INTERNATIONAL TELECOMMUNICATION UNION

ITU-T Q.1302

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(10/95)

INTELLIGENT NETWORK

TELECOMMUNICATION APPLICATIONS FOR SWITCHES AND COMPUTERS (TASC) – TASC FUNCTIONAL SERVICES

ITU-T Recommendation Q.1302

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation Q.1302 was prepared by ITU-T Study Group 11 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 17th of October 1995.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

Page

1 Scope 1

2 References 1

3 Terms and definitions 1

4 Abbreviations 1

5 Generic Functional Service requirements 2

5.1 Templates 2

5.1.1 Functional Service template 2

5.1.2 Event template 4

5.2 Error definitions 4

6 Functional Service descriptions 6

6.1 General Functional Services 6

6.1.1 Functional Service name: Answer Call 6

6.1.2 Functional Service name: Call Completion 8

6.1.3 Functional Service name: Clear Call 10

6.1.4 Functional Service name: Conference Call 11

6.1.5 Functional Service name: Divert Call 14

6.1.6 Functional Service name: Drop CP 16

6.1.7 Functional Service name: Hold Call 17

6.1.8 Functional Service name: Make Call 19

6.1.9 Functional Service name: Retrieve Call 21

6.1.10 Functional Service name: Transfer Call Service 22

6.2 Specialized Functional Services 25

6.2.1 Functional Service name: Alternate Call 25

6.2.2 Functional Service name: Consultation Call 26

6.2.3 Functional Service name: Reconnect Call 29

6.3 Manipulate Functional Services 31

6.3.1 Manipulate Feature Functional Services 31

6.3.1.1 Functional Service name: Manipulate Feature 31

6.3.1.2 Functional Service name: Query Feature 33

6.3.2 Manipulate Agent Functional Services 36

6.3.2.1 Functional Service name: Manipulate Agent 36

6.3.2.2 Functional Service name: Query Agent 37

6.4 Routing Functional Services 39

6.4.1 Call Related Routing 40

6.4.1.1 Functional Service name: Set routing 40

6.4.1.2 Functional Service name: Route Call 43

6.4.1.3 Functional Service name: Route Call Selected 45

6.4.1.4 Functional Service name: Route Used 47

6.4.2 Non-Call Related Routing 49

6.4.2.1 Functional Service name: Route Information 50

6.4.2.2 Functional Service name: Route Information Selected 51

6.4.2.3 Functional Service name: Re-Route Information 53

6.4.2.4 Functional Service name: Route Information End 55

Page

6.5 Monitor Functional Services 56
6.5.1 Functional Service name: Change Monitor Filter Service 57
6.5.2 Functional Service name: Monitor Start Service 59
6.5.3 Functional Service Name: Monitor Stop Service 61
6.6 Snapshot Functional Services 63
6.6.1 Functional Service name: Snapshot Call 63
6.6.2 Functional Service name: Snapshot CE 65
6.7 Functional Service name: Event Report 67
6.8 Application Status Services 68
6.8.1 Functional Service name: Application Activity Check 68
6.8.2 Functional Service name: Application Congestion Report 69
6.9 Functional Service name: Send Private Data 70
7 Event descriptions 71
7.1 Agent events 71
7.1.1 Event name: Agent Busy 71
7.1.2 Event name: Agent Logged Off 72
7.1.3 Event name: Agent Logged On 73
7.1.4 Event name: Agent Not Ready 73
7.1.5 Event name: Agent Ready 74
7.1.6 Event name: Agent Working After Call 75
7.2 Call progress events 76
7.2.1 Event name: Call Arrived 76
7.2.2 Event name: Call Cleared 77
7.2.3 Event name: Call Conferenced 77
7.2.4 Event name: Call Delivered 78
7.2.5 Event name: Call Diverted 79
7.2.6 Event name: Call Established 80
7.2.7 Event name: Call Failed 80
7.2.8 Event name: Call Held 81
7.2.9 Event name: Call Originated 82
7.2.10 Event name: Call Received 83
7.2.11 Event name: Call Retrieved 83
7.2.12 Event name: Call Transferred 84
7.2.13 Event name: CP Dropped 85
7.2.14 Event name: Service Pending 86
7.3 Cause List 87

SUMMARY

This Recommendation identifies the Functional Services that comprises TASC (Telecommunications Applications for Switches and Computers) and is one in the Q.1300-Series on TASC. This Recommendation identifies the TASC Functional Services which provide building blocks for application developers. The main purpose of TASC is to allow applications to be developed which integrate the services provided by both computing and telecommunication platforms. This would typically allow business applications to use TASC to integrate the computer workstation and telephone at the user's desktop. This Recommendation does not define how the Functional Services should be implemented. These Functional Services can be used to support applications that integrate the use of telecommunication and computing platforms.

INTRODUCTION

The definition of individual Functional Services is based on a model described in the TASC Architecture Recommendation and goes down to the level of parameters, not protocol details. It is assumed that the reader has read the TASC General Overview Recommendation Q.1300 and the TASC Architecture Q.1301 Recommendations.

The Functional Services are offered by either switch or computer. Events reports are generated on changes of states. A common template is used to describe the Functional Services and is structured into Relationship, General Description, Parameters, Client/Server procedures and Management Requirements.

BACKGROUND

This Recommendation was developed at the same time as the Architecture and Management Requirements were developed for TASC. It is based on the experience of ECMA (Standardizing Information and Communication Systems) and ANSI (American National Standards Institute) member companies in developing switch-to-computer interfaces and takes direction from CSTA (Computer Supported Telecommunications Applications) and SCAI (Switch-to-Computer Applications Interface) standards.

KEYWORDS

Events, Functions, Services, TASC.

Recommendation Q.1302

TELECOMMUNICATION APPLICATIONS FOR SWITCHES AND COMPUTERS (TASC) – TASC FUNCTIONAL SERVICES

(Geneva, 1995)

1 Scope

This Recommendation defines a set of Functional Services for the area of Telecommunications Applications of Switches and Computers, that allow functional integration between a computing network and a telecommunications network.

The emphasis of TASC has been to define third-party call control functions which also encompasses first-party call control. TASC is independent of any underlying mechanism and is applicable to public, private and hybrid networks. TASC is designed to be flexible in order to support other communication environments in addition to ISDN and Intelligent Networks (IN).

It is focussed on providing an application service interface between a switch and a computer.

TASC supports both a single-ended (originating and terminating) view as well as a global call view.

Computing platforms (e.g. Application Programming Interfaces – APIs) that may support such functionally-integrated applications are outside the scope of this Recommendation.

2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation, or Annexes to this Recommendation. At the time of adoption of this Recommendation, the reference editions indicated were valid. Recalling that all Recommendations and other material incorporated by reference herein are subject to future revision, all users of this Recommendation are therefore advised that changes in the reference text that constitute future decisions of the work of Organizations or Study Groups other than ITU-T Study Group 11, do not automatically apply as amended provisions to the Recommendation.

- ITU-T Q.1300 (1995), *Telecommunication Applications for Switches and Computers (TASC) – TASC general overview.*
- ITU-T Q.1301 (1995), *Telecommunication Applications for Switches and Computers (TASC) – TASC architecture.*
- ITU-T Q.1303 (1995), *Telecommunication Applications for Switches and Computers (TASC) – TASC management: architecture, methodology and requirements.*

3 Terms and definitions

This Recommendation uses terms defined in Recommendation Q.1300.

4 Abbreviations

For the purposes of this Recommendation the following abbreviations are used:

CE	Communication Entity
CP	Communication Party
CV	Call View
FS	Functional Service
HLCp	Higher Layer Capability
ISDN	Integrated Services Digital Network

LLCp Lower Layer Capability

TASC Telecommunication Applications for Switches and Computers

TE Terminal Equipment

5 Generic Functional Service requirements

The Functional Service provided by a server to a client consists of observing and acting upon objects which can be accessed by the server on behalf of the client. In the TASC context, both the switch and the computer may take the role of the client or the server.

A Functional Service is used by a client to request a server to perform a specified function, and is used by the server to reply to the request. If the request for a Functional Service is refused, the server shall send diagnostic information indicating the reason that the Functional Service was refused.

For Functional Services requesting server activity from a switch, the following generic mechanisms, in principle, decouples from the specifics of the switching activity.

1) The precise moment at which the response is generated in relation to the switching activity is implementation and service dependent:

   – Some implementations may generate the response after checking the correctness of the request and before completing the requested service.

   – Other implementations may delay the response until the requested service has completed or is guaranteed to complete. In this case, a failure of the requested service is reflected in the response.

2) If a response is sent before the action requested by the Functional Service is completed, i.e. the response only indicates acceptance of the request, monitoring mechanisms could be used to keep track of the subsequent server activity.

Implementations based on TASC Functional Service descriptions need to indicate which of the above is implemented.

Monitoring issues

– If it is desirable for the computer application to know the actions the switch has performed, monitoring is desirable, i.e. the computer may start the Monitoring Functional Service and receive call related event reports from the switch, as defined in the filter of the Monitoring Functional Service. Alternatively an application may use the Snapshot Functional Services to determine the actual states.

– If the success response indicates that the requested Functional Service has been completed, monitoring may not be necessary.

The Functional Service description identifies the conditions of the client and the server and what actions they may perform.

5.1 Templates

5.1.1 Functional Service template

The Functional Service descriptions that are provided in clause 6 follow the service template as given below:

Functional Service name

a) Relationship

The involved entities and their relationship shall be indicated. As an example, if the request for a Functional Service is sent by the computing entity (client) to the switching entity (server), the relationship is indicated as follows:

Computer → Switch

b) General description

The general description sections are provided as normative text, and should provide a general overview of the actions.

c) Parameters

These sections provide normative text for describing the parameters and the optionality of the parameters associated with each request for a Functional Service.

If as a result of performing a Functional Service request, new objects of interest are created (e.g. CVs resulting from a Make Call), then references to these objects shall be provided in mandatory parameters in the success response.

Absence of a mandatory parameter will cause a reject of the Functional Service request. A present, but invalid mandatory parameter will result in an error response. Absence of an optional parameter will result in the usage of a default. Details are provided in the Operational Rules. Absence of "private data" will not affect the performing of a Functional Service.

REQUEST:

The request indicates the operation to be performed and shall include the parameters needed for the server to perform the operation. It shall allow inclusion of non-standardized or private data. The absence of "private data" shall not affect the performing of the Functional Service.

RESPONSE:

The response confirms the request either positively (success) or negatively (error).

Success:

A success response indicates receipt of the correctly formatted message, and may be generated after the requested Functional Service has been completed or is guaranteed to complete.

Error:

The error response always conveys the information that the request was received, but not performed. An error value indicates the server's best evaluation of the condition that was encountered that caused the server to send an error response to the request.

Parameter description

All parameters listed are described to indicate their usage.

d) Client (computer/switch)

The exact description of the procedures (normal or in error cases) as performed in the invoking/requesting entity (client) is provided. This is informative.

- **normal procedure**

Preconditions:

The conditions at the point at which the request is initiated are given (entry conditions). The order is considered to be a logical sequence.

(1)

(..)

(x) Allowable basic call view states.

Postconditions:

The conditions/actions after the request has been sent shall be given (exit conditions). The order is considered to be a logical sequence.

(1)

(..)

(x) Resulting basic call view states.

Operational rules

This section provides detailed information of the actions performed from the client's perspective. This includes handling of optional parameters in case they are present or not present.

- **error handling**

Specific error handling is to be described herein.

e) Server (switch/computer)

The exact description of the procedures (normal or in error cases) as performed in the responding entity (server) is provided. This is informative.

- **normal procedure**

Preconditions:

The conditions at the point at which the requested operation could be performed are given (entry conditions). The order is considered to be a logical sequence.

(1)

(..)

(x) Allowable basic call view states.

Postconditions:

The conditions/actions after the request has been understood shall be given (exit conditions). The order is considered to be a logical sequence.

(1)

(..)

(x) Resulting basic call view states.

Operational rules

This section provides detailed information of the actions performed from the server's perspective. This includes handling of optional parameters in case they are present or not present.

If monitoring is enabled, events arising from normal handling of the request will be listed.

- **error handling**

Specific error handling is to be described herein.

f) Management requirements

General requirements that are considered useful for the support of the Functional Service will be described; no order is intended.

5.1.2 Event template

For event reporting, which does not require a response to be sent, the Functional Service Template is adapted accordingly.

5.2 Error definitions

The error response always conveys that the request was received, but not completed. The definitions apply equally to both Services requested by a TASC Computing Function and those requested by a TASC Switching Function. It shall provide the reason why the server did not complete the function. The specific error values are organized into error groups as follows:

– FS specific errors	indicate that the error is specific to the FS for which it is defined.
– Request errors	indicate that there is an error in the FS request.
– State incompatibility errors	indicate that the FS request was not compatible with the condition of a related TASC entity.

– System resource availability errors — indicate that the FS request cannot be completed because of a lack of system resources.

– Subscribed resource availability errors — indicate that the FS request cannot be completed because a required resource must be purchased or contracted by the client system.

– Performance errors — indicate that an error has been returned as a result of system performance problems.

– Unspecified errors — indicate that an error has occurred that is not among the other error types.

– Private errors — indicate that an error has occurred which is private to the TASC application. The type of error is not part of the TASC specification.

– TASC private data information errors — indicate that an error has occurred in the TASC private data information of the FS request. The type of error is not part of the TASC specification.

An error value indicates the server's best evaluation of the condition that was encountered that caused the server to send a negative response to the Service Request. Within each error group, specific errors may be defined. Errors from the FS Specific Errors group must be chosen if the error is appropriate. A parameter value in error and additional error information may also be returned as part of the error. Thus, the returned error for TASC provides the following information:

errorGroupNumber	ErrorGroups	
errorNumber	TASCErrors	OPTIONAL
parameterValueinError	ErroredValue	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

errorGroupNumber	identifies the group of errors to which the error report belongs.
errorNumber	may specify a specific error within the error group.
parameterValueinError	may provide the parameter value which initiated the error (the identity of the parameter is implied by the errorNumber).
privateData	may identify custom error data specific to an implementation or application.

Key to TASC errors

TASC errors generally comprise of an adjective followed by the type of entity involved, e.g. unknownCalledCE.

Error adjectives:

– unknown — the switch or computer has no knowledge of the entity referenced. This includes values out of range and values which do not correspond to known entities.

– invalid — the switch or computer recognizes the entity referenced, but the entity is not valid in the requested context or existing scenario.

– incompatible — is a more specific form of Invalid where the identified entity does not match the characteristics required of it.

– unavailable — where the entity is not currently available for the use required.

– ambiguous the switch or computer is unable to uniquely select an entity from the information provided.

– occupied the entity is fully occupied and unable to comply with the request.

– unauthorized the entity is not authorized to comply with the request.

The most specific error defined for the FS which is relevant to the error situation should be reported in the error response.

6 Functional Service descriptions

6.1 General Functional Services

General Functional Services consist of functions concerned with general call handling.

6.1.1 **Functional Service name:** Answer Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to establish an incoming call to a Line CE.

Note that normally, these CEs are able to accept a call without manually going off-hook (e.g. operator head-set, speakerphone capability).

c) Parameters

REQUEST:

terminatingLineCE	CEID	
terminatingCall	CVID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

terminatingLineCE identifies the Line CE on whose behalf the switch shall establish the call.

terminatingCall identifies the call to be established.

privateData identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Answer Call" Functional Service identifies the terminating entity as well as the call to be established to this entity. After having sent the request, the computer expects either a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified entity has the capability to answer the call.
2) Allowable basic call view states: Arrived, Received.

Postconditions:

1) The switch established the incoming call to the specified entity.
2) Resulting basic call view states: Established.

Operational rules

On receipt of the service request the switch performs the following actions:

– The specified identifiers are validated.
– The call to the specified entity is established.
– If "private data" is present, they are interpreted; if not understood, discarded.
– A success response is generated.

If monitoring is enabled, the call event "Call Established" may be reported to the computer application.

- **error handling**

An error response should be generated. If the switch is unable to establish the incoming call, the call state may remain unchanged.

f) Management requirements

TASC Management should ensure that:

– The computer is able to verify that the switch can cause the CE to answer calls under switch control. It may not be possible to indicate this accurately in some situations.
– The CE identifier relates to the relevant piece of physical equipment, i.e. the computer is kept informed of any changes (including initialization) to the identifiers for equipment in the operation domain.
– The computer can determine the number of calls that can be simultaneously connected to the CE.
– The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.
– The computer is able to determine if the CE is suitable to be connected to the type of incoming call (if this information is available to switch management).
– Any call forwarding conditions associated with CEs can be determined, and possibly affected.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

6.1.2 **Functional Service name:** Call Completion

a) Relationship: Computer → Switch

b) General description

This Functional Service shall invoke features that complete a call which may otherwise fail or terminate before being answered. This Functional Service is invoked when a call is in progress and encounters a busy called CE or no answer.

c) Parameters

REQUEST:

feature	FeatureInformation	
requesting	CE	CEID
callToBeCompleted	CVID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

feature	identifies the feature to be invoked. The allowed features are: 1) Camp On – queue the call until the CE is available. 2) Call Back – the called CE shall return the call when it returns to idle. 3) Intrude – the caller shall be added to an existing call at the called CE. 4) Call Back Message – leaves a message for the called CE to return the call.
requestingCE	identifies the CE that requests the feature invocation.
callToBeCompleted	identified the call that is to be completed.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer or user detects that the call will not be established.

Postconditions: None additional.

Operational rules

After having sent the request, the client waits for a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that the requested feature is invoked.

If monitoring is enabled, call related event reports may indicate the server actions.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The called destination CE is busy or does not answer the call.
2) The relevant feature is available at the identified CE.
3) Allowable basic call view states for the call to be completed: Arrived, Received.
4) Allowable basic call view states for the call at the called CE: All basic call view states except "Held" and "Null" are valid; however, the for Intrude feature only "Established" is valid.

Postconditions:

1) The specified feature is performed.
2) Resulting basic call view states: Arrived (camped on), Null (call cleared because of call back or Call Back Message), Established (as a result of Intrude).

Operational rules

On receipt of the service request, the server will perform the requested feature.

- If "Call Back" was invoked, the requesting CE returns to the Null state. The switch calls the requesting CE and the called CE when the called CE becomes free.
- If "Call Back Message" was invoked, a message is sent to the called CE requesting a return call.
- If monitoring is enabled, the call related events "Call Cleared" may be reported to the computer application.
- If "Camp On" was invoked, the call is queued at the requesting CE until the called CE has finished its current call and previously camped on calls.
- If "Intrude" was invoked, the switch adds the call to the existing call at the called device.
- If monitoring is enabled, all above followed by Make Call related events (e.g. Service Pending, Originated, Delivered, Received, Established) may be reported to the computer application.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

- The computer can determine the number of calls that can be simultaneously connected to the CE.
- The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.
- The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).
- Any call affecting conditions (i.e. call forwarding) associated with CEs can be determined, and possibly affected.
- The calls currently being held by a CE can be determined.
- The calls currently active at a CE can be determined.
- The switch should be informed if a service becomes temporarily unavailable.
- Co-ordination of which computer applications can use a service and their domain of influence within the service.

– The computer should be able to determine, and possibly alter, the timers used by the switch for TASC Functional Services:

  a) when expecting a reply to a TASC FS request;

  b) when a TASC FS is deemed completed.

– The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.

– CE identifiers can be verified as valid, active and not barred.

– The calls currently queued ("camped-on") at a CE can be determined.

– The call completion facilities available can be determined.

– The length of text messages which may be displayed can be determined.

– Text messages left at a CE can be modified/deleted.

– A "call back" request can be cancelled.

– The call back requests that are currently pending can be determined for the originating CE.

– The call back requests that are currently pending can be determined for the destination CE.

6.1.3 Functional Service name: Clear Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to release all CEs from the specified call and to eliminate the call itself as a result. The call ceases to exist and all identifiers are released.

c) Parameters

REQUEST:		
callToBeCleared	CVID	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

callToBeCleared	identifies the call to be cleared.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.

2) The user is permitted to clear the call.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Clear Call" Functional Service identifies the call to be cleared. After having sent the request the computer expects either a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The client is permitted to send the “Clear Call” request.
2) The specified call exists.
3) Allowable basic call view states: Pending, Originated, Delivered, Arrived, Received, Distributed, Established, Failed.

Postconditions:

1) The switch performs for each CE the actions to release the call.
2) All Identifiers are freed.
3) Resulting basic call view states for each CE: Null

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.
– The call clearing process is started.
– If “private data” is present, they are interpreted; if not understood, discarded.
– A success response is generated.

If monitoring is enabled, the call related event “Call Cleared” may be reported for each CE released to the computer application.

- **error handling**

An error response should be generated. CEs which have been released are not re-established with the call. The Functional Service “Snapshot Call” could be used to clarify the status of the call.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.
– Any calls and/or associated IDs which were not properly cleared during the TASC FS process can be detected and cleared.

6.1.4 Functional Service name: Conference Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to create a conference between an existing “held” call and another “active” call at a conferencing CE. The conference is identified by a new Call View ID. The basic call view identifiers formerly associated with the conferenced calls are released, and a new basic call view identifier for the resulting conference is created.

Note that it may also be used to create an "n" way conference by repeated use of the service request (i.e. one of the calls is already a conference).

c) Parameters

REQUEST:

heldRelation	RelationID	
activeRelation	RelationID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

conferenceCall	CVID	
controllerCE	CEID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Optionally, for each CE in the Operation Domain which is involved in the conference the following parameters may be given:

involvedCE	CEID	OPTIONAL
originalCall	CVID	OPTIONAL
newCall	CVID	OPTIONAL

Error: ErrorGroups

Parameter description

heldRelation	identifies the held CP at the conferencing CE. The information includes the CE and the associated CV if there is more than one call.
activeRelation	identifies the active CP at the conferencing CE. The information includes the CE and the associated CV if there is more than one call.
conferenceCall	identifies the conference call.
controllerCE	identifies the controlling CE for a "three-party" conference.
involvedCE	identifies a CE involved in the conference.
originalCall	identifies the original call at the involved CE before this use of the Conference FS.
newCall	new identifier for the call involved in the conference at the involved CE.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer process continues.

2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Conference Call" Functional Service identifies the active and held CPs to be conferenced.

After sending the request to the switch, the computer waits for a success response or an error response. A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that there are active and held CPs.

- **error handling**

Not Applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CPs are validated.
2) The specified CPs have a common CE where one CP is active and the other CP is held.
3) Allowable basic call view states: Originated, Delivered or Established.
4) Allowable CP states: CP1 Held, CP2 Active.

Postconditions:

1) The active and held CPs now participate in a new established call.
2) The CE which is common to both the held and active CPs does participate in the new call.
3) Resulting basic call view states: unchanged.
4) Resulting CP states: CP1 Active, CP2 Active.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified CPs are validated and the involved calls and CEs determined.
– The server verifies that the CE common to both parties can support the Conference service.
– The calls associated with the CPs are merged into a single conference call.
– The resulting conference call has the same CV ID as one of the calls just conferenced, or optionally a new call view identifier is generated for the resulting conference call.
– If PrivateData is present, they are interpreted; if not understood, they are discarded.
– A success response is generated.

If monitoring is enabled the call event “Call Conferenced” may be reported to the computer application.

- **error handling**

An error response should be generated:

– If the held and active CPs do not have a common CE.
– If the conference service fails the held and active CP states remain unchanged.

f) Management requirements

TASC Management should ensure that:

– The conference controller can be changed to another CE in the conference.
– The number of conferences that may be created can be determined.
– The limits on the number of calls in a conference can be determined.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The computer is able to determine the service profile for a CE (to determine if Conference is available to the user).

– The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.

– Any calls and/or associated IDs which were not properly cleared during the TASC FS process can be detected and cleared.

6.1.5 Functional Service name: Divert Call

a) Relationship: Computer → Switch

b) General description

This Functional Service sends an incoming call from one CE to another CE, or takes a call ringing at one or more CEs and brings it to a new CE.

c) Parameters

REQUEST:

diversionType	DiversionInformation	
newCalledCE	CEID	
callToBeDiverted	CVID	OPTIONAL (Note)
calledCE	CEID	OPTIONAL (Note)
privateData	TASCPrivateData	OPTIONAL

NOTE – Optional for diversionType "Group pickup" (see Operational Rules).

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

diversionType indicates the type of diversion requested. The parameter should include one of the following:

– deflect: indicates deflect to another CE;

– directed pickup: indicates pickup at a new destination CE;

– group pickup: indicates pickup at a group member CE.

newCalledCE identifies the new destination CE at which the call is to be picked up or to which the call is to be deflected.

callToBeDiverted identifies the call to be diverted.

calledCE identifies the CE which was originally called.

privateData identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions: None additional.

Postconditions: None additional.

Operational rules

If the Diversion Type indicates “deflect” or “directed pickup”, the calledCE and the callToBeDiverted parameters should be included in the request.

After having sent the request, the client waits for a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that call processing is initiated. It does not imply success of call establishment.

If monitoring is enabled, the event report “Call Diverted” may indicate that the service has been performed.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The CE to which the call is to be diverted to is able to receive the call.
2) Allowable basic call view states: Received, Distributed, Established.

Postconditions:

1) The switch performs the call processing actions to bring the call to the new called CE.
2) If monitoring is enabled, the event report “Call Diverted” is sent to the computer application.
3) Resulting basic call view states: Null (calledCE), Delivered (newCalledCE).

Operational rules

On receipt of the service request, the server will check validity of the new called CE and send the call to the specified new destination.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The computer can determine the number of calls that can simultaneously be connected to the CE.

– The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.

– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).

– Any call forwarding conditions associated with CEs can be determined, and possibly affected.

– Any known routing problems which may affect the call are determinable.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a call is deemed connected when it is routed out (of the TASC domain) via a trunk (i.e. when trunk supervision signalling is not available).

– The computer is able to determine if the switch is able to support a new call set-up (e.g. switch loading, trunk availability, etc.).

– Members of the pickup group can be determined and potentially changed.

6.1.6 Functional Service name: Drop CP

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to release a specified CE from a designated call.

Note that this service is used to drop CEs from a conference call and in a basic call has the same effect as the Clear Call Functional Service.

c) Parameters

REQUEST:

involvedRelation	CPID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

involvedRelation	identifies the CP to drop from the call.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.

2) The user is permitted to clear the call.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational Rules

The request for the "Drop CP" Functional Service identifies the CP to be dropped. After having sent the request, the computer expects either a success response or an error response.

- **error handling**

Not Applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Originated, Delivered, Established, Distributed, Failed.

2) Allowable CP states: Active, Held.

Postconditions:

1) Resulting basic call view state: Null.

2) Resulting CP state: Null.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified CP are validated and the involved call and CE determined.

– The involved CE is released from the call.

– If there were only two CEs in the call then the call is cleared.

– If PrivateData is present, they are interpreted; if not understood, they are discarded.

– A success response is generated.

If monitoring is enabled the call event "CP Dropped" may be reported to the computer application.

- **error handling**

An error response should be generated. If the "Drop CP" fails then the CP state remains unchanged.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– Any calls and/or associated IDs which were not properly cleared during the TASC FS process can be detected and cleared.

6.1.7 Functional Service name: Hold Call

a) Relationship: Computer $\rightarrow$ Switch

b) General description

This Functional Service is used to request the switch to temporarily interrupt the existing call at a Line CE and to put in on hold at that CE.

Note that the hold relationship between the holding CE and the "held" call lasts until the call is retrieved or cleared.

c) Parameters

REQUEST:

activeRelation	CPID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

activeRelation	identifies the active CP to be held; the information includes the CE, and the CV, if there is more than one call.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer processes continue.
2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Hold Call" Functional Service identifies the CP to be held. After having sent the request, the computer expects either a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CP exists.
2) Allowable basic call view states: Originated, Delivered, Established.
3) Allowable CP state: Active.

Postconditions:

1) Resulting basic call view states: unchanged.
2) Resulting CP state: Held.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.
– The switch performs the actions to put the active CP at the specified CE on hold.
– If "private data" is present, they are interpreted; if not understood, discarded.
– A success response is generated.

If monitoring is enabled, the call event "Call Held" may be reported to the computer application.

- **error handling**

If the CV within the CP identifier is ambiguous, an error response should be generated.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the Operation Domain.

– The CPs currently being held by a CE can be determined.

– The CPs currently active at a CE can be determined.

6.1.8 Functional Service name: Make Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to create a call between two CEs. The effect is as if the Originating CE placed a call to the Destination CE.

c) Parameters

REQUEST:		
originatingCE	CEID	
destinationCE	CEID	
callType	CallType	OPTIONAL
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
newCall	CVID	
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

originatingCE	identifies the CE on whose behalf the call is being originated.
destinationCE	identifies the CE to be called.
callType	indicates either voice or non-voice, including additional information, e.g. ISDN related information, i.e. Bearer Capability, LLCp, HLCp, the characteristics of the channel.
newCall	identifies the call instance to be attempted between the two CEs.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.

2) The user is permitted to make the call.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational rules

After having sent the request for a call attempt to the switch, the computer waits for a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that call processing is initiated. It does not imply success of call establishment.

If monitoring is enabled, subsequent call related event reports may indicate the progress of the request. The call attempt may fail because the destination CE is busy or otherwise unavailable.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The client is permitted to send the "Make Call" request.

2) The CE on whose behalf the call is to be originated is able to support a new call.

3) The switch is able to initiate a new call (no overload situation).

4) Allowable basic call view states: Null, Pending.

Postconditions:

– Resulting basic call view states: Originated, Established, Failed.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The originating CE identifier is validated.

– The privilege of the originating CE is checked.

– If "callType" is not present, the default is what is compatible to the TE; if the switch does not know the TE, a voice connection will be attempted.

– If "private data" is present, they are interpreted; if not understood, discarded.

– The validity of the destination CE is checked.

– A call attempt is made to set up the requested call.

– A success response is generated.

If monitoring is enabled, the following call related events may be reported to the computer application: "Service Pending", "Call Originated", "Call Delivered", "Call Established", "Call Cleared", "Call Failed".

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

- The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
- The computer can determine the number of calls that can simultaneously be connected to the CE.
- The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.
- The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).
- Any call forwarding conditions associated with CEs can be determined, and possibly affected.
- Any known routing problems which may affect the call are determinable.
- The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
- The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.
- The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a call is deemed connected when it is routed out (of the TASC domain) via a trunk (i.e. when trunk supervision signalling is not available).
- The computer is able to determine if the switch is able to support a new call set-up (e.g. switch loading, trunk availability etc.).

6.1.9 Functional Service name: Retrieve Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to retrieve a previously "held" call at a Line CE.

c) Parameters

REQUEST:

heldRelation	CPID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

heldRelation	identifies the held CP to be retrieved; the information includes the CE, and the associated CV, if there is more than one call.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

- The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Retrieve Call" Functional Service identifies the CP to be retrieved. After having sent the request, the computer expects either a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CP exists.

2) Allowable basic call view states: Originated, Delivered, Established.

3) Allowable CP state: Held.

Postconditions:

1) Resulting basic call view states: unchanged.

2) Resulting CP state: Active.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.

– The switch performs the actions to retrieve the held CP at the specified CE.

– If "private data" is present, they are interpreted; if not understood, discarded.

– A success response is generated.

If monitoring is enabled, the call event "Call Retrieved" may be reported to the computer application.

- **error handling**

If the CV within the CP identifier is ambiguous, an error response should be generated.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the Operation Domain.

– The CPs currently being held by a CE can be determined.

– The CPs currently active at a CE can be determined.

6.1.10 Functional Service name: Transfer Call Service

a) Relationship: Computer $\rightarrow$ Switch

b) General description

This Functional Service is used to request the switch to transfer a call "held" at a CE to the "active" call at the same CE. The "held" and "active" calls at the common CE shall be merged into a new call.

c) Parameters

REQUEST:

heldRelation	CPID	
activeRelation	CPID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

newCall	CVID	
privateData	TASCPrivateData	OPTIONAL

Error:

error	ErrorGroups

Parameter description

heldRelation	identifies the held CP to be transferred. The information includes the CE and the assocaited CV if there is more than one call.
activeRelation	identifies the active CP to which the call should be transferred. The information includes the CE and the associated CV if there is more than one call.
newCall	identifies the new call view as seen by the CE to which the call has been transferred.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer processes continue.
2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the “Transfer Call” Functional Service identifies the active and held CP to be transferred.

After sending the request to the switch, the computer waits for a success response or an error response. A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that there is an active and held CP.

- **error handling**

Not Applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CPs are validated.
2) The specified CPs have a common CE where one CP is active and the other CP is held.

3) Allowable basic call view states: Originated, Delivered, Established.

4) Allowable CP state: CP1: Held, CP2: Active.

Postconditions:

1) The active and held CPs now particpate in a new established call.

2) The CE which was common to both the held and active CPs does not particpate in the new call.

3) Resulting basic call view states: Unchanged.

4) Resulting CP states: CP1 Active, CP2: Active.

Operational rules

On receipt of the Service request, the server performs the following actions:

– The specified CPs are validated.

– The server verifies that the CE common to both parties can support the transfer service.

– The transfer service is initiated at the common CE.

– If privateData is present, they are interpreted; if not understood, they are discarded.

– A success response is generated.

If monitoring is enabled, the call event “Call Transferred” may be reported to the computer application.

- **error handling**

An error response shall be generated:

– If the held and active CPs do not have a common CE.

– If the transfer service fails, the switch does not proceed with the attempt to set up a new call. The held and active CP states remain unchanged.

f) Management requirements

TASC Management should ensure that:

– The common CE identifier relates to a relevant piece of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The computer is able to determine whether the server can support the transfer service and be able to determine at which CE this service is supported.

– The calls currently active and held at a CE can be determined.

– The CE participating within a call can be determined.

– The computer can determine the number of simultaneous calls which can be connected to the CE.

6.2 Specialized Functional Services

The specialized Functional Services group consists of compound functions, that may also be provided by an ordered sequence of two basic Functional Services.

One typical example could be that an agent would require information from a specialist to serve the customer's request of the incoming call, using "Consultation Call". The agent then may still want to keep this connection for further inquiries, using "Alternate Call", and finally he may clear the consulted connection and return to the customer call, using "Reconnect Call".

6.2.1 **Functional Service name:** Alternate Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to provide a compound action, that causes an existing call at a Line CE to be held, followed by the retrieval of a previously "held" call at the same CE. An accepted request causes the specified Line CEs "held" and "active" calls to be swapped.

Note that the effect would be the same as if two Functional Services "Hold Call" and "Retrieve Call" were used in sequence.

c) Parameters

REQUEST:		
activeRelation	CPID	
heldRelation	CPID	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

activeRelation	identifies the active CP to alternate; the information includes the CE, and the Established CV, if more than one call is established.
heldRelation	identifies the Held CP to alternate; the information includes the CE, and the associated CV, if ambiguous.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

1) The computer processes continue.
2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Alternate Call" Functional Service identifies the two CPs to be alternated.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that the specified CE has two calls, one with an active CP and one with a held CP.

If monitoring is enabled, event reports "Call Held" of the call with an active CP, followed by "Call Retrieved" of the call with a held CP may indicate the actions the switch has performed.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CE has at least two calls, one with an active CP and one with a held CP.
2) Allowable CP states: CP1: Active, CP2: Held.
3) Allowable basic call view states: Originated, Delivered, Established.

Postconditions:

1) Resulting basic call view states: unchanged.
2) Resulting CP states: CP1: Held, CP2: Active.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.
– Availability of an active and a held CP at the CE is determined.
– The switch performs the actions to put the active CP at the CE on hold.
– The switch performs the actions to retrieve the previously held CP at the CE.
– If "private data" is present, they are interpreted; if not understood, discarded.
– A success response is generated.

If monitoring is enabled, the call events "Call Held", "Call Retrieved" may be reported to the computer application.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
– The CPs currently being held by a CE can be determined.
– The CPs currently active at a CE can be determined.

6.2.2 Functional Service name: Consultation Call

a) Relationship: Computer → Switch

b) General description

The Consultation Call Functional Service is used to request the switch to provide the compound action, that causes an existing call at a Line CE to be held, followed by the initiation of a new call from the same CE.

Note that the effect would be the same as if two Functional Services "Hold Call" and "Make Call" were used in sequence.

c) Parameters

REQUEST:		
activeRelation	CPID	
destinationCE	CEID	
callType	CallType	OPTIONAL
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
newCall	CVID	
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

activeRelation	identifies the active CP to be held; the information includes the CE, and the Established CV, if ambiguous.
destinationCE	identifies the CE to be called.
callType	indicates either voice or non-voice, including additional information, e.g. ISDN related information, i.e. Bearer Capability, LLCp, HLCp, the characteristics of the channel.
newCall	identifies the call instance to be attempted between the CEs.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.
2) The user is permitted to make a call.

Postconditions:

1) The computer processes continue.
2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Consultation Call" Functional Service identifies the CP to be held and that a new call is to be initiated to the destination CE on behalf of the CE specified with the held CP.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid, that the specified CE has a call with an active CP and that call processing is initiated to set up a new call to the destination CE. It does not imply success of call establishment.

If monitoring is enabled, call related event reports "Call Held" of the call with an active CP, followed by call related event reports that may indicate the progress of the new call. The call attempt may fail because the destination CE is busy or otherwise unavailable.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The switch is able to initiate a new call (no overload situation).
2) The client is permitted to send the request.
3) The specified CE has at least one call with an active CP.
4) The specified CE on whose behalf the new call is to be originated is able to support a new call.
5) Allowable CP state: Active.
6) Allowable basic call view states: Originated, Delivered, Established.

Postconditions:

1) Resulting basic call view states: Originated, Established, Failed (new call).
2) Resulting CP state: Null.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.
– If the CV cannot be identified at the CE, an error response should be generated.
– The privilege of the originating CE is checked.
– If "callType" is not present, the default is what is compatible to the TE; if the switch does not know the TE, a voice connection will be attempted.
– If "private data" is present, they are interpreted; if not understood, discarded.
– The switch performs the actions to put the active CP at the CE on hold.
– The switch performs the actions to set up the requested call.
– A success response is generated.

If monitoring is enabled, the following call events may be reported to the computer application: "Service Pending", "Call Originated", "Call Delivered", "Call Established", "Call Cleared", "Call Failed", "Call Held".

- **error handling**

An error response should be generated.

– If the "hold call" part fails, the switch does not proceed with any attempt to set up a new call.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The CPs currently active at a CE can be determined.

– The computer can determine the number of calls that can simultaneously be connected to the CE.

– The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.

– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).

– Any call forwarding conditions associated with CEs can be determined, and possibly affected.

– Any known routing problems which may affect the call are determinable.

6.2.3 Functional Service name: Reconnect Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to provide a compound action, that causes an existing active CP (in any state) at the associated CE to be cleared, followed by the retrieval of a previously "held" call at the same CE.

An accepted request causes the CE's "active" call to be cleared and the "held" call to become active.

Note that the effect would be the same as if two Functional Services "Clear Call" and "Retrieve Call" were used in sequence.

c) Parameters

REQUEST:		
activeRelation	CPID	
heldRelation	CPID	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

activeRelation	identifies the active CP to be cleared; the information includes the CE, and the Established CV, if ambiguous.
heldRelation	identifies the Held CP to be retrieved; the information includes the CE, and the associated CV, if ambiguous.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.

2) The user is permitted to clear the call.

Postconditions:

1) The computer processes continue.

2) If monitoring is enabled, event reports are expected.

Operational rules

The request for the "Reconnect Call" Functional Service identifies the CPs to be released/retrieved) A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that the specified CE has two calls, one with an active CP (in any state) and one with a held CP.

If monitoring is enabled, call related event reports "Call Cleared" of the call with an active CP, followed by "Call Retrieved" of the call with a held CP may indicate the actions the switch has performed.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The client is permitted to send the request.
2) The specified CE has at least two calls, one with an active CP and one with a held CP.
3) Allowable CP states: CP1: Active, CP2: Held.
4) Allowable basic call view states: Originated, Delivered, Established.

Postconditions:

1) Resulting basic call view states: Null (active call), Unchanged (held call).
2) Resulting CP states: CP1: Null, CP2: Active.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.
– Availability of an active and a held CP at the CE is determined.
– The switch performs the actions to clear the active CP.
– The switch performs the actions to retrieve the previously held CP at the CE.
– If "private data" is present, they are interpreted; if not understood, discarded.
– A success response is generated.

If monitoring is enabled, the call events "Call Cleared", followed by "Call Retrieved" may be reported to the computer application.

- **error handling**

An error response should be generated. If the Clear Call part fails, the retrieval action is not performed.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– Any calls and/or associated IDs which were not properly cleared during the TASC FS process can be detected and cleared.

6.3 Manipulate Functional Services

6.3.1 Manipulate Feature Functional Services

Feature Functional Services consist of functions concerned with manipulating user features at a CE. It does not set system features or allow administration.

6.3.1.1 **Functional Service name:** Manipulate Feature

a) Relationship: Computer → Switch

b) General description

The Manipulate Feature Functional Service is a request towards the switch to set or clear features on a line CE.

A success response indicates receipt of the correct message and that the requested Functional Service has been completed.

c) Parameters

REQUEST:		
lineCEToBeSet	CEID	
featureInformation	FeatureInformation	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

lineCEToBeSet identifies the line CE on whose behalf the feature is being set or cleared.

featureInformation identifies one of the following feature types and the information associated. The parameter shall have one of the following values:

1) Forward – sets forwarding calls. If this parameter is chosen the following additional parameters shall be included:

   a) Type of Forwarding – The value shall be one of the following with meaning as indicated:

   - Unconditional – forwarding all calls.
   - Unconditional Internal – forwarding all internal calls.
   - Unconditional External – forwarding all external calls.
   - Busy – forwarding when busy for all calls.
   - Busy Internal – forwarding when busy for an internal call.

- Busy External – forwarding when busy for an external call.
- No Answer – forwarding after no answer for all calls.
- No Answer Internal – forwarding after no answer for an internal call.
- No Answer External – forwarding after no answer for an external call.

b) Forwarded to CE – shall indicate the CE to which calls are forwarded.

c) Type of request – shall indicate whether to set or clear the feature.

2) Message Waiting – shall indicate messages available. If this parameter is chosen the following parameter shall be included:

- Type of request – shall indicate whether to set or clear the feature.

3) Do Not Disturb – shall set Do Not Disturb If this parameter is chosen the following additional parameter shall be included:

- Type of request – shall indicate whether to set or clear the feature.

privateData identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.

2) The user is allowed to set or clear a line CE feature.

Postconditions:

– The computer processes continue.

Operational rules

After having sent the request to the switch to set or clear feature at a specified CE, the computer waits for a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid, and that the requested Functional Service has been completed.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The client is permitted to send the request.

2) The CE on whose behalf the feature is to be set or cleared is able to provide the requested feature.

3) The specified CE may be involved in a call at the time of this request.

Postconditions:

– The switch performs the actions to set or clear the requested feature at the specified CE.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.

– The features available at the CE are determined.

– The requested feature is set or cleared.

– If a call forwarding feature was to be set or cleared, the forwarded CE is validated.

- **error handling**

Not applicable.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– CE identifiers can be verified as valid, active and not barred.

– The calls currently active at a CE can be determined.

– The computer can determine the number of calls that can simultaneously be connected to the CE.

– The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.

– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).

– A history (timed and dated) and source of changes (e.g. from network management) to the call forwarding settings of a CE can be determined.

– The privileges necessary to set or clear a feature can be determined.

– The features supported by a CE can be determined.

6.3.1.2 Functional Service name: Query Feature

a) Relationship: Computer $\rightarrow$ Switch

b) General description

The Query Feature Functional Service is a request towards the switch to query features on a line CE.

c) Parameters

REQUEST:		
lineCEToBeQueried	CEID	
featureOfInterest	QueryFeature	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

featureInformation	QueryFeatureInformation	
privateData	TASCPrivateData	OPTIONAL

Error: ErrorGroups

Parameter description

lineCEToBeQueried identifies the line CE to be queried.

featureOfInterest identifies the feature to be queried and consists of one of the following:

- Message Waiting – shall indicate messages available
- DoNotDisturb – shall indicate Do Not Disturb
- Forward – shall indicate Forwarding.

featureInformation identifies one of the following feature types and the information associated. The parameter shall have one of the following values:

1) Forward – forwarding calls is set. If this parameter is chosen the following additional parameters shall be included:

   a) Type of Forwarding – The value shall be one of the following:

      - Unconditional – forwarding all calls.
      - Unconditional Internal – forwarding all internal calls.
      - Unconditional External – forwarding all external calls.
      - Busy – forwarding when busy for all calls.
      - Busy Internal – forwarding when busy for an internal call.
      - Busy External – forwarding when busy for an external call.
      - No Answer – forwarding after no answer for all calls.
      - No Answer Internal – forwarding after no answer for an internal call.
      - No Answer External – forwarding after no answer for an external call.

   b) Forwarded to CE – shall indicate the CE to which calls are forwarded.

2) Message Waiting – shall indicate Message Waiting is set.

3) Do Not Disturb – shall indicate Do Not Disturb is set.

privateData identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that the specified Functional Service request has to be sent.

Postconditions:

– The computer processes continue.

Operational rules

After having sent the request to the switch to query a feature at a specified CE, the computer waits for a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid, and that the requested Functional Service has been completed.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

– The specified CE may be involved in a call at the time of this request.

Postconditions: None.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The specified identifiers are validated.

– The feature of interest is determined.

– If “private data” is present, they are interpreted; if not understood, discarded.

– A success response is generated.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– CE identifiers can be verified as valid, active and not barred.

– The calls currently active at a CE can be determined.

– The computer can determine the number of calls that can simultaneously be connected to the CE.

– The computer is able to determine, and possibly affect, the restrictions about calls which may be connected to a specific CE.

– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).

– A history (timed and dated) and source of changes (e.g. from network management) to the call forwarding settings of a CE can be determined.

– The features supported by a CE can be determined.

6.3.2 Manipulate Agent Functional Services

6.3.2.1 **Functional Service name:** Manipulate Agent

a) Relationship: Computer → Switch

b) General description

This Functional Service provides the ability for an agent to log-on, log-off, or change state at a CE.

c) Parameters

REQUEST:		
agentLineCE	CEID	
agentRequest	AgentRequestInformation	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

agentLineCE identifies the line CE of the agent.

agentRequest specifies the agent operation and associated information, i.e:

– Agent Operation (AgentLogOn, AgentLogOff, AgentReady, AgentNotReady, AgentBusy, AgentWorkingAfterCall);

– Agent Identifier;

– Agent Password;

– Group Identifier.

privateData identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that the specified Functional Service request has to be sent.
2) An agent must be associated with the line CE.
3) The agent can log-on to a specific group only once.
4) The agent can log-on to multiple groups.

Postconditions:

1) The computer processes continue.
2) If monitoring is enabled, event reports are expected.

Operational rules

After having sent the request, the computer expects either a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

– The agent can be associated with the specified line CE.

Postconditions:

– The agent state is manipulated as indicated by the agent operation.

Operational rules

– The agent is permitted to carry out the functions at the CE.

– The Agent Identifier, Agent Password and Group Identifier are validated.

– The server initiates the process to manipulate the agent as initiated in the Functional Service request.

– If "private data" is present, they are interpreted; if not understood, discarded.

– A success response is generated.

If monitoring is enabled, the events "AgentLoggedOn", "AgentLoggedOff", "AgentNotReady", "AgentReady", "AgentBusy", "AgentWorkingAfterCall" may be reported to the computer application.

- **error handling:**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.

– CE identifiers can be verified as valid, active and not barred.

– The "features" supported by a CE can be determined.

– The CEs at which an agent is permitted to "log on" can be determined, and potentially altered.

– The agent group(s) permitted to the agent can be determined, and potentially altered.

– The agent's security arrangements (e.g. password) can be determined and altered.

– The agent state can be altered.

6.3.2.2 Functional Service name: Query Agent

a) Relationship: Computer $\rightarrow$ Switch

b) General description

This Functional Service provides the ability to query an agent object and identify the agent states, CE and group identifier.

c) Parameters

REQUEST:		
agent	AgentID	
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
agentInformation	QueryAgentInformation	
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

agent	identifies the agent to query.
agentInformation	the information returned in response to the query, i.e.: – Agent Identifier; – Agent States (agent states in the groups); – CE Identifiers (CEs logged onto); – Group Identifier (groups logged onto at the CEs).
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– An agent identifier is allocated and valid.

Postconditions:

– The computer processes continue.

Operational rules

After having sent the request, the computer waits for a success response or an error response.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions: None.

Postconditions: None.

Operational rules

On receipt of the request, the switch performs the following actions:

– The specified parameters are validated.

– If “private data” is present, they are interpreted; if not understood, discarded.

– A success response is generated.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

- The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.
- CE identifiers can be verified as valid, active and not barred.
- Members of an agent group can be determined, and potentially altered.

6.4 Routing Functional Services

This is a grouping of Functional Services which enable the switch to seek routing information from the computer.

The Functional Services comprising this group fall into two categories:

Call Related Routing Functional Services:

- Set Routing: allows conditions for switch to generate Route Call requests to be set.
- Route Call: requests the computer to provide an alternative target destination CE for a call.
- Route Call Selected: route information returned by the computer to the switch for a call.
- Route Used: indicates to the computer the final route taken by a call.

Non-Call Related Routing Functional Services:

- Route Information: requests the computer to provide alternative target destination information.
- Route Information Selected: route information returned by the computer to the switch as part of a dialogue.
- Re-Route Information: request for the computer to provide an alternative route based upon a previous Route Call request.
- Route Information End: indication given that a routing dialogue is terminated.

The call related Functional Services are invoked for individual calls (incoming and outgoing) to assist in determining the route for the call. The non-call related Functional Services are used by the switch to update or verify its own routing information without help of the computer application. This information can then be used by the switch when routing calls.

Route Call and Route Information are equivalent functional services with the former being call related and the latter non-call related. These two functional services share a common set of parameters used as the basis for the computer to provide its reply. In a similar way Route Call Selected and Route Information Selected are also equivalent functional services with the former being call related and the latter non-call related and share a common set of parameters. These common parameters are set out below and will be referred to in the relevant functional service definitions.

Common parameters

commonRouteCallInfo CommonRouting

This type is comprised of the following parameters:

targetDestinationCE	CEID	
callType	CallType	OPTIONAL
routeCategory	RouteType	OPTIONAL
iSDNSet-up	ISDNValues	OPTIONAL
privateData	TASCPrivateData	OPTIONAL
commonCallInfoSelected	CommonInfo	

This type is comprised of the following parameters:

selectedDestinationCE	CEID	
moreRoutes	MoreRouteFlag	OPTIONAL
iSDNSet-up	ISDNValues	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Common parameter descriptions

targetDestinationCE	indicates the CE that the switch currently has targeted as a destination. This is the default destination.
callType	indicates either voice or non-voice, including additional information, e.g. ISDN related information, i.e. Bearer Capability, LLCp, HLCp, the characteristics of the channel.
routeCategory	indicates the routing category which should be used: – ACD: for distributing calls to multiple devices; – Emergency: for emergency calls; – Least Cost Routing: selects route of least costs; – Normal routing: an additional route is required without the use of specialized algorithms.
iSDNSet-up	information specific to ISDN type calls which may not be fully represented by the callType parameter.
selectedDestinationCE	indicates the CE that the computer suggests as an alternative to the switches "targetDestinationCE".
moreRoutes	indicates that the computer has options of more routes available.
privateData	identifies custom data specific to an implementation or application.

6.4.1 Call Related Routing

These Functional Services are interdependent as depicted in Figure 1. Although Route Call may have an error response, the requested information is returned by the Route Call Selected FS.

6.4.1.1 **Functional Service name:** Set routing

a) Relationship: Computer → Switch

b) General description

This Functional Service is used by the computer to set the conditions for which the switch should generate a Route Call request.

Each CE shall support only a single "trip" but this may specify multiple conditions. Prior to the first use of this functional service for a CE, the "trip" is set according to the switch default.

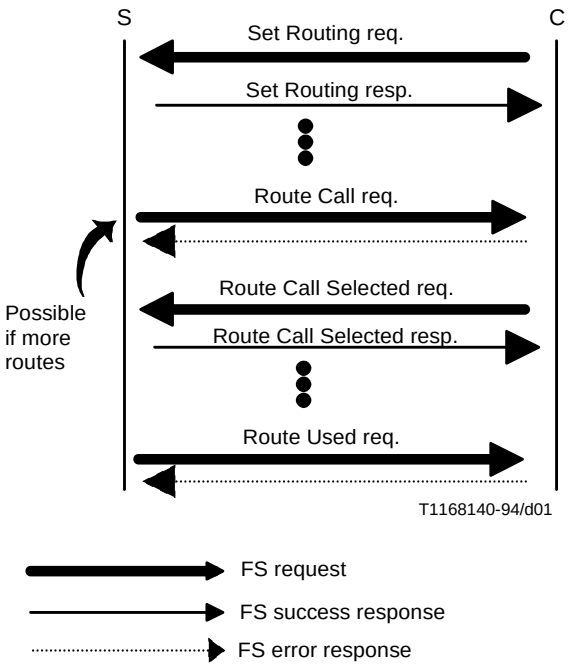

FIGURE 1/Q.1302

Relationship between FSs in Call Related Routing group

c) Parameters

REQUEST:

subjectCE	CEID	
routingRequired	ActivateRoutingInfo	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

subjectCE	identifies the CE for which the routing "trip" is to be manipulated.
routingRequired	gives information relevant to enabling or disabling the generation of a Route Call request. Alternatively, it may be set to its switch maintained default state. If enabling then consists of:
tripRequired	identifies if the generation of a Route Call request is to be enabled or disabled for the CE
	If enabled then the following information applies:
tripState	optionally identifies the valid call states.

	tripType	optionally indicates if the routing "trip" is enabled for either voice or non-voice, internal/external. It consists of callType and routeCategory parameters.
	privateData	Identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has determined that a routing "trip" has to be altered for a CE.

2) The computer has determined the conditions for the routing "trip".

3) The computer shall provide routing information for the conditions set by the "trip".

Postconditions: Unchanged.

Operational rules

None additional.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The CE identified is able to support a routing "trip" with the indicated conditions.

2) If the routing "trip" for the CE has not been altered before then, the switch is able to support further routing "trips" which differ from the switch default.

Postconditions:

1) The switch notes that a routing "trip" which replaces the default now applies for the CE.

2) Calls to the CE which match the conditions of the routing "trip" will cause a Route Call request to be sent to the computer.

Operational rules

The switch is maintaining a routing "trip" on the relevant CE. A call meeting the conditions specified will activate the trip. Such conditions do not impact the use of the Route Information FS.

On receipt of the service request, the switch performs the following actions:

– The CE identifier is validated.

– The suitability of the CE to support a routing "trip" is determined (all conditions must match).

– The availability of resources to support the routing trip is determined.

– The "trip" conditions are noted for the CE.

Changing the "trip" for a CE does not affect any routing dialogue that has already started.

- **error handling**

The "trip" conditions remain unchanged for the CE.

An error response should be generated.

f) Management requirements

TASC Management should ensure that:

- The default states for the routing "trips" on CEs shall be determined, and potentially altered.
- It is possible to identify which CEs can support routing "trips".
- It is possible to determine how many non-default routing "trips" the switch can support.
- It is possible to determine which condition categories the switch can support.
- All, or individual, CEs can be forced back to their default state for routing "trips".
- All, or individual, CEs can be forced to disable their routing "trip".
- The current status of all, or individual, routing "trips" can be determined.
- The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

6.4.1.2 Functional Service name: Route Call

a) Relationship: Switch → Computer

b) General description

This Functional Service is used to request the computer to provide an alternative target destination CE for a call based upon the given call details.

Note that routing towards the alternative target CE provided by the computer may still be subject to changes based upon other routing considerations (e.g. Line CE forwarding, route optimization).

c) Parameters

REQUEST:

callToRoute	CVID	
commonRouteCallInfo	CommonRouting	
originalDestination	CEID	OPTIONAL
callingCE	CEID	OPTIONAL
cause	Cause	OPTIONAL

RESPONSE:

The server shall not give a success response to the service request; however, the server may provide an error response. The requested route information shall be provided via a Route Call Selected reply from the computer.

Success:	Not applicable
Error:	ErrorGroups

Parameter description

commonRouteCallInfo	see the Common Parameter descriptions in the Routing group section.
callToRoute	identifies the call to be routed.
originalDestination	indicates the dialled number that the call originally had specified as a destination.

callingCE	indicates the CE that originated the call.
cause	identifies the transition cause leading to the request.

d) Client (switch)

- **normal procedure**

Preconditions:

1) The switch has determined that routing information should be sought for the call.
2) Allowable basic call view states: Arrived, Failed, Pending, Originated, Distributed.

Postconditions:

1) The switch starts a timer for a Route Call Selected reply. If this timer expires then the call will be routed to the "targetDestinationCE".
2) The switch expects a Route Call Selected reply for the call.
3) Resultant basic call view states: unchanged.

Operational rules

The switch may decide to use this functional service for the following reasons:

– The call conditions coincided with conditions set by the Set routing FS.
– A previous Route Call Selected reply for the same "callToRoute" indicated that the computer had further alternatives which the switch wishes to see.
– Proprietary reasons.

If the switch is seeking further routing alternatives for a call, then the optional parameters are omitted and the mandatory parameters are specified the same as in the initial Route Call request for the call.

Having sent the request for routing information to the computer, the switch waits for one of the following:

– A reply in a form of a Route Call Selected message for the call.
– An error response from the computer.
– A timeout.

In the event of a timeout or error response (dependent on the type of error) the switch will continue routing the call using the targetDestinationCE.

- **error handling**

On receipt of an error response, default procedures are activated for general errors. Dependent upon the type of error, the switch may either:

a) retry with another Route Call message with the same, or a revised timer;
b) continue routing the call using the targetDestinationCE;
c) if a "Clear Call" FS request is received, the call is released and the routing dialogue is ended.

e) Server (computer)

- **normal procedure**

Preconditions:

– The computer has sufficient capacity and response time to handle the request within the switch timeout period.

Postconditions:

– The computer responds via the Route Selected FS which contains the required information.

Operational rules

On receipt of the service request, the computer performs the following actions:

– Validates CEs.

– Determines if this is a request for alternative routes.

– Selects the type of routing to apply based upon the "routeCategory", if it has not already done so.

– Uses the supplied information to determine a suitable destination CE or the next alternative.

– Reply information is returned via Route Call Selected FS.

- **error handling**

An error response should be generated. On error, the routing dialogue is terminated.

f) Management requirements

TASC Management is not responsible for coordinating the routing information maintained by the computer with configuration changes which take place in the telecommunications network. Such coordination will require the computer to interact with the network management for the telecommunications network.

TASC Management should ensure that:

– The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.

– The switch should be informed if this service becomes temporarily unavailable on the computer.

– The switch should be informed when any changes are made to the routing information held by the computer that is likely to change the routing information given.

– The computer should be able to view the conditions used by the switch to determine when to seek routing information from the computer. The computer should also be able to request changes to those conditions.

– Coordination of which computer applications can use this service and their domain of influence within the service.

– The switch is able to determine the likely computer response times to service requests.

– The computer should be able to determine, and possibly alter, the timer used to restrict the period for which the call is paused whilst waiting to be routed.

6.4.1.3 Functional Service name: Route Call Selected

a) Relationship: Computer → Switch

b) General description

This Functional Service is used by the computer to return route information for a current call to the switch in reply to a Route Call request.

c) Parameters

REQUEST:

callToRoute	CVID
commonCallInfoSelected	CommonInfo
provideRouteUsed	RouteUsedFlag

RESPONSE:

Success:

routingStarted	RoutingStartedFlag	
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

commonCallInfoSelected	see the Common Parameter descriptions in the Routing group section.
provideRouteUsed	requests the switch to use the Route Used FS to send information about the eventual route.
routingStarted	indicates that the switch has started routing the call and that no further Route Call requests will be made for this call. Thus, the computer will not have to provide further alternative routes for the call.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has used the information provided in a Route Call request to identify a suitable target CE.

2) The computer has determined if it has other alternative targets to suggest for the same call.

3) The computer has decided if it wishes to be informed of the CE to which the call was eventually routed.

Postconditions:

1) If the computer has indicated that it has further alternative targets available, then it will retain awareness of the "callToRoute" for further Route Call requests.

2) If the computer has requested to be informed of the CE to which the call was eventually routed, then it will retain awareness of the "callToRoute" and expect a Route Used reply.

Operational rules

None additional.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

– The switch has sent a Route Call request for the call identified by "callToRoute".

Postconditions:

– A routing failure when the computer has indicated that it has further alternatives available may cause the switch to try another Route Call request for the same call.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The switch verifies the destination CE identified.

– The switch attempts to route the call to the destination CE.

– The switch prepares to note the final destination of the call if this has been requested by the computer via a "provideRouteUsed" flag.

– If the "provideRouteUsed" flag indicates the need to track the call and send a Route Used reply then this will be noted. The use of Route Used will not affect any monitoring activity.

- **error handling**

If there is a failure during routing and the computer has indicated that it has alternative destination CEs, then the switch may either:

a) retry with another Route Call request using the same timer;

b) continue routing using the targetDestinationCE to route the call if the timer has expired or is likely to expire before the computer can respond;

c) if a "Clear Call" FS request is received, the call is released and the routing dialogue is ended.

f) Management requirements

TASC Management should ensure that:

– The computer should be able to determine, and possibly alter, the timer used to restrict the period for which the call is paused whilst waiting to be routed.

6.4.1.4 Functional Service name: Route Used

a) Relationship: Switch → Computer

b) General description

This Functional Service is used by the switch to indicate to the computer the eventual Line CE selected for the call for which a Route Call request had been made.

This Functional Service is enabled by the "provideRouteUsed" flag in the Route Call Selected FS.

No success response will be given.

c) Parameters

REQUEST:

routedCall	CVID	
targetDestination	CEID	
cause	RoutingCause	OPTIONAL
callingCE	CEID	OPTIONAL
inDomain	DomainFlag	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

The server shall not give a success response to the service request; however, the server may provide an error response.

Success:	Not applicable
Error:	ErrorGroups

Parameter description

routedCall	identifies the call that was routed (i.e. "callToRoute" in Route Call request).
targetDestination	identifies the selected destination line CE at which the call is expected to terminate.
cause	identifies the transition cause leading to the request, and is a selection of the following, e.g.: • Destination Alerting; • ReRouted; • Route Determined; • Destination Not Obtainable; • Destination Out of Order; • Incompatible Destination; • Network Congestion/Not Obtainable/Out of Order; • No Route to Destination; • Number Changed.
callingCE	indicates the CE that originated the call.
inDomain	indicates whether the target destination is within the Operating Domain.
privateData	identifies custom data specific to an implementation or application.

d) Client (switch)

- **normal procedure**

Preconditions:

1) The identified call was the subject of a previous Route Call request.
2) The computer had requested this service in the last Route Call Selected request for the call.
3) Allowable basic call view states: Arrived, Received, Failed.

Postconditions:

1) The switch has finished tracking the call.
2) Resulting basic call view states: unchanged.

Operational rules

The switch has noted that a call for which tracking had been requested in a previous Route Call Selected request from the computer has either Arrived (or been Received) or has Failed.

- **error handling**

Not applicable.

e) Server (computer)

- **normal procedure**

Preconditions:

– The computer had requested this service in the last Route Call Selected request for the call.

Postconditions:

– The computer does not expect further call routing information for the call.

Operational rules

The computer may use the information about the target destination of the call.

- **error handling**

If a previous Route Used FS had been received for the call then:

– If the information provided was the same, then the request will be ignored and TASC Management notified.

– If the information is different, then the previous and current route information will be treated as unsafe and an error reported to TASC Management.

f) Management requirements

TASC Management should ensure that:

– The computer is aware of the current relationship of CE identifiers to physical equipment within the TASC Working Domain.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

6.4.2 Non-Call Related Routing

These Functional Services are interdependent, as depicted in Figure 2. Although Route Information and Re-Route may have an error response the requested information is returned by the Route Information Selected FS.

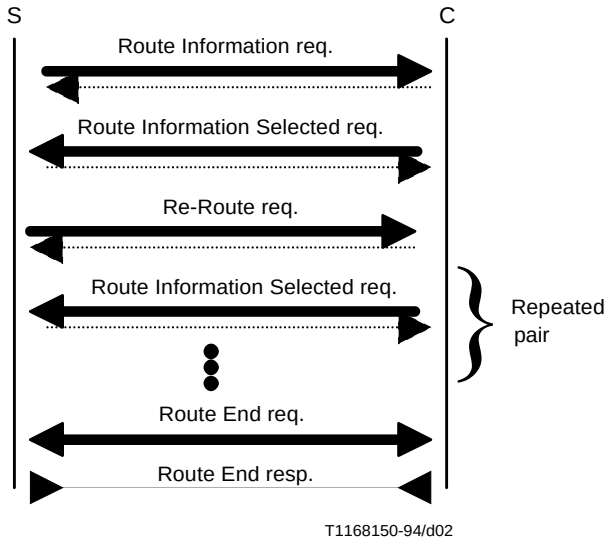

FIGURE 2/Q.1302

Relationship between FSs in Non-Call Related Routing group

The sequence of interactions depicted in Figure 2 is referred to in this Recommendation as a "routing dialogue". These dialogues start with the Route Information FS and end with the Route Information End FS.

6.4.2.1 Functional Service name: Route Information

a) Relationship: Switch → Computer

b) General description

This Functional Service is used to request the computer to provide an alternative target destination CE based upon described call conditions.

Note that routing towards the alternative target CE provided by the computer may still be subject to changes based upon other routing considerations (e.g. Line CE forwarding, route optimization).

c) Parameters

REQUEST:

crossRefID	RoutingRefID	
commonRouteCallInfo	CommonRouting	
typeOfService	RoutingServiceType	OPTIONAL

RESPONSE:

The server shall not give a success response to the service request; however, the server may provide an error response. The requested route information shall be provided by the computer via the Route Information Selected FS.

Success:	Not applicable
Error:	ErrorGroups

Parameter description

commonRouteCallInfo	see the Common Parameter descriptions in the Routing group section.
crossRefID	identifier to link together messages that form part of a routing dialogue (see Figure 2).
typeOfService	indicates the type of routing service for which the switch wants the information, e.g: – Time of Day: routing based on time; – Night Service: routing based on a day or night distinction; – Alternative: routing based on alternatives to the original route; – Optimized: routing based on the "best" route; – Personal: routing based on individual user requirements.

d) Client (switch)

- **normal procedure**

Preconditions:

1) The switch has determined that routing information should be sought.

2) The switch has sufficient capacity to process the reply.

Postconditions:

– The switch expects a Route Information Selected FS for the "crossRefID".

Operational rules

Having sent the request for routing information to the computer, the switch expects the Route Information Selected FS to return the appropriate information or an error response from the computer.

The switch allocates a unique cross Reference Identifier which lasts until "Route Information End" terminates the routing dialogue. The exact nature of the types of routing service is implementation dependent.

- **error handling**

Not applicable.

e) Server (computer)

- **normal procedure**

Preconditions:

1) The computer has sufficient capacity and response time to handle the request.

2) The "crossRefID" is not already in use.

Postconditions:

1) The "crossRefID" is noted for further routing dialogue.

2) The computer invokes a Route Information Selected FS to return the appropriate information.

Operational rules

On receipt of the service request, the computer performs the following actions:

– Verifies that the "crossRefID" is not already in use.

– Validates CEs.

– Selects the type of routing to apply based upon the "routeCategory" and "typeOfService"; if the information given is not sufficient, an error response is generated.

– Uses the supplied information to determine a suitable destination CE.

– Invokes a Route Information Selected FS to return the appropriate information.

- **error handling**

An error response should be generated.

In addition, if the computer is unable to support requests for the "crossRefID" or the "crossRefID" is already in use then it will send a Route Information End request to the switch.

f) Management requirements

TASC Management should ensure that:

– It is possible to clear individual or all "crossRefID" in use (initiated by either computer or switch). When a "crossRefID" is cleared, all processing associated with the reference should be terminated.

6.4.2.2 Functional Service name: Route Information Selected

a) Relationship: Computer → Switch

b) General description

This Functional Service is used by the computer to return route information to the switch as requested by the Route Information FS, or subsequently by the Re-Route Information FS.

c) Parameters

REQUEST:

crossRefID	RoutingRefID
commonCallInfoSelected	CommonInfo

RESPONSE:

The server shall not give a success response to the service request; however, the server may provide an error response.

Success:	Not applicable
Error:	ErrorGroups

Parameter description

commonCallInfoSelected	see the Common Parameter descriptions in the Routing group section.
crossRefID	identifier to link together messages that form part of a routing dialogue (see Figure 2).

d) Client (computer)

- **normal procedure**

Preconditions:

1) The computer has used the information provided in the Route Information request identified by the "crossRefID" to determine an alternative target CE.
2) The computer has determined if it has other alternative targets to suggest for the same call.

Postconditions:

– The computer retains awareness of the "crossRefID" for further Re-Route Information requests until a Route Information End request for the "crossRefID" is received.

Operational rules

None additional.

- **error handling**

On receipt of an error response, default procedures are activated for general errors.

If the computer is unable to continue the routing dialogue, then it will not specify a "selectedDestinationCE" and use "moreRoutes" to indicate that there are no further alternatives.

e) Server (switch)

- **normal procedure**

Preconditions:

– The switch has sent a Route Information request identified by "crossRefID".

Postconditions:

– The switch uses the routing information.

Operational rules

On receipt of the service request, the switch performs the following actions:

– If the switch does not require further routing information, then it sends a Route Information End request to the computer.

– If the computer has indicated that it does not have any further routing information for the conditions indicated in the original Route Information request for the "crossRefID", then it sends a Route Information End request to the computer.

– If the computer has indicated that it has further routing information for the conditions indicated in the original Route Information request for the "crossRefID", then it may send a Re-Route Information request to the computer.

- **error handling**

If the "crossRefID" is unknown to the switch or the switch wishes to terminate the routing dialogue due to the failure, then it will send a Route Information End request to the computer.

If the computer has not set "selectedDestinationsCE" and has indicated that there are no more alternatives, then the switch will send a Route Information End request to the computer to terminate the dialogue.

f) Management requirements

TASC Management should ensure that:

– It is possible to clear individual or all "crossRefID" in use (initiated by either computer or switch). When a "crossRefID" is cleared all processing associated with the reference should be terminated.

6.4.2.3 Functional Service name: Re-Route Information

a) Relationship: Switch → Computer

b) General description

This Functional Service is used by the switch to request the computer to provide information on an alternative route based upon a previous Route Information request as part of a routing dialogue.

c) Parameters

REQUEST:

crossRefID	RoutingRefID	
selectedDestinationCE	CEID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

The server shall not give a success response to the service request; however, the server may provide an error response.

Success:	Not applicable
Error:	ErrorGroups

Parameter description

crossRefID	identifier to link together messages that form part of a routing dialogue (see Figure 2).
selectedDestinationCE	duplicates the last destination that the computer indicated in a Route Information Selected response.
privateData	identifies custom data specific to an implementation or application.

d) Client (switch)

- **normal procedure**

Preconditions:

1) Further routing information is required for the same call conditions as specified in an earlier Route Information request.

2) The switch has sufficient capacity to process the Route Information Selected FS.

Postconditions:

– The switch expects a Route Information Selected FS.

Operational rules

Having sent the request for routing information to the computer, the switch expects a Route Information Selected FS or an error response from the computer.

- **error handling**

On receipt of an error response, default procedures are activated for general errors.

Any error response will terminate the routing dialogue.

e) Server (computer)

- **normal procedure**

Preconditions:

1) The computer has sufficient capacity and response time to handle the request.

2) The "crossRefID" is already in use.

3) The computer has indicated more alternatives in the last Route Information Selected FS.

Postconditions:

– The computer invokes another Route Information Selected FS to return the appropriate information.

Operational rules

On receipt of the service request, the computer performs the following actions:

– Verifies that the "crossRefID" is already in use.

– Uses the previously supplied information, and the "selectedDestinationCE", to determine the next suitable destination CE.

– Invokes a Route Information Selected FS to send the appropriate information.

- **error handling**

An error response should be generated.

An error response terminates the routing dialogue so the Route Information End FS must be used to follow an error condition.

f) Management requirements

TASC Management should ensure that:

– It is possible to clear individual or all "crossRefID" in use (initiated by either computer or switch). When a "crossRefID" is cleared all processing associated with the reference should be terminated.

6.4.2.4 **Functional Service name:** Route Information End

a) Relationship: Switch ↔ Computer

b) General description

This Functional Service indicates that a routing dialogue is terminated. It may be given by either the computer or switch.

No response is sent to this request.

c) Parameters

REQUEST:

crossRefID	RoutingRefID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

The server shall not give a success response to the service request nor an error response.

Success:	Not applicable
Error:	Not applicable

Parameter description

crossRefID	identifier to link together messages that form part of a routing dialogue (see Figure 2).
privateData	identifies custom data specific to an implementation or application.

d) Client (switch/computer)

- **normal procedure**

Preconditions:

1) Further routing information is not required/possible.
2) The “crossRefID” is currently in use by either switch or computer.

Postconditions:

1) The “crossRefID” is freed.
2) Any routing algorithms affected by the “crossRefID” are reset.

Operational rules

None additional.

- **error handling**

Not applicable.

e) Server (computer/switch)

- **normal procedure**

Preconditions:

– The “crossRefID” is in use by either switch or computer.

Postconditions:

1) The “crossRefID” is freed.
2) Any routing algorithms affected by the “crossRefID” are reset.

Operational rules

None additional.

- **error handling**

Not applicable.

f) Management requirements

TASC Management should ensure that:

– It is possible to clear individual or all “crossRefID” in use (initiated by either computer or switch). When a “crossRefID” is cleared, all processing associated with the reference should be terminated.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when expecting a reply to a TASC FS request.

– The current log (switch and computer) of unknown “crossRefID's” can be reviewed.

6.5 Monitor Functional Services

TASC defines a group of functional services for reporting state changes of TASC objects. This is achieved by first invoking a monitor, with an associated monitor filter, upon a TASC object. State changes of TASC objects (CEs and CVs) are then reported by the switch to the computer through an event reporting mechanism.

Monitoring can be accomplished in two ways:

– Statically before association, via subscription arrangements, which indicate which CEs are to be visible and which events are to be reported for each CE. The mechanisms used for this are not covered by TASC Recommendations.

– Dynamically via a Monitor function as described below.

Monitor Start Service

The Monitor Start Service enables event reports for a Call or CE. Event reports will be generated for all end-points which are visible. There are two types of monitors defined by TASC: a CE monitor type and a Call monitor type.

CE Monitor Type

The CE Monitor Type will report call progress information for calls which reside at CEs which have enabled CE monitors. The set of enabled CE monitors established by an application forms the TASC Working Domain. With this type of monitor call progress events are sent from a switch to a computer for all the calls involving the specified CE as those calls progress through the TASC originating and terminating call models.

Call Monitor Type

Call monitoring provides call progress information for all CEs involved in a call. During the life of a call, regardless of the operations performed on that call, this service will continue to provide call progress information for as long as the call remains within the TASC Operation Domain. Call Monitoring will continue to provide call progress information after transfers, forwarding and conference operations. During the life of a call many CEs may participate; some with enabled CE monitors and some without. This service enables a computer to receive call progress information without having to explicitly place CE monitors on all the CEs which have participated in that call.

To enable monitoring use the Monitor Start Functional Service and specify a filter. The server will respond with either a positive acknowledgement confirming the enabled monitor and associated filter or with an error response. If the monitor is successful enabled events will be sent by the server until the client or server terminates the enabled monitor using the Monitor Stop Service. Call progress events are generated for all calls which arrive at the specified CE after the monitor start functional service is acknowledged and for calls which are at the specified CE at the time of the acknowledgement. Call Progress events which occurred before the start monitor acknowledgement are not reported. These events are delivered to the client through the Event Report Service or as discrete Functional Services depending upon the implementation. If the server does not support filtering the response will show all events as filtered, i.e. the specified events will not be sent. Once a monitor has been enabled, the associated filter may be modified by the Change Monitor Filter Functional Service.

6.5.1 **Functional Service name:** Change Monitor Filter Service

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request that a previously specified filter for an existing monitor be changed.

c) Parameters

REQUEST:

monitorID	MonitorID
callFilter	CallFilter
agentFilter	AgentFilter
privateData	TASCPrivateData

RESPONSE:

Success:

callFilter	CallFilter	OPTIONAL
agentFilter	AgentFilter	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Error:

error	ErrorGroups

Parameter description

monitorID	indicates the monitor whose filter is to be changed.
callFilter	indicates the call progress events to be filtered out by the server and therefore not sent to the client. From one event to all available call progress events may be specified. Within a response this parameter indicates the actual filter set by the server which may differ from the requested filter.
agentFilter	indicates the agent events to be filtered out by the server and therefore not sent to the client. From one event to all available agent events may be specified. Within a response this parameter indicates the actual filter set by the server which may differ from the requested filter.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The client has an active monitor using the specified monitorID.

Postconditions:

– The client expects event reports as indicated by the filters contained within the response to the service request.

Operational rules

After sending the request to the switch, the computer waits for a success response or an error response. A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid. The server may respond with a CallFilter or AgentFilter which differs from the requested filter.

- **error handling**

On receipt of an error event information may continue for the specified monitorID as defined by the filter previously activated.

e) Server (switch)

- **normal procedure**

Preconditions:

– The server has an active monitor using the specified monitorID.

Postconditions:

– The filters are set as specified in the success response.

Operational rules

After receiving the request from the client, the server performs the following actions:

– Verifies the specified monitorID.

– Determines the filter mask possible that coincides with the request.

– Changes the specified filter.

– The server notifies the enabled filter to client.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The computer is able to determine whether the server can support filtering and be able to determine the filter types supported within that service.

– The computer is able to determine whether implicit monitoring is supported.

– Monitor Identifiers can be interrogated to identify the associated CE and their monitor type.

6.5.2 **Functional Service name:** Monitor Start Service

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to generate call progress events and agent events for a call, CE or for one or more calls involving a CE. The TASC object, Call View or CE, to be monitored will be specified in the service request.

The filter specified in the Service Request indicates event reports that are to be filtered out by the server (switch) and not sent to the client (computer).

c) Parameters

REQUEST:

monitorCE	CEID	
monitorCall	CVID	
monitorType	MonitorType	OPTIONAL
callFilter	CallFilter	OPTIONAL
agentFilter	AgentFilter	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

monitorID	MonitorID	
callFilter	CallFilter	OPTIONAL
agentFilter	AgentFilter	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Error:

error	ErrorGroups

Parameter description

monitorCE	identifies the CE to be monitored.
monitorCall	identifies the call to be monitored.
monitorType	identifies the type of monitor requested; Call or CE.
callFilter	indicates the basic call progress events to be filtered out by the server and therefore not sent to the client. From one event to all available call progress events may be specified. Within a response this parameter indicates the actual filter set by the server which may differ from the requested filter.
agentFilter	indicates the agent events to be filtered out by the server and therefore not sent to the client. From one event to all available agent events may be specified. Within a response this parameter indicates the actual filter set by the server which may differ from the requested filter.
monitorID	indicates a value that is unique within the association for the duration of the monitor and that can be used to relate subsequent event reports to the monitor start service request that initiated them. It is also used to correlate Monitor Stop and Change Monitor requests with the original monitor start request.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The client wishes to receive event information for the specified object.

Postconditions:

– Event information is provided to the client.

Operational rules

After sending the request to the switch, the computer waits for a success response or an error response. A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid and that the requested monitor has been enabled.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The specified CE or CV is validated.

2) The specified CE may have calls already "active" at the specified CE.

Postconditions:

– Event reports begin to flow as requested in the filter returned to the client.

Operational rules

On receipt of the Service request, the server performs the following actions:

– The specified CE identifier or CV identifier is validated.

– The specified CE can support monitoring.

– Verifies that the specified monitor filter can be supported and notifies the client of the activated filter which may differ from that requested.

– If the client does not specify a filter in the service request then the server will generate all event reports, i.e. no event reports are filtered out. The filter is optional in both client and server and so the client must be prepared to receive event reports that it had requested to be filtered out.

– If the MonitorType parameter is not requested in the request then the type of monitoring is selected by the server. This standard does not specify the default selected by the server.

– The specified monitor and its associated filter are initiated at the specified object.

– If privateData is present, they are interpreted; if not understood, they are discarded.

– The server will terminate the monitor if the object being monitored ceases to exist or if the monitored object leaves the TASC operation domain.

– A server which does not support all event reports or does not support filtering may accept Requests even if the requested filter cannot be provided. In these circumstances the response shall indicate the actual set of events that will be provided, i.e. a server which does not support filtering shall respond to requests showing all events unfiltered. Events which cannot be supported should be shown as filtered in the response.

- **error handling**

Not applicable.

f) Management requirements

TASC Management should ensure that:

- The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
- The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
- The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.
- The computer is able to determine whether the server can support filtering and be able to determine the filter types supported within that service.
- The computer is able to determine whether implicit monitoring is supported.
- The calls currently active at a CE can be determined.
- The CEs participating within a call can be determined.
- Monitor Identifiers can be interrogated to identify the associated CE and their monitor type.

6.5.3 Functional Service Name: Monitor Stop Service

a) Relationship: Computer ↔ Switch

b) General description

This Functional Service is used by either the server or the client to cancel a previously initiated Monitor Start Service. This is a bidirectional functional service.

c) Parameters

REQUEST:

monitorID	MonitorID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

privateData	TASCPrivateData	OPTIONAL

Error:

error	ErrorGroups

Parameter description

monitorID	indicates the monitorID value that was provided in the original Monitor Start Service response and correlates the service request with events and the Monitor Stop Service request.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer/switch)

- **normal procedure**

Preconditions:

– The client has an active monitor using the specified monitorID.

Postconditions:

– The client removes all references to the specified monitorID. The monitorID is now free.

Operational rules

After sending the request to the server, the client waits for a success response or an error response. A success response indicates, that the server has received the request, that the message was correct and interpreted, that the identifiers are valid and that the requested monitor has been disabled.

- **error handling**

On receipt of an error event, information may continue for the specified monitorID where applicable.

e) Server (computer/switch)

- **normal procedure**

Preconditions:

– The server has an active monitor using the specified monitorID.

Postconditions:

– The server removes all references to the specified monitorID.

Operational rules

After receiving the request from the client, the server performs the following actions:

– Verifies the specified monitorID.

– Disables the specified monitor.

– The server acknowledges the removal of the monitor to the client.

– If the server is the switch, then it sends no further events for the specified monitorID to the client.

– Further monitor requests may reuse the monitorID.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.

– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.

– The monitorID currently in effect can be determined, and potentially cleared.

– Monitor Identifiers can be interrogated to identify the associated CE and their monitor type.

6.6 Snapshot Functional Services

6.6.1 **Functional Service name:** Snapshot Call

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to provide information about the specified call independently of the monitor function.

The nature of the Snapshot Call Functional Service is to obtain status and return it in a response. This does not affect the states of any objects in the switch.

c) Parameters

REQUEST:

callToSnapshot	CVID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

The success response shall include the following parameters for every endpoint in the call:

endpointCE	CEID
snapshotCallData	CVStates
snapshotCPData	CPStates

The success response may also include the following parameter:

privateData	TASCPrivateData	OPTIONAL

Error:

Parameter description

callToSnapshot	identifies the call to snapshot.
endpointCE	identifies a visible CE for an endpoint in the call to snapshot.
snapshotCallData	identifies the call view state for an endpoint in the call to snapshot.
snapshotCPData	identifies the call party state for an endpoint in the call to snapshot.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that a “Snapshot Call” request has to be sent.

Postconditions:

– The computer processes the response from the switch.

Operational rules

After having sent the request for a snapshot call to the switch, the computer waits for a positive acknowledgement or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid. It includes information of the current status of the call, the CEs involved in that call and the call states.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The call to snapshot is known.
2) Allowable basic call view states: All.

Postconditions:

1) The switch performs the actions to identify all visible endpoints in the identified call and the call and CP states for each endpoint.
2) Resulting basic call view states: All.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The call identifier is validated.
– The CEs involved in the call are identified.
– The call states at each endpoint in the call are determined.
– The CP states at each endpoint in the call are determined.
– If "Private Data" is present, they are interpreted; if not understood, discarded.
– A success response is generated.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.
– CE identifiers can be verified as valid, active and not barred.
– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).
– A history (timed and dated) and source of changes (e.g. from network management) to the call forwarding settings of a CE can be determined.
– The features supported by a CE can be determined.

6.6.2 **Functional Service name:** Snapshot CE

a) Relationship: Computer → Switch

b) General description

This Functional Service is used to request the switch to provide information about calls associated with the specified CE independently of the monitor function.

The nature of the Snapshot CE Functional Service is to obtain status and return it in a response. This does not affect the states of any objects in the switch.

c) Parameters

REQUEST:

cEToSnapshot	CEID	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:

The success response shall include the following parameters for every call at the identified CE:

associatedCall	CVID
snapshotCallData	CVStates
snapshotCPData	CPState

The success response may also include the following parameter:

privateData	TASCPrivateData	OPTIONAL

Error:

Parameter description

cEToSnapshot	identifies the CE to snapshot.
associatedCall	identifies a call at the specified CE.
snapshotCallData	identifies the call view states visible for all endpoints in the associated call.
snapshotCPData	identifies the call party states visible for all endpoints in the associated call.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer)

- **normal procedure**

Preconditions:

– The computer has determined that a “Snapshot CE” Functional Service request has to be sent.

Postconditions:

– The computer processes continue.

Operational rules

After having sent the request for a snapshot CE to the switch, the computer expects a success response or an error response.

A success response indicates, that the switch has received the request, that the message was correct and interpreted, that the identifiers are valid. It includes information of all the calls that are present at the specified CE and the call states.

- **error handling**

Not applicable.

e) Server (switch)

- **normal procedure**

Preconditions:

1) The CE to snapshot is known.
2) Allowable basic call view states: All.

Postconditions:

– Resulting basic call view states: All.

Operational rules

On receipt of the service request, the switch performs the following actions:

– The CE identifier is validated.
– The calls associated with the identified CE are identified.
– The call state for each visible endpoint in the call is determined.
– The party state for each visible endpoint in the call is determined.
– If “Private Data” is present, they are interpreted; if not understood, discarded.
– A success response is generated.

- **error handling**

An error response should be generated. Details are implementation specific.

f) Management requirements

TASC Management should ensure that:

– The CE identifiers relate to the relevant pieces of physical equipment, i.e. the computer is kept informed of any changes to the identifier for equipment in the operation domain.
– The computer is able to determine, and possibly affect, any timer values that the switch uses to determine when a TASC FS is deemed completed.
– The switch is able to determine, and possibly affect, any timer values that the computer uses to determine when expecting a reply to a TASC FS request.
– CE identifiers can be verified as valid, active and not barred.
– The computer is able to determine the type (i.e. call characteristics) of calls which the terminal can support (if this information is available).
– A history (timed and dated) and source of changes (e.g. from network management) to the call forwarding settings of a CE can be determined.
– The features supported by a CE can be determined.

6.7 **Functional Service name:** Event Report

a) Relationship: Switch → Computer

b) General description

Event Report messages shall be sent from client (switch) to server (computer) when a monitor request has been positively acknowledged and a TASC reportable event has occurred. Event Reports reflect changes in object states within the TASC domain and may be visible via monitors on CEs or calls.

Note that events may be delivered to the client through the Event Report Service or as discrete Functional Services depending upon the implementation.

c) Parameters

REQUEST:

monitorID	Monitor ID	
eventType	EventTypeID	
eventInfo	EventInfo	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:	Not applicable
Error:	Not applicable

Parameter description

monitorID	uniquely identifies the monitor request that resulted in the Event Report. This parameter shall allow the differentiation of Event Reports resulting from multiple monitors.
eventType	uniquely identifies the event being sent, i.e. a call cleared call progress event.
eventInfo	parameters as defined in each discrete event.
privateData	identifies custom data specific to an implementation or application.

d) Client (switch)

- **normal procedure**

Preconditions:

– The client has an active monitor using the specified monitorID.

Postconditions: None.

Operational rules

After sending the request to the computer, the switch does not expect any response or error message. Event information is notified to the computer using the event report functional service for each state change that occurs at the monitored CE, in accordance with the specified filter.

- **error handling**

Not applicable.

e) Server (computer)

- **normal procedure**

Preconditions:

– The server has an active monitor using the specified monitorID.

Postconditions: None.

Operational rules

None.

- **error handling**

Not applicable.

f) Management requirements

TASC Management should ensure that:

– The computer is able to determine whether the server can support filtering and be able to determine the filter types supported within that service.

– The monitor types and filters applicable to a CE can be determined.

– Monitor Identifiers can be interrogated to identify the associated CE and their monitor type.

6.8 Application Status Services

6.8.1 Functional Service name: Application Activity Check

a) Relationship: Computer ↔ Switch

b) General description

This Functional Service is a request for a peer application to indicate messages are being received over the association. The request may be issued by the switch or computer.

c) Parameters

REQUEST:		
privateData	TASCPrivateData	OPTIONAL
RESPONSE:		
Success:		
application OK	BOOLEAN	
privateData	TASCPrivateData	OPTIONAL
Error:	ErrorGroups	

Parameter description

application OK	true indicates the peer application is operating normally; false indicates some type of trouble condition.
privateData	identifies custom data specific to an implementation or application.
error	identifies the response error.

d) Client (computer/switch)

- **normal procedure**

Preconditions:

– There appears to be a problem with the peer application (e.g. the peer has not communicated for some time).

Postconditions:

– Status of peer application determined.

Operational rules

This FS should be invoked when there is suspicion that there is trouble with the peer application (e.g. the peer has not communicated for some time).

- **error handling**

Not applicable.

e) Server (switch/computer)

- **normal procedure**

Preconditions: None.

Postconditions:

– Status returned to peer.

Operational rules

One application queries or polls another application. The responding application returns a true indication if the peer application is operating normal. A false indication indicates trouble which may require corrective action. If the other application does not respond within a specific time–frame, the polling application may retry or take corrective action:

– return true if operation is normal;

– return false if there is trouble which requires corrective action.

- **error handling**

Corrective actions are not addressed in TASC and are left to individual implementations.

f) Management requirements

TASC Management should ensure that:

– needs to specify a time value for which the service is needed (i.e. a time period in which no communication has occurred from the peer).

6.8.2 Functional Service name: Application Congestion Report

a) Relationship: Computer ↔ Switch

b) General description

Although a peer application is working well, a switch or computer application may be congested. In such a case, the application uses this Application Congestion Report to indicate the situation to the peer application. Furthermore, this Functional Service may also be used to report that congestion has recovered.

c) Parameters

REQUEST:

congestionLevel	CongestionLevel	
privateData	TASCPrivateData	OPTIONAL

RESPONSE:

Success:	Not applicable
Error:	Not applicable

Parameter description

congestionLevel	identifies one of the following congestion levels: “Red” = congestion, “Green” = No congestion, “Amber” = some congestion.
privateData	identifies custom data specific to an implementation or application.

d) Client (computer/switch)

- **normal procedure**

Preconditions:

– A significant change in congestion level requires reporting.

Postconditions: None.

Operational rules

After having sent the request, the client application expects the server application to decrease the number of messages in a proprietary manner.

- **error handling**

Not applicable.

e) Server (switch/computer)

- **normal procedure**

Preconditions: None.

Postconditions:

– The server acts upon congestion level.

Operational rules

On receipt of the service request, the server will attempt to reduce the number of messages sent to the client.

- **error handling**

Not applicable.

f) Management requirements

Not applicable.

6.9 Functional Service name: Send Private Data

a) Relationship: Computer ↔ Switch

b) General description

This Functional Service is used to convey “private data” to its peer.

c) Parameters

REQUEST:

privateData TASCPrivateData

RESPONSE:

Success:

privateData TASCPrivateData

Error: Not applicable

Parameter description

privateData identifies custom data specific to an implementation or application.

d) Client (computer/switch)

- **normal procedure**

Preconditions: None.

Postconditions: None.

Operational rules

Implementation specific.

- **error handling**

Implementation specific.

e) Server (switch/computer)

- **normal procedure**

Preconditions: None.

Postconditions: None.

Operational rules

Implementation specific.

- **error handling**

Implementation specific.

f) Management requirements

Not applicable.

7 Event descriptions

7.1 Agent events

The agent events are concerned with the reporting of agent state-change information as indicated in the agent model (see Recommendation Q.1301: "TASC Architecture").

Events can be reported either using discrete events or using the Event Report Functional Service.

7.1.1 **Event name:** Agent Busy

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent-busy operation has been invoked. The agent is busy with a distributed call.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE on whose behalf the associated agent has indicated "Agent Busy".
agent	the agent's identifier.
agentGroup	identifies the group the agent is logged into.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent state: AgentReady.

Postconditions:

– Resulting agent state: AgentBusy.

7.1.2 Event name: Agent Logged Off

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent log-off operation has been invoked. The agent is no longer associated with the line CE and the agent is no longer able to accept distributed calls.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter descriptions

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE the agent has logged-off from.
agent	the agent's identifier.
agentGroup	identifies the group the agent has logged out of.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent states: AgentReady, AgentNotReady, AgentBusy, AgentWorkingAfterCall.

Postconditions:

– Resulting agent state: AgentNull.

7.1.3 **Event name:** Agent Logged On

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent log-on operation has been invoked. The agent is logged on at the CE and group indicated. The agent is not yet ready to accept distributed calls.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that the agent can accept calls which are not from the distributor function, i.e. from another agent.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE the agent is being logged onto.
agent	the agent's identifier.
agentGroup	identifies the group the agent is logging into.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent state: AgentNull.

Postconditions:

– Resulting agent state: AgentNotReady.

7.1.4 **Event name:** Agent Not Ready

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent-not-ready operation has been invoked. The agent is not ready to receive distributed calls.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that an agent can accept calls which are not from the distributor function, i.e. from another agent.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE on whose behalf the associated agent has indicated "Agent Not Ready".
agent	the agent's identifier.
agentGroup	identifies the group the agent is logged into.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent states: AgentReady, AgentBusy, AgentWorkingAfterCall.

Postconditions:

– Resulting agent state: AgentNotReady.

7.1.5 Event name: Agent Ready

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent ready operation has been invoked. The agent is ready to receive distributed calls.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE on whose behalf the associated agent has indicated "Agent Ready".
agent	the agent's identifier.
agentGroup	identifies the group the agent is logged into.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent states: AgentNotReady, AgentBusy, AgentWorkingAfterCall.

Postconditions:

– Resulting agent state: AgentReady.

7.1.6 Event name: Agent Working After Call

a) Relationship: Switch → Computer

b) General description

This event indicates that the agent-after-call-work operation has been invoked. The agent is occupied with after-call-work and is not ready to receive distributed calls.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that the agent can accept calls which are not from the distributor function, i.e. from another agent.

c) Parameters

monitorID	MonitorID	
agentLineCE	CEID	
agent	AgentID	OPTIONAL
agentGroup	AgentGroupID	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
agentLineCE	identifies the line CE on whose behalf the associated agent has indicated "Agent Working After Call".
agent	the agent's identifier.
agentGroup	identifies the group the agent is logged into.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable agent state: AgentBusy.

Postconditions:

– Resulting agent state: AgentWorkingAfterCall.

7.2 Call progress events

The call progress events are concerned with the reporting of call state changes.

General conditions

1) Events are generated as a result of monitoring.

2) Events can be reported either using discrete events or using the Event Report Functional Service.

3) Events do not have a response (success or error).

7.2.1 **Event name:** Call Arrived

a) Relationship: Switch → Computer

b) General description

This event indicates the arrival of an incoming call and the identification of a destination line CE for the call.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that the call has not yet been delivered to that CE.

c) Parameters

monitorID	MonitorID	
arrivedCall	CVID	
destinationCE	CEID	
callingCE	CEID	OPTIONAL
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
arrivedCall	identifies the call that arrived.
destinationCE	identifies the CE to receive the call.
callingCE	identifies the CE that is calling.
calledCE	identifies the CE that was called; this may be different from the destinationCE, e.g. because of the invocation of a feature.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view state: Null.

Postconditions:

– Resulting basic call view state: Arrived.

7.2.2 **Event name:** Call Cleared

a) Relationship: Switch → Computer

b) General description

This event indicates a call has ended. All resources associated with the call have been released.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that this event is sent for every monitored CE participating in the call.

c) Parameters

monitorID	MonitorID	
clearedCall	CVID	
clearingCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
clearedCall	identifies the call that was cleared.
clearingCE	identifies the CE which caused the call to be cleared.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Pending, Originated, Delivered, Established, Failed, Arrived, Received.

Postconditions:

– Resulting basic call view state: Null.

7.2.3 **Event name:** Call Conferenced

a) Relationship: Switch → Computer

b) General description

This event indicates that two calls have been merged into one call with no CEs removed from the resulting call in the process.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
heldCall	CVID	
activeCall	CVID	
conferenceCE	CEID	
addedCE	CEID	
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
heldCall	identifies the existing call which was held at the conference CE prior to the conference.
activeCall	identifies the added call that was active at the conference CE prior to the conference.
conferenceCE	identifies the CE for which the conference was initiated.
addedCE	identifies the CE that was added to the call.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Originated, Delivered or Established.

2) Allowable CP states: CP1 Held, CP2 Active.

Postconditions:

1) Resulting basic call view state: Unchanged.

2) Resulting CP state: Active.

7.2.4 Event name: Call Delivered

a) Relationship: Switch → Computer

b) General description

This event indicates that an outgoing call has been assigned to a destination line CE and the destination line CE is "alerting".

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that this is equivalent to "listening to ringback" or "ringing tone".

c) Parameters

monitorID	MonitorID	
deliveredCall	CVID	
alertingCE	CEID	
callingCE	CEID	
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
deliveredCall	identifies the call that is alerting.

alertingCE	identifies the CE at which the call is alerting.
callingCE	identifies the CE that is calling.
calledCE	identifies the CE that was called; this may be different from the alertingCE, e.g. because of the invocation of a feature.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view state: Originated.

Postconditions:

– Resulting basic call view state: Delivered.

7.2.5 Event name: Call Diverted

a) Relationship: Switch → Computer

b) General description

This Event Report shall indicate that a call has been diverted from the called CE to another CE. The called CE is no longer involved in the call.

c) Parameters

newCalledCE	CEID	
callToBeDiverted	CVID	
calledCE	CEID	
diversionType	DiversionInformation	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

diversionType	indicates the type of diversion requested.
newCalledCE	identifies the CE to which the call is to be diverted.
callToBeDiverted	identifies the call to be diverted.
calledCE	identifies the CE which was originally called.
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Received, Distributed, Established.

Postconditions:

– Resulting basic call view states: Null (calledCE), Arrived (newcalledCE).

7.2.6 Event name: Call Established

a) Relationship: Switch → Computer

b) General description

This event indicates that a call has been established at a line CE and communication may take place between CEs.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
establishedCall	CVID	
answeringCE	CEID	
callingCE	CEID	OPTIONAL
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
establishedCall	identifies the call that is established at the CE.
answeringCE	identifies the CE that answers the call.
callingCE	identifies the CE that is calling.
calledCE	identifies the CE that was called; this may be different from the answeringCE, e.g. because of the invocation of a feature.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Delivered, Originated, Arrived, Received.

Postconditions:

– Resulting basic call view state: Established.

7.2.7 Event name: Call Failed

a) Relationship: Switch → Computer

b) General description

This event indicates that a call cannot be completed. Normal call progress has been aborted.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
failedCall	CallID	
callingCE	CEID	OPTIONAL
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
failedCall	identifies the call that did not complete.
callingCE	identifies the CE that was calling.
calledCE	identifies the CE that was being called.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Received, Arrived, Pending, Originated, Delivered.

Postconditions:

– Resulting basic call view state: Failed.

7.2.8 Event name: Call Held

a) Relationship: Switch → Computer

b) General description

This event indicates that the server (switch) has detected that an existing call has been temporarily interrupted at one of the CEs on the call.

If this event is reported using the Event Report Functional Service, the “monitorID” and “private data” may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
heldRelation	CPID	
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
heldRelation	identifies the CP which is to be held.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Pending, Originated, Delivered, Received, Established.

2) Allowable CP state: Active.

Postconditions:

1) Resulting basic call view state: Unchanged.

2) Resulting CP state: Held.

7.2.9 Event name: Call Originated

a) Relationship: Switch → Computer

b) General description

This event indicates the completed collection of dialled digits and a call, rather than a feature, has been originated to another line CE.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
originatedCall	CVID	
callingCE	CEID	
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
originatedCall	identifies the call that is being originated.
callingCE	identifies the CE that is originating the call.
calledCE	identifies the CE that is being called.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Null, Pending.

Postconditions:

– Resulting basic call view state: Originated.

7.2.10 Event name: Call Received

a) Relationship: Switch → Computer

b) General description

This event indicates an incoming call has been assigned to a line CE and the line CE "alerting".

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that this is equivalent to the phone "ringing".

c) Parameters

monitorID	MonitorID	
receivedCall	CVID	
alertingCE	CEID	
callingCE	CEID	OPTIONAL
calledCE	CEID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
receivedCall	identifies the call that was received.
alertingCE	identifies the CE at which the call is alerting.
callingCE	identifies the CE that is calling.
calledCE	identifies the CE that was called; this may be different from the alertingCE, e.g. because of the invocation of a feature.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

– Allowable basic call view states: Null, Arrived.

Postconditions:

– Resulting basic call view state: Received.

7.2.11 Event name: Call Retrieved

a) Relationship: Switch → Computer

b) General description

This event indicates that the server (switch) has detected that a previously "held" call at a CE has been retrieved.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
retrievedRelation	CPID	
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
retrievedRelation	identifies the CP which is to be retrieved.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Pending, Originated, Delivered, Received, Established.
2) Allowable CP state: Held.

Postconditions:

1) Resulting basic call view state: Unchanged.
2) Resulting CP state: Active.

7.2.12 Event name: Call Transferred

a) Relationship: Switch → Computer

b) General description

This event indicates that an existing call has been transferred to another CE and that the CE requesting the transfer has been dropped from the call. The transferring CE does not appear in any future event reports for the new call.

If this event is reported using the Event Report Functional Service, the “monitorID” and “private data” may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
transferringCE	CEID	
transferredToCE	CEID	
previousHeldRelation	CPID	OPTIONAL
previousActiveRelation	CPID	OPTIONAL
newRelations	CPID	OPTIONAL
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
transferringCE	identifies the line CE from which the transfer was initiated.

transferredToCE	identifies the line CE to which the call has been transferred.
previousHeldRelation	identifies the CP which was previously held, especially the call view identifier.
previousActiveRelation	identifies the CP which was previously active, especially the call view identifier.
newRelations	lists the CPs participating in the new call.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Originated, Delivered, Established, Received.

2) Allowable CP states: CP1 Held, CP2 Active.

Postconditions:

1) Resulting basic call view state: Unchanged.

2) Resulting CP states: CP1 Active, CP2 Active.

7.2.13 Event name: CP Dropped

a) Relationship: Switch → Computer

b) General description

This event indicates that a CE was dropped from a call. It is not used to indicate that a transferring CE has left the call in the act of transfer.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

c) Parameters

monitorID	MonitorID	
droppedRelation	CPID	
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
droppedRelation	identifies the CP that was dropped from the call.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) Allowable basic call view states: Originated, Delivered, Established, Failed, Arrived, Received.

2) Allowable CP states: Active, Held.

7.2.14 Event name: Service Pending

a) Relationship: Switch → Computer

b) General description

This event indicates that a line CE is attempting to place a call to another line CE or attempting to invoke a feature or supplementary service. This is equivalent to a line CE going off-hook.

If this event is reported using the Event Report Functional Service, the "monitorID" and "private data" may be omitted, since they are contained in the Event Report Functional Service.

Note that this event is not sent for *en-bloc* dialling which encompasses headsets.

c) Parameters

monitorID	MonitorID	
initiatedCall	CVID	
initiatingCE	CEID	
cause	TransitionCause	OPTIONAL
privateData	TASCPrivateData	OPTIONAL

Parameter description

monitorID	used to link the event to the enabled monitor.
initiatedCall	identifies the initiated call.
initiatingCE	identifies the CE initiating the call.
cause	identifies the cause for the event (as appropriate to the Cause List).
privateData	identifies custom data specific to an implementation or application.

d) Server (switch)

- **normal procedure**

Preconditions:

1) This event is only sent when the line CE goes off-hook and prior to digits being collected.

2) Allowable basic call view state: Null.

Postconditions:

– Resulting basic call view state: Pending.

7.3 Cause List

Causes can be used to clarify information provided in Event Reports. The following cause list shows the causes that are meaningful for Event Reports and may appear in any call or agent related Event Report where they make sense.

Cause	*indicates that*
Alternate	a call has been alternated. Among others, this cause makes sense in the following event reports: Call Held, Call Retrieved, Call Established, Call Failed.
Authorization Failure	the calling CE is restricted from establishing a call to the destination CE or route. Among others, this cause makes sense in the following event report: Call Originated.
Busy	a call encountered a busy or unavailable CE. Among others, this cause makes sense in the following event report: Call Failed.
Call Back	the Call Back feature has been invoked. Among others, this cause makes sense in the following event reports: Service Pending, Call Cleared, Call Originated, Call Delivered.
Call Cancelled	the call has been terminated before the associated CE has gone on-hook. Among others, this cause makes sense in the following event reports: Service Pending, Call Originated, Call Delivered, Call Cleared.
Call Forwarded	the call has been redirected via a Call Forwarding feature. Among others, this cause makes sense in the following event reports: Call Received, Call Diverted, Call Failed.
Call Forwarded – Immediate	the call has been redirected via a Call Forwarding feature, set for all conditions. Among others, this cause makes sense in the following event reports: Call Received, Call Diverted, Call Failed.
Call Forwarded – Busy	the call has been redirected via a Call Forwarding feature, set for a busy endpoint. Among others, this cause makes sense in the following event reports: Call Received, Call Diverted, Call Failed.
Call Forwarded – No Answer	the call has been redirected via a Call Forwarding feature, set for an endpoint that does not answer. Among others, this cause makes sense in the following event reports: Call Received, Call Diverted, Call Held, Call Failed.
Call Not Answered	the call was not answered because a timer elapsed. Among others, this cause makes sense in the following event reports: Call Cleared, Call Diverted, Call Failed.

Call Pickup	the call was redirected via a Call Pickup feature. Among others, this cause makes sense in the following event reports: Call Diverted, Call Established.
Camp On	a Camp On feature was invoked or has matured. Among others, this cause makes sense in the following event reports: Call Received, Call Failed.
Destination Alerting	the terminating CE in a call is alerting. Among others, this cause makes sense in the following event reports: Call Originated, Call Arrived.
Destination Answered	the terminating CE in a call has answered (e.g. off-hook). Among others, this cause makes sense in the following event reports: Call Originated, Call Delivered, Call Arrived, Call Received.
Destination Not Obtainable	the call could not reach the destination. Among others, this cause makes sense in the following event reports: Call Cleared, Call Failed.
Distributed	the call was distributed by an ACD or hunt group. Among others, this cause makes sense in the following event reports: Call Cleared, Call Delivered, Call Established, Call Diverted, Call Failed.
Do Not Disturb	the call encountered a Do Not Disturb condition. Among others, this cause makes sense in the following event reports: Call Cleared, Call Diverted, Call Failed.
En-bloc Call Set-up	addressing information provided and a new call has been originated. Among others, this cause makes sense in the following event reports: Call Pending.
Entering Distribution	the call was delivered to a distributor function (ACD). Among others, this cause makes sense in the following event reports: Call Delivered, Call Established.
Failure Treatment Completed	switch has completed providing feedback to the end user concerning the failure. Among others, this cause makes sense in the following event reports: Call Originated, Call Failed.
Host Initiated Clearing	call cleared as a result of a host's request. Among others, this cause makes sense in the following event reports: Call Pending, Call Delivered, Call Established, Call Failed, Call Received.
Information Analysis Failure	the destination cannot be determined from the addressing information provided. Among others, this cause makes sense in the following event report: Call Originated.

Information Collection Complete	all addressing information has been received and the call can proceed. Among others, this cause makes sense in the following event report: Call Pending.
Information Collection Timeout	addressing information was not provided within the time allowed. Among others, this cause makes sense in the following event reports: Call Pending.
Invalid Account Code	the account code is invalid. Among others, this cause makes sense in the following event report: Call Originated.
Incompatible Destination	the call encountered an incompatible destination. Among others, this cause makes sense in the following event reports: Call Cleared, Call Diverted, Call Failed.
Key Operation	the reported event occurred at a bridged or twin CE. This cause makes sense in the all call related event reports.
Lockout	the call encountered inter-digit time-out while dialling. Among others, this cause makes sense in the following event report: Call Failed.
Maintenance	the call encountered a facility or endpoint in a maintenance condition. Among others, this cause makes sense in the following event reports: Call Cleared, Call Failed.
Network Congestion	the call encountered a congested network. Among others, this cause makes sense in the following event report: Call Failed.
Routing Failure Unspecified	unable to route the call; the call encountered an unspecified problem. Among others, this cause makes sense in the following event report: Call Originated.
Network Not Obtainable	the call could not reach a destination network. Among others, this cause makes sense in the following event report: Call Failed.
Network Signal	the event was provided as result of a network signal. Among others, this cause makes sense in the following event reports: Call Received, Call Established, Call Transferred, Call Conferenced, Call Held, Call Failed.
Network Tone	the call encountered a network condition. When this occurs, the network usually provides a Network Tone to indicate that a request was not recognizable. Among others, this cause makes sense in the following event report: Call Failed.
New Call	the call has not yet been redirected. Among others, this cause makes sense in the following event reports: Call Originated, Call Delivered, Call Established, Call Conferenced, Call Transferred.

No Available Agents	the call could not access any agent. Among others, this cause makes sense in the following event reports: Call Diverted, Call Failed.
Normal Call Clearing	the call cleared in a normal way. Among others, this cause makes sense in the following event report: Call Cleared.
Number Changed	the called number has been changed to a new number. Among others, this cause makes sense in the following event report: Call Failed.
Overflow	the call overflowed a queue, group, or target. Among others, this cause makes sense in the following event reports: Call Diverted, Call Cleared, Call Failed.
Recall	the call is alerting due to a time-out associated with a feature that failed to complete or that anticipated further action from the user. Among others, this cause makes sense in the following event reports: Call Delivered, Call Established, Call Diverted, Call Transferred, Call Conferenced, Call Held, Call Retrieved, Call Failed.
Redirected	the call has been redirected. Among others, this cause makes sense in the following event reports: Call Delivered, Call Diverted, Call Transferred, Call Failed.
Resources Not Available	resources were not available. Among others, this cause makes sense in the following event reports: Call Cleared, Call Failed.
Transfer	a Transfer is in progress or has occurred. Among others, this cause makes sense in the following event reports: Call Transferred, Call Held, Call Retrieved, Call Received, Call Established.