that device.



Activity	MONITORED DEVICE D1		MONITORED DEVICE D2		MONITORED DEVICE D3	Comments
Device DI goes "off-	ServiceInitiatedEvent					
hook".	initiatedConnection	D1C1				
	• localConnectionInfo	Initiated				
	• cause	newCall				
Device D1 completes	OriginatedEvent					
dialling.	originatedConnection	DICI				
	callingDevice	D4/NK				
	calledDevice	D2				
	<ul> <li>originatingDevice</li> </ul>	DI/NR				
	• localConnectionInfo	Connected				
	• correlatorData	$\sim$				
	• cause	newCall				
The call begins to alert at	DeliveredEvent		DeliveredEvent			Information about D4 may
the queue.	• connection	D2C1	<ul> <li>connection</li> </ul>	D2C1		exist in some types of trunks
	<ul> <li>alertingDevice</li> </ul>	D2	<ul> <li>alertingDevice</li> </ul>	D2/NR		like ISDN or TI with support
	<ul> <li>callingDevice</li> </ul>	D4	<ul> <li>callingDevice</li> </ul>	D4		of ANI.
	• calledDevice	D2	<ul> <li>calledDevice</li> </ul>	D2		
	<ul> <li>lastRedirectionDevice</li> </ul>	NR	<ul> <li>lastRedirectionDevice</li> </ul>	NR		
	<ul> <li>originatingConnection</li> </ul>	DICI	<ul> <li>originating Connection</li> </ul>	DICI		
	• localConnectionInfo	Connected	ullet local Connection Info	Alerting		
	• correlatorData	$\supset$	<ul> <li>correlatorData</li> </ul>	$\sim$		
	• cause	entering	• cause	entering		
		Distribution		Distribution		

Call is queued in Q1 represented by device D2.	QueuedEvent queuedConnection queue callingDevice calledDevice numberedQueued callshFron callshFron localConnectionInfo correlatorData cause	D2C1 Q1 Q1 D4 D2 NR () () Connected () noAvailable Agents	QueuedEvent • queuedConnection • queue • callingDevice • calledDevice • lastRedirectionDevice • numberedQueued • callshFront • localConnectionInfo • correlatorData • cause	D2C1 Q1 Q1 D4 D2 NR () () Queued () noAvailable Agents			QI may identify an ACD group or the ACD queuing mechanism. A call may reside within multiple queues.
The queued call gets diverted to device D3. D3 represents an agent which is now free.			DivertedEvent  connection  divertingDevice  newDestination  localConnectionInfo  correlatorData  cause	D2C1 D2/NR D3 Null ()			An annoucement can be provided prior to this event for which events may be generated.
Device D3 is alerted.	DeliveredEvent  connection  alertingDevice  callingDevice  callingDevice  lastRedirectionDevice  originatingConnection  correlatorData  canse	D3C1 D3 D4 D4 D2 D2 D2 Connected () Distributed			DeliveredEvent  connection  alertingDevice  callingDevice  icalledDevice  lastRedirectionDevice  originatingConnection  localConnection  correlatorData  cause	D3C1 D3/NR D3/NR D4 D2 D2 D1C1 Alerting ()	
Device D3 answers the call.	EstablishedEvent • establishedConnection • answeringDevice • callingDevice • calledDevice • lastRedirectionDevice • originatingConnection • localConnectionInfo • correlatorData • cause	D3C1 D3 D4 D2 D2 D2 D1C1 Connected () Distributed			EstablishedEvent  e establishedConnection  answeringDevice  callingDevice  calledDevice  lastRedirectionDevice  originatingConnection  localConnectionInfo  correlatorData  cause	D3C1 D3NR D4 D2 D2 D2 D1C1 Connected ()	

This ECMA Technical Report TR/68 is available free of charge from:

ECMA 114 Rue du Rhône CH-1204 Geneva Switzerland

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

This Technical Report can also be downloaded as file T068-DOC.EXE or file T068-PSC.EXE from ECMANEWS