Network Working Group
Request for Comments: 3064
Category: Informational

B. Foster
Cisco Systems

February 2001

MGCP CAS Packages

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document contains a collection of media gateway Channel Associated Signaling (CAS) packages for R1 CAS, North American CAS, CAS PBX interconnect as well as basic FXO support. Included are six packages. The "MS" package covers MF single stage dialing trunks. This includes wink start and immediate start PBX DID/DOD trunks as well as basic R1 and Feature Group D (FGD) Terminating protocol [3]. The "DT "package covers immediate start and basic DTMF and dial-pulse trunks and the "BL" package covers the interface to a basic PBX (digital or FXS interface). In addition to the Terminating protocol, there are three other FGD protocols described in [3]. These include EAIN and EANA which is covered by the enclosed "MD" package and the Operator Service Signaling protocol which is handled by the "MO" package. Support for basic FXO interconnect is provided by "DO" package.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119.

IESG Note:

This document is being published for the information of the community. It describes a protocol that is currently being deployed in a number of products. Implementers should be aware of developments in the IETF Megaco Working Group and ITU SG16 who are currently working on a potential successor to this protocol.

Table of Contents

1.0.Introduction

1.1. Functional Partitioning

There are a number of different possible approaches for partitioning the functional responsibility between the Call Agent and the Media Gateway:

* The Call Agent takes all of the responsibility for the CAS state machine giving the media gateway detailed commands

* The media gateway contains the CAS state machine and provides an abstract interface to the Call Agent

Timing requirements of CAS protocols often involve reacting within time intervals measured in tens of milliseconds which makes direct control of timing impossible. The approach used here is to allow the media gateway to handle low level CAS protocol and timing details where at all possible and have the Call Agent involved only whenever higher level processing is required.

Taking this approach, the ideal situation would be to allow the Call Agent to treat as many CAS protocols in a similar way, leaving the details to the media gateway. Example: for an incoming MF trunk that involves a single incoming digit string, the Call Agent should not care whether this is a wink start trunk or an immediate start trunk (media gateway should not have to provide the wink-start signal).

Some goals in partitioning responsibility between the media gateway and media gateway:

* Minimize the number of interactions between the Call Agent and the media gateway.

* The media gateway should not have to do digit analysis (e.g., to determine that the incoming digits contain carrier access information). This is a Call Agent’s responsibility.

* Provide some reasonable level of abstraction for the Call Agent so that it can reuse call flows when possible (e.g., Call Agent should not have to differentiate between wink start and immediate start interfaces when only one digit string is involved).

* The media gateway should take care of the CAS protocol (and timeouts) where possible with the Call Agent taking over responsibility where the media gateway leaves off.

Use of Embedded Notifications: Rather than depending on the use of embedded notifications, signals and events were defined that had the specific semantics required. There are two reasons for this:

a) It allows an abstract interface for the Call Agent so that for example, the same incoming call-setup event can be used in the case of MF wink start and MF immediate start trunks, presenting a common interface to the Call Agent even though the semantics at the CAS state-machine level are slightly different (i.e., in the MF wink start case, a wink-start signal is provided reflexively as a result of an incoming seizure, where as in the immediate start case, this is not required).

b) Potential events that might trigger an embedded notification (e.g., the incoming seizure mentioned above) typically needed to be visible to the Call Agent for billing anyway.

This does not say that embedded notifications cannot be used. It simply does not necessitate their use.

Out-pulsing Approach: In order to provide the semantics for outpulsing, special higher level signals (e.g., "sup" for call set-up and "inf" for information) are included that contain the necessary semantics.

Off-hook and On-hook Signals and Events: A higher level view of off-hook and on-hook events is taken in order to make the interface Q.931-like. This provides the advantage that:

* Similar call flows result when dealing with Q.931-based interfaces (e.g., PRI)

* It’s more evident (for ease in debug) when looking at message as to exactly what is going on without having to refer to previous events

1.2. CAS Trunk Types

The following describes the types of trunks supported by the various packages. Configuration of the specific trunk type (e.g., wink start versus immediate start) is done within the Media Gateway (MG) via provisioning facilities outside the scope of MGCP. The Call Agent’s responsibility is to support the particular package (i.e., in general the Call Agent does not have to differentiate between wink start and immediate start, since those differences are taken care of by the MG). However, the Call Agent needs to know which trunks are incoming, outgoing or bi-directional.

1.2.1. "MS" Package

The "MS" package is used for PBX DID/DOD trunks as indicated in the following table. It is also used for incoming or outgoing MF wink start trunks (R1 and FGD Terminating protocol [6]).

Table 1 MF PBX Trunks

Trunk Type	Direction (w.r.t. the gateway)
MF, wink start	Incoming - originate from PBX (the same as FGD terminating protocol)
MF, wink start	Outgoing - terminate on PBX
MF, wink start	Bi-directional
MF, Immediate start	Incoming (originate from PBX)
MF, Immediate start	Outgoing (terminate on PBX)

1.2.2. "DT" Package

DTMF and dial-pulse (DP) trunks (except basic PBX) are covered by the "DT" package along with the DTMF "D" package:

Table 2 DTMF and DP Wink Start and Immediate Start Trunks

Trunk Type	Direction (w.r.t. the gateway)
DTMF, Immediate start, wink start	Incoming (originate from PBX)
DTMF, Immediate start, wink start	Outgoing (terminate on PBX)

1.2.3. "BL" Package

DTMF and dial-pulse (DP) basic PBX trunks are covered by the "BL" package - along with the DTMF "D" package (essentially this is like a "basic line with no features") - either digital or FXS trunk interface:

Table 3 Basic FXS Interface

Trunk Type	Direction (w.r.t. the gateway)
Basic, DTMF and DP, Loop Start	Bi-directional
Basic, DTMF and DP, Ground Start	Bi-directional

1.2.4. "DO" Package

The "DO" package is used for analog FXO loop-start and ground-start analog trunks as indicated in the following table.

Table 4 FXO analog PBX Trunks

Trunk Type	Direction (w.r.t. the gateway)
FXO, loop-start	Bi-directional
FXO, ground-start	Bi-directional

1.2.5. "MD" Package

The MD package provides support for North American MF Feature Group D EANA and EAIN [3], allowing the Media Gateway to be at either the end office, the carrier or the tandem side of the circuit. The CAS Signaling Type column of the following tables is intended to indicate signaling differences that are of common interest to both the Call Agent and Media Gateway. Configuration information that is only of interest to the Media Gateway is not identified.

Table 4 Feature Group D MF Trunks Supported

Trunk Type	Direction (w.r.t. the gateway)
FGD, EANA	Outgoing (End Office to Carrier)
FGD, EANA	Incoming (Carrier to End Office)
FGD, EAIN	Outgoing (End Office to Carrier)
FGD, EAIN	Incoming (Carrier to End Office)

Note that EANA and EAIN signaling may be requested on the same trunk on a call-by-call basis.

1.2.6. "MO" Package

The "MO" package is used for FGD Operator Services Signaling, outgoing trunks only. Feature Group C can also be supported by the same interface.

2.0. Event Packages

This section defines the event packages. The terms "signal" and "event" are used to differentiate a command from a Call Agent to a Media Gateway ("signal") from an "event" that is detected by the Media Gateway and then is "Notified" to the Call Agent.

Each package definition includes a package name, plus the event name codes and the definitions for each of the events in the package. In the tables of events/signals for each package, there are five columns:

* Code The package unique event code used for the event/signal.

* Description A short description of the event/signal.

* Event An "x" appears in this column if the event can be Requested by the Call Agent. Alternatively, one or more of the following symbols may appear:

  - "P" indicating that the event is persistent,

  - "S" indicating that the event is an event-state that may be audited,

- "C" indicating that the event/signal may be detected/applied on a connection. If "C" is associated with an event, this refers to an event that can occur on the media stream. However, "C" may also be associated with a signal (in the signal column), the signal can be requested to sent over a connection.

Note that the intent of being able to audit state ("S") in an event in the following packages is to answer one of the two questions:

1. Has a call been initiated on this line/trunk? For example in the packages that follow, call setup initiation is indicated by either a "sup" event or an "hd" (FXS - "BL" packages) or in the case of the "DO" package below (FXO), by the "rg" event so that those events have an "S" in the event column indicating that they are auditable.

2. The other question of interest is to know whether the telephony leg of the call is in the idle state so that a new call can be initiated. This is indicate by the "rlc" (release complete) event-state for packages that have that event.

* Signal If nothing appears in this column then this event cannot be signaled on request by the Call Agent. Otherwise, one of the following symbols is provided to identify the type of signal:

- "OO" On/Off signal. The signal is turned on until commanded by the Call Agent to turn it off, and vice versa.
- "TO" Timeout signal. The signal lasts for a given duration unless it is superseded by a new signal or terminated on detection of an event. Default time-out values are supplied. A value of zero indicates that the time-out period is infinite. The provisioning process may alter these default values.
- "BR" Brief signal. The signal has a short, known duration.

* Additional info Provides additional information about the event/signal, e.g., the default duration of TO signals.

Unless otherwise stated, all of the events/signals are detected/applied on endpoints and audio generated by them is not forwarded on any connection the endpoint may have. Audio generated by events/signals that are detected/applied on a connection will however be forwarded on the associated connection irrespective of the connection mode.

2.1. Events and Signals for the "MS" package:

The following codes are used to identify events and signals for the "MS" package:

Table 5 "MS" Package Events and Signals

Code	Description	Event	Signal	Additional Info
ans	Call Answer	P	BR	
bl	Block	S	BR	
bz	Busy tone	-	TO	Time-out = 30 seconds
inf	Information Digits	x	-	
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
rel	Release Call	P	BR	
res	Resume call	P	BR	
rlc	Release complete	P,S	BR	
ro	Reorder tone	-	TO	Time-out = 30 seconds
rt	Ringback tone	-	TO	Time-out = 180 seconds
sup	Call Setup	P,S	TO	Time-out when signal completes out-pulsing
sus	Suspend call	P	BR	

2.2. Events and Signals for the "DT" package:

The following codes are used to identify events and signals for the "DT" package:

Table 6 "DT" Package Events and Signals

Code	Description	Event	Signal	Additional Info
ans	Call Answer	P	BR	
bl	Block	S	BR	
bz	Busy tone	-	TO	Time-out = 30 seconds
dl	Dial tone	-	TO	Time-out = 16 seconds
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
rel	Release Call	P	BR	
res	Resume call	P	BR	
rlc	Release complete	P,S	BR	
ro	Reorder tone	-	TO	Time-out = 30 seconds
rt	Ringback tone	-	TO	Time-out = 180 seconds
sup	Call Setup	P,S	TO	Time-out when signals completed out-pulsing
sus	Suspend call	P	BR	

2.3. Events and Signals for the "BL" package (Basic PBX)

The following codes are used to identify events and signals for the "BL" package. This package looks very much like a simplified line package:

Table 7 "BL" Package Events and Signals

Code	Description	Event	Signal	Additional Info
bz	Busy tone	-	TO	Time-out = 30 seconds
dl	Dial tone	-	TO	Time-out = 16 seconds
hd	Off-hook	P,S	-	
hf	Flash hook	P	-	
hu	On-hook	P,S	-	
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
rel	Release	-	BR	
rg	Ringing	-	TO	Time-out = 180 seconds
ro	Reorder tone	-	TO	Time-out = 30 seconds
rt	Ringback tone	-	C,TO	Time-out = 180 seconds

2.4. Events and Signals for the "DO" package:

The following codes are used to identify events and signals for the "DO" package:

Table 8 "DO" Package Events and Signals

Code	Description	Event	Signal	Additional Info
ci	Caller id	x	-	
hd	Offhook	-	BR	
hf	Hook flash	-	BR	
hu	Onhook	-	BR	
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
rel	Release call	P	-	
rg	Ringing	P,S	-	
rlc	Release complete	P,S	-	
sup	Call Setup	-	TO	Time-out when signal completes out-pulsing

2.5. Events and Signals for the "MD" package:

The following codes are used to identify events and signals for the "MD" package.

Table 9 "MD" Package Events and Signals

Code	Description	Event	Signal	Additional Info
ans	Call Answer	P	BR	
awk	Acknowledge wink	P	BR	
bl	Call Block	S	BR	
bz	Busy tone	-	TO	Time-out = 30 seconds
cwk	Continue Wink	-	BR	
inf	Information Digits	x	TO	Time-out when signals completed out-pulsing
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
rel	Release Call	P	BR	
res	Resume call	P	BR	
rlc	Release complete	P,S	BR	
ro	Reorder tone	-	TO	Time-out = 30 seconds
rt	Ringback tone	-	TO	Time-out = 180 seconds
sup	Call Setup	P,S	TO	Time-out when signals completed out-pulsing
sus	Suspend call	P	BR	
swk	Start Wink	x	-	

2.6. Events and Signals for the "MO" package:

The following codes are used to identify events and signals for the "MO" package.

Table 10 "MO" Package Events and Signals

Code	Description	Event	Signal	Additional Info
ans	Call Answer !Note	P	-	
oc	Operation Complete	x	-	
of	Operation Fail	x	-	
orbk	Operator Ringback	x	-	
rbz	Reverse make busy	P,S	-	
rcl	Operator Recall	-	BR	
rel	Release Call	P	BR	
res	Resume Call	-	BR	
rlc	Release complete	P,S	BR	
sup	Call Setup	-	TO	
sus	Suspend Call	-	BR	
swk	Start Wink	x	-	

!Note: There is no indication that the operator answered the call. The "ans" event is an indication that off-hook was received from the far end which simply indicates that the destination address was received properly and the calling number is in the process of being outpulsed.

2.7. Event and Signal Descriptions

The following provides a list of the event and signal descriptions.

The event/signal name appears in parenthesis followed by the corresponding Event + Signal attribute code plus a list of the packages in which the event/signal occurs.

Call answer (ans; P + BR; DT,MD,MS,MO): Off-hook signal normally indicates that the call has been answered and that cut-through has been established. The exception is the "MO" package where it simply indicates that off-hook was received and the calling number is in the process of being sent (i.e., there is no event available to indicate that the operator answered the call for operator services signaling).

Acknowledgement Wink (awk; P + BR; MD): This event is only applicable to the "md" package. It provides an indication that all digits have been received correctly. In an outgoing trunk, the event is requested and when received indicates that the connecting switch

received all of the addressing information. On an originating trunk, this signal is sent to inform the other end that all addressing information has been received. If the Call Agent is providing a transit application for example, in which incoming and outgoing trunks are both EANA trunks, then after acknowledgement wink is received from the terminating trunk, it is passed to the originating side so that the originating side knows that addressing has passed to the destination switch.

Call Block (bl; S + BR; DT,MS,MD): A steady off-hook signal applied to one-way incoming trunks to indicate that no further calls will be accepted. When "bl" is used as a signal then the "rel" signal is used to release the blocking condition.

A Call Agent should only request the "bl" event in a case where it knows that this is a one-way outgoing trunk, and it should never see an incoming call-setup request ("sup" event). As such if "bl" is requested as an event, then "sup" is suppressed as a persistent event.

Busy tone (bz ; - + TO; BL,DT,MD,MS): Refer to ITU E.180. The definition of the tone is defined by the national characteristics and may be established via provisioning. Station Busy is defined in GR-506-CORE - LSSGR, SIGNALING, Section 17.2.6. as a combination of two AC tones with frequencies of 480 and 620 Hertz and levels of -24 dBm each, to give a combined level of -21 dBm. The cadence for Station Busy Tone is 0.5 seconds on followed by 0.5 seconds off, repeating.

Caller Id (ci(time, number, name); x + -; DO): See TR-NWT-001188, GR-30-CORE, and TR-NWT-000031. Each of the three fields are optional, however each of the commas will always be included.

The time parameter is coded as "MM/DD/HH/MM", where MM is a two-digit value for Month between 01 and 12, DD is a two-digit value for Day between 1 and 31, and Hour and Minute are two-digit values coded according to military local time, e.g., 00 is midnight, 01 is 1 a.m., and 13 is 1 p.m.

The number parameter is coded as an ASCII character string of decimal digits that identify the calling line number. White spaces are permitted if the string is quoted, however they will be ignored.

The name parameter is coded as a string of ASCII characters that identify the calling line name. White spaces are permitted if the string is quoted.

A "P" in the number or name field is used to indicate a private number or name, and an "O" is used to indicate an unavailable number or name. The following example illustrates the use of the caller-id event:

O: ci(10/14/17/26, "555 1212", somename)

Continue Wink (cwk ; - + BR; MD): This signal is only applicable to the "md" package. It provides an indication that digits sent have been accepted, and further digits must be sent in order to process the call. For example, when using FGD EAIN signaling, this would correspond to sending a wink after the country access code had been received to indicate readiness to receive identification and address fields.

Dial-tone (dl ; - + TO; BL,DT): Refer to ITU E.180. The definition of the tone is defined by the national characteristics and may be established via provisioning. In GR-506-CORE - LSSGR, SIGNALING, Section 17.2.1, sial Tone is defined as a combination of two continuous AC tones with frequencies of 350 and 440 Hertz and levels of -13dBm each to give a combined level of -10 dBm. It is considered an error to try and play dial-tone on a phone that is on hook and an error should consequently be returned when such attempts are made (error code 402 - phone on hook).

Information Digits (inf(<inf-digits>); x + TO; MS,MD): On an outgoing call ("md" package only) it is used as a signal to out-pulse the address information when doing overlapped sending.

On an incoming call it is used as an event to indicate that an MF digit string has been received. In this case, <inf-digits> are all of the digits accumulated up to and including the digit delimiters ST, ST’, ST’’, ST’’’. Multiple sequences of digits ending with one of the ST digits may be passed in a single "inf" event. (Note that K0 is the same as KP, K1 is sometimes referred to as KP’ etc. Similarly S0 is the same as ST, S1 is the same as ST’ and so on.

The value of <inf-digits> is a comma separated list of MF digits: MF1, MF2, ..., MFn

where each of MFi will be one of the following MF digit symbols:

Table 11 MF Digit Symbols

Symbol	MF digit
0	MF 0
1	MF 1
2	MF 2
3	MF 3
4	MF 4
5	MF 5
6	MF 6
7	MF 7
8	MF 8
9	MF 9
K0	MF K0 or KP
K1	MF K1
K2	MF K2
S0	MF S0 or ST
S1	MF S1 or ST’
S2	MF S2 or ST’’
S3	MF S3 or ST’’’

Thus, an example signal or event might look like:

```
inf(k0, 5,5,5,1,2,3,4, s0)
```

An example where the inter-digit timer expired after the 5,5,5 would appear as follows:

```
inf(k0, 5,5,5)
```

Operation Complete (oc ; x + -; all): The operation complete event is generated when the gateway was asked to apply one or several signals of type TO on the endpoint, and one or more of those signals completed without being stopped by the detection of a requested or persistent event such as setup. The completion report may carry as a parameter the name of the signal that came to the end of its live time, as in:

```
O: ms/oc(ms/sup)
```

or

```
O: bl/oc(bl/rg)
```

When the operation complete event is requested, it cannot be parameterized with any event parameters.

Note that when requested at the same a signal for "sup" (out-pulsing - a TO event), the operation complete event will indicate when out-pulsing is complete.

Operation failure (of; x + -; all): In general, the operation failure event may be generated when the endpoint was asked to apply one or several signals of type TO on the endpoint, and one or more of those signals failed prior to timing out. The completion report may carry as a parameter the name of the signal that failed, as in:

O: ms/of(ms/sup)

or

O: bl/of(bl/rg)

When the operation failure event is requested, it cannot be parameterized with any event parameters.

Operator ringback (orbk; x + -; MO): The description of the signaling MF CAS signaling that results in this event is describe in the appendix of TR-NPL-000258 [3]. In brief, it is normally a wink-on signal which may or may not be followed by an MF tone. This event will be generated when the operator service requests that the calling party be alerted ("mo" package only).

Reverse make busy (rbz; P + -; MO): This event corresponds to a "blocking" (off-hook) generated by the other end of the one-way operator services trunk ("mo" package). It has the same semantics as of the "bl" event in other packages.

Operator recall (rcl; - + BR; MO): This signal may be applied to invoke operator recall, e.g., due to customer hook-flash to bring the operator back.

Release call (rel; P,S + BR; BL,DT,MD,MO,MS,DO): A "rel" signal sent by the Call Agent to the Media Gateway is a request to release all of the resources associated with the telephony leg of the call. This may also result in an off-hook signal being sent when appropriate. As a result of an "rel" signal, the gateway will respond with an "rcl" event, whenever the resources have been released. Releasing resources associated with the telephony leg of the call does not affect existing connections (network legs). It’s up to the Call Agent to send the appropriate delete connection commands in order to release any network connections to that endpoint.

In the case of the FXS ("BL") package, the "rel" signal is used to provide a tip-ground release for ground-start trunks. In the case of loop-start trunks, requesting to play this signal has no effect.

The Media Gateway generates a "release call" event whenever a call is released as a result of an on-hook event from an originating end of a call (normal release) or due to abnormal event that resulted in releasing the call. The event may be parameterized with one of the following cause codes indicating the reason for the release:

Table 12 Release Reason Codes

Cause Code	Reason
0	Normal release
44	Requested channel/circuit not available (glare or incoming seizure detected during call setup)
111	Protocol/signaling error, unspecified (e.g. timeout)

Note that a "rel" event with reason code "0" indicating normal release (due to an incoming on-hook) will only be "notified" by a gateway where a call origination occurred. This behavior follows the rule that when an originator releases the call, all resources may be released. The corresponding event for on-hook on the terminating end of a call is the "sus" event which only indicates hook-status and does not result in any resources being released. It is always up to the Call Agent to release the call (by sending the "rel" signal) for the terminating end of a call.

For FXO ground-start case ("DO" package), the Media Gateway generates a "release call" event whenever a call is released as a result of a tip-ground release event from the far end.

Resume call (res ; P + BR; DT,MD,MS,MO): This indicates that the called party resumed the call, i.e., the party went off-hook after a previous suspend ("sus") but before the originating switch released ("rel") the trunk. The "sus" and "res" events/signals are used to propagate on-hook and off-hook events without releasing the resources associated with the call. In all but the operator services case ("MO" package), these events would normally be propagated from the terminating to the originating end (i.e., requested as events from the terminating end of the call and sent to the gateway as signals to a gateway on the originating side of the call).

However, it is up to the Call Agent to decide whether it wants to do "suspend"/"resume" processing. If it doesn't, when it receives a "sup" event from the terminating end of the call it can simply go ahead and tear down the call immediately (send "rel" and delete connections to the endpoints on gateways at both originating and terminating end of the call).

In the case of operator services and 911, "sus" and "res" are used to pass off-hook and on-hook signals to the operator without releasing any of the resources associated with the call.

Ringing (rg; P,S + TO; BL,DO): This signal is used for outgoing basic trunks ("bl" package). See GR-506-CORE - LSSGR: SIGNALING, Section 14. The provisioning process may define the ringing cadence. It is considered an error to try and ring if the trunk indicates off hook and an error should consequently be returned when such attempts are made (error code 401 - phone off hook).

In the case of the "DO" package, "rg" is defined as an event used to indicate detection of ringing.

Release complete (rlc;P,S + BR; DO,DT,MD,MO,MS): The endpoint and Call Agent use the release complete event/signal to confirm the call has been released and the trunk is available for another call. For FXO ground-start ("DO" package), this represents the release of the tip-ground event from the PBX after the gateway goes on-hook.

Reorder tone (ro; - + TO; BL,DT,MD,MS): Reorder tone is a combination of two AC tones with frequencies of 480 and 620 Hertz and levels of -24 dBm each, to give a combined level of -21 dBm. The cadence for reorder tone is 0.25 seconds on followed by 0.25 seconds off, repeating continuously. See GR-506-CORE - LSSGR: SIGNALING, Section 17.2.7.

Ring back tone (rt; - + TO; BL,DT,MD,MS): Audible Ring Tone is a combination of two AC tones with frequencies of 440 and 480 Hertz and levels of -19 dBm each, to give a combined level of -16 dBm. In the US the cadence for Audible Ring Tone is defined to be 2 seconds on followed by 4 seconds off. The definition of the tone is defined by the national characteristics of the Ring-back Tone, and MAY be established via provisioning. See GR-506-CORE - LSSGR: SIGNALING, Section 17.2.5.

Call Setup (sup ; P,S + TO; DO,DT,MD,MS,MO): The event/signal is used both for outgoing and incoming call setups. Each will be described separately in the following.

Outgoing call setup:

On an outgoing trunk, the "sup" signal is used to seize a trunk and out-pulse digits. The "sup" signal is parameterized with up to four parameters sup(<ct>, <ca>, <id>, <addr>), depending on the package. The order of these parameters does not matter. The following table indicates which ones are mandatory ("M"), optional ("O") or forbidden ("F") for the various packages.

Table 13 "sup" parameters.

Parameter	MS	DT	MO	MD	DO
<ct>	F	F	F	M	F
<ca>	F	F	F	O	F
<id>	F	F	M	M	F
<addr>	M	M	M	O	M

The <ct> parameter is of the format ct(<ct-value>) where <ct-value> indicates the CAS signaling type and can have one of two values "nda" (North American Direct Access) for EANA and "nta" (North American Tandem Access) for EAIN. The reason this parameter is needed in the case of trunks that handle the "MD" packages is because the same trunk can be used for both. The <addr> field contains the destination number and when present will be on the form

addr(dig1, dig2, ..., dign)

The <id> field contains the identification of the caller and when present will be of the form:

id(dig1, dig2, ..., dign)

The <ca> field contains the country address information and when present will be of the form:

ca(dig1, dig2, ..., dign)

When present, the <addr> field contains the destination number and will be of the form

addr(dig1, dig2, ..., dign)

where digi is an MF symbol as defined in table 11 in the case of "MS", "MO", and "MD" packages and digi is a DTMF symbol (0-9, *,#,A,B,C,D) in the case of the "DT" and "DO" packages.

The following table shows some interactions between the Media Gateway (MG) and the Switched Circuit Network (SCN) for single stage outpulsing applications ("DT", "MS" and "DO" packages):

Table 14 SCN Sequencing during Call Setup (single stage outpulsing)

Interface Type	Setup	Interactions
wink start	sup(add(<addrvalue>))	MG: off-hook -> SCN
		MG: <- wink SCN
		MG: <addrvalue> -> SCN
Immediate Start or FXO	(sup(addr(<addrvalue>))	MG: off-hook -> SCN
		MG: <addrvalue> -> SCN

Call setup signal example for this case (MF signaling):

```
sup(addr(s0,5,5,5,1,2,3,4,k0))
```

The "MO" and "MD" packages involve multi-stage signaling and multiple parameters. In the case of the "MD" package the following table shows some of the interactions:

Table 15 SCN Sequencing during Call Setup (EANA and EAIN)

Setup	Interactions
sup(ct(nda),addr(<addrvalue>), id(<idvalue>))	MG: off-hook -> SCN
	MG: <- wink SCN
	MG: <idvalue> -> SCN
	MG: <addrvalue> -> SCN
sup(ct(nta), ca(<cavalue>), addr(<addrvalue>), id(<idvalue>))	MG: off-hook -> SCN
	MG: <- wink SCN
	MG: <cavalue> -> SCN
	MG: <- wink SCN
	MG: <idvalue> -> SCN
	MG: <addrvalue> -> SCN
sup(ct(nta), ca(<cavalue>), id(<idvalue>))	MG: off-hook -> SCN
	MG: <- wink SCN
	MG: <cavalue> -> SCN
	MG: <- wink SCN
	MG: <idvalue> -> SCN

The last example is an overlapped sending example where the address value would be sent later using the "inf" signal.

An example setup:

```
sup(ct(nta),ca(k0,1,3,8,9,9,0,0,1,0,s0),id(k0,0,5,5,5,1,2,3,4,s0))
```

In all of the above cases, the "ans" event is an indication of off-hook from the far end (the other end answered). However, in the case of the operator service signaling (OSS) protocol of Feature Group D - shown in the following table, off-hook from the operator is part of the protocol (a request for the calling number) so that "ans" in this case does not indicate that the operator answered (only that off-hook/request for calling number was received).

Table 16 SCN Sequencing during Call Setup OSS Protocol ("MO" Package)

Setup	Interactions		
sup(ct(nda),addr(<addrvalue>), id(<idvalue>))	MG	off-hook ->	SCN
	MG	<- wink	SCN
	MG	<- off-hook	SCN
	MG	<addrvalue> ->	SCN
	MG	<idvalue> ->	SCN

Incoming Call Setup: A "sup" event is used to indicate when an incoming call arrives (corresponding to the incoming off-hook event). The event is provided without parameters.

Suspend call (sus; P + BR; DT,MD,MS,MO): Suspend ("sus") is an on-hook event that is an indication that the called party went on-hook.

An on-hook event will be "notified" to a Call Agent as a "sus" event for interfaces that use the "MS", "DT" and "MD" packages from an endpoint at a terminating end of a call (as compared to a "rel" event from the originating side). The "sus" event from the terminating endpoint gives the Call Agent the option of doing "suspend/resume" processing or to simply release the call.

The "sus" signal may be used to send an on-hook to the originating party without releasing the resources associated with the telephony leg of the call. The "rel" signal on the other hand would send an on-hook and release the resources associated with the call.

Because of this "sus" can be followed by "res" (off-hook) and allow the call to resume, while "rel" cannot be followed by "res" because the call no longer exists.

For E911 ("MO" package), the operator is normally in control of releasing the call so that, "sus" (on-hook), "res" (off-hook) and "rcl" (flash-hook) can be used to pass user hook events to the operator without releasing the call.

Start Wink (swk; x + - MD,MO):. An Call Agent can optionally request the MG to notify it when the wink start signal occurs. Note that wink start ("swk") cannot be applied by the Call Agent as a signal. The occurrence of wink-start on an incoming trunk is a reflexive action that does not require Call Agent involvement.

3.0. Hook-State Signals and Events

3.1. Overview of Approach

As mentioned in the introduction, a higher level view is taken for on-hook and off-hook events for many of the CAS packages to make the interface Q.931-like. This provides the advantage that:

* Similar call flows result as when dealing with Q.931-based interfaces (e.g., PRI)
* It’s more evident (for ease in debug) when looking at message as to exactly what is going on without having to refer to previous flows.

This does require that media gateways maintain some state but this is a relatively small price to pay.

One example of this is the "sup" signal which involves sending off-hook followed by digits as a high level signal. The "ans" event is also used to represent off-hook but from the terminating end at the point where the call is answered.

3.2. Suspend/Resume Processing

Other signals and events "sus" for suspend, "res" for resume and "rel" for release are based on the concept that one end (the originator) is in control of the call. If the controlling end goes on-hook a "rel" is notified to the Call Agent, and results in a the call being released. However, if the non-controlling (terminating) end goes on-hook, a "sus" event occurs (instead of the "rel" event). This gives the Call Agent the option of doing suspend/resume processing.

If the Call Agent decides not to do suspend/resume processing, it can simply send "rel" and delete connection commands to the endpoints after it receives "sus" from the non-controlling (terminating) end of the call.

On the other hand, if it decides to do suspend/resume processing, it can start a timeout when it receives the "sus" event from the non-controlling (terminating) end of the call and continue the call if it receives a "res" (off-hook) event. It also has the option of propagating the "sus" and "res" as signals back to the ingress gateway and allow it an opportunity to release the call ("rel" event) or not. In any case the use of "sus" and "res" signals give another level of control over the "rel" signal which will not only send on-hook but also release the resources associated with the telephony leg of the call.

3.3. Control over Disconnect for E911

Note that for E911 (the "MO" package) is a special case in that the operator (terminating end) is always the controlling end. The "sus" and "res" signals are used to pass user hook state forward to the operator. The "rel" event is passed back as a notify to the Call Agent when on-hook is received from the operator indicating that the Call should be released. If the "rel" is not received the call should continue to stay up.

3.3. Release and Release Complete

The "rel" signal/event generally means on-hook but more that that it also indicates "release of resources" for the telephony leg of the call. If a Call Agent sends a "rel" signal instead of "sus" it is requesting the call to be abandoned (i.e., "rel" cannot be followed by "res").

The "rel" signal does not also imply that connections should be deleted so that to completely release the call including connections would require a DLCX in addition to (or conjunction with) the signal "rel".

In addition to being a signal, "rel" can also be an event triggered by either:

* An on-hook from the controlling end of the call, or

* Some abnormal event within the media gateway such that the telephony leg of the call can no longer be maintained.

In any case, "rel" (release) is generally followed by an "rlc" (release complete). The release complete signal/event indicates that the trunk resources are now completely released and available for another call. This is also an event state that can be audited as indicated by the "S" in the column for that event (allowing the Call Agent to check to see if that trunk is released and available).

Examples of the use of "rel" and "rlc":

* Call Agent sends a "rel" to release a trunk, resulting in an outgoing off-hook being sent for that trunk. When the media gateway receives the on-hook from the other end, it returns an "rlc" event indicating that the trunk is released and available.
* The media gateway receives a on-hook from the trunk at the controlling end of the call, resulting in a "rel" event being sent to the Call Agent. The Call Agent then sends an "rlc" to the media gateway, resulting in on-hook being sent in the opposite direction.
* An "rel" event is sent to the Call Agent in the event of some abnormal condition in which the media gateway is unable to sustain the telephony leg of the call (e.g., glare condition). The Call Agent sends an "rlc" to the gateway to complete the release of the call. (note that "rlc may not correspond to on-hook but is generally sent anyway in response to a "rel".)
* The Call Agent can send a "rel" (instead of "sus") signal to the controlling (originating) end of the call to abandon the call. The gateway will return with "rlc" when an off-hook has been received from the other end and all the resources have been released.
* A "rel" can be sent on one-way incoming trunk to release a block ("bl") sent earlier.

The "BL" (FXS) package is a simple line package, so does not have these events (uses "hd", "hf", and "hu" as the only hook state events).

The "DO" (FXO) package, however, does have "rel" and "rlc" because in the ground-start case there is the ability to "release" the call as result of a tip-ground release. The signal "rel" is used if the PBX releases the call first (followed by S: hu from the call Agent to complete the release). Alternatively, the Call Agent can send the S: hu to initiate the release - followed by an "rlc" event from the media gateway to Call Agent when the PBX does the tip ground release. Although the loop-start trunks would not normally have this behavior (only applies to ground-start), the media gateway should emulate the behavior in the case of loop-start in order to allow the Call Agent a common interface.

3.4. Blocking CAS Trunks

In addition to the above signals and events, there is the "bl" signal/event which is used for blocking one-way trunks (does not work for two way trunks) by providing a continuous off-hook.

3.5. Summary of Hook-State Events

The following summarizes the use of the various events that involve off-hook and on-hook from call establishment to tear-down. This applies mainly to "MS", "DT", "MD" and to a lesser extent the "DO" package.

* The "sup" event represents off-hook origination.
* The "sup" signal with parameters provides off-hook with digit outpulsing on the terminating side.
* Once outpulsing is completed, an "ans" event indicates off-hook from the termination side (the called party has answered).
* The call agent can then send an "ans" signal (off-hook) to the originating end to indicate to the caller that the called party has answered.
* The Call Agent can send a "rel" to either end at any time to tear down the call (e.g., to abort the call).
* The media gateway can send "rel" to indicate abnormal termination of the call (with a reason as a parameter).
* However, under normal operation once a call is established, the Call Agent can expect a "sus" (suspend) event from the termination end to indicate that the caller went on-hook and a "res" if the called party goes off-hook again before the Call Agent tears down the call. The Call Agent can send these same signals to the originating end to indicate off-hook and on-hook to the calling party without tearing down the call.
* During normal operation, once the call is established, on-hook from the calling party (origination side) would result in a "rel" signal. The Call Agent would then normally send the "rel" signal to the terminating end to terminate the call. "rel is normally followed by "rlc" (e.g., media gateway indicates calling party on-hook with "rel" and the Call Agent responds with "rlc", which sends on on-hook back to the calling party to indicated release complete.

The "MO" package is a bit different in that normally only the terminating side (the operator) can release the call ("rel" event). The "sus" and "res" are forward signals to the operator indicating user hook-status.

4.0. Glare Handling

4.1. Glare on MF Bi-directional Wink-start Trunks

Gateways may have a configurable glare bit on a per-DS0 basis that can be set to indicate whether the gateway is the controlling or non-controlling "switch". However, in general, PBXs are either pre-configured or can be configured to behave as non-controlling switches. In this case if they see an off-hook that exceeds allowable wink length, they will attach a receiver, go on-hook, and await digits for a new call. Meanwhile the PBX will retry its original call on another trunk.

If the gateway behaves like a controlling switch, when glare is detected, the gateway will wait for up to some timeout value (default value of 4 seconds) until the incoming off-hook changes to an on-hook state at which time it will start out-pulsing in the normal manner. If the timeout occurs before the state change to on-hook occurs, the gateway will send a release event to the Call Agent (a "rel(44)" event - cause code indicating glare).

When "rel(44)" is sent by the gateway, that is an indication to the Call Agent that the call is in the process of being released and that the Call Agent should give up on that trunk. However, the gateway may not actually want to send the on-hook at that time in order to avoid the possibility that the other end takes the on-hook as a wink. Instead, it may start a second timer and wait some longer period of time (e.g., 16 seconds or so) before releasing the trunk. If it receives an on-hook prior the timeout, it completes the release by going on-hook. If, on the other hand, the timer expires before the other end goes on-hook, it will simply go on-hook and wait for the other end to go on-hook. In any case, once both ends have returned to the on-hook state, an "rlc" event is sent to the Call Agent.

4.2. Glare Handling - Basic PBX Trunks

In order to reduce the chances of glare, the gateway should try a ringing pre-trip test prior to sending ringing on a basic ground start trunk. If glare is detected on an outgoing seizure of a basic PBX trunk, the request for ringing should be "Nacked" (error code 401 - phone off-hook) to the Call Agent.

5.0. Example Call Flows

5.1. PBX to PBX ("MS", "DT, and "BL" packages).

The following call flows involve a single Call Agent that handles both sides of the call (i.e., the inter-Call-Agent signaling is ignored). The components involved in the call are:

* The Call Agent (CA)

* The originating Media Gateway (GW-o) and

* The terminating Media Gateway (GW-t)

5.1.1. Call Setup Flows

The following describes some PBX to PBX call. The table gives an overview of the initial part of the call flow with details to follow.

Steps	GW-o	CA	GW-t
A1	NTFY[seizure] ->		
A2		<- Ack	
A3		<- RQNT[request digits]	
A4		Ack ->	
A5	NTFY[digits] ->		
A6		<- Ack	
B1		<- CRCX [M:recvonly, LCO]	
B2	Ack[SDP1] ->		
B3		CRCX [M:sendrecv, LCO, SDP1] ->	
B4		<- Ack [SDP2]	
B5		<- MDCX [recvonly,SDP2]	
B6		Ack ->	

Step A1 PBX seizure results in a notify to the Call Agent indicating the start of a call setup:

```
NTFY 3001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
O: ms/sup (or dt/sup)
```

In the case of the "BL" package (basic PBX) the interface looks like a simplified line interface with the standard "hd" event for off-hook:

```
NTFY 3001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
O: bl/hd
```

Another alternative would have been to use an embedded request in the RQNT that resulted in this notify and combine that request with step A3. Example - "ms" package:

```
RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
R: ms/sup(E(R(ms/inf, ms/rel))
```

Step 3 could also be eliminated by the use of "loop" mode e.g.:

```
RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
Q:loop
R: ms/sup, ms/inf, ms/rel
```

This would result in both notifies occurring without requiring the RQNT in step A3.

Step A2 The Call Agent sends an Ack:

```
200 3001 OK
```

Step A3 The Call Agent requests that digits be collected. The approach used here depends on the type of PBX interface. For an MF interface the Call Agent requests that information digits be collected as follows:

```
RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
R: ms/inf, ms/rel
```

The Call Agent also asks to be told if the trunk gets released for some reason ("rel" event) in the process of call setup (release event may be due to some signaling error for example). For DTMF trunks (wink-start, immediate start and Basic PBX), the request is based on a digit map so looks a bit different:

```
RQNT 2001 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
R: d/[0-9*#T](D), dt/rel (bl/hd in the case of Basic PBX)
```

```
D: (xxxxxxx | x.[T#])
S: dt/dl

Note: the request to signal dial-tone may or may not be here
depending on PBX interface requirement - bl/dl  required for
Basic PBX;  dt/dl for some Immediate Start interfaces.
```

Step A4 The gateway responds with an ack:

```
200 2001 OK
```

Step A5 Once the digits are collected the gateway notifies the call agent. In the case of an MF interface, the resulting notify will look like the following

```
NTFY 3002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
O: ms/inf(k0,5,5,5,1,2,3,4,s0)
```

In the case of a DTMF interface (including Basic PBX), it will look like the following:

```
NTFY 3002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
O: d/5,d/5,d/5,d/1,d/2,d/3,d/4
```

Step A6 The Call Agent responds with an ack:

```
200 3002 OK
```

Step B1 The Call Agent now requests that a receive-only connection be made.

```
CRCX 2002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
C: A7453949499
L: a:PCMU,s:off,e:on
M: recvonly
X: 0123456789B1
R: ms/rel (or dt/rel or bl/hu).
```

Step B2 The Gateway acks with a connection ID and provides the SDP information:

```
200 2002 OK
I: 23474FE
```

```
v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B3 The Call Agent passes this SDP information to the terminating gateway (GW-t) as part of the connection request:

```
CRCX 4001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
C: A7453949499
X: 45375840
L: a:PCMU,s:off,e:on
M: sendrecv

v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

Note that the call setup on the terminating trunk can be done with this CRCX, although in this call flow - it is shown later (step C1).

Step B4 The terminating gateway, responds with an ack and its SDP information:

```
200 4001 OK
I: 34738A

v=0
o=- A7453949499 0 IN IP4 47.123.34.33
s=-
c=IN IP4 47.123.34.33
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B5 Call Agent sends a modify connection request with connection mode receive-only to the origination gateway and includes the SDP information with the selected profile from the termination gateway.

```
MDCX 2003 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
C: A7453949499
I: 34738A
M: recvonly
```

```
v=0
o=- A7453949499 0 IN IP4 47.123.34.33
s=-
c=IN IP4 47.123.34.33
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B6 The Gateway Acks the modify connection request

```
200 2003 OK
```

The following table shows the remainder of the call flow to set up the call except for Basic PBX (Basic PBX shown in) with details to follow.

Steps	GW-o	CA	GW-t
C1		RQNT [S: ms/sup, R: ms/oc, ms/rel, ms/ans] ->	
C2		<- Ack	
C3		<- NTFY [O:ms/oc(ms/sup)]	
C4		Ack ->	
C5		<- NTFY [O: ms/ans]	
C6		Ack ->	
C7	<- MDCX [M:sendrecv, S: ms/ans, R: ms/rel]		
C8	Ack ->		
C9		RQNT[R: ms/sus] ->	
C10		<- Ack	

Step C1 The Call Agent does a setup request to the terminating gateway The setup request for an MF PBX interface (wink start or immediate start) will be the following:

```
RQNT 4002 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
Q: loop
S: ms/sup(addr(ko,5,5,5,1,2,3,4,s0))
R: ms/oc, ms/rel, ms/ans
```

Note that the result of the "sup" signal is the following sequence on the interface to the PBX:

* off-hook -> PBX
* wink -> PBX (for wink-start trunks; for immediate start this part of the sequence does is not included)
* Digits sent to PBX

For DTMF PBX interface (except Basic PBX), the only difference is that the MF start and end delimiters (k0 and s0) are not included:

```
RQNT 4002 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
Q: loop
S: dt/sup(addr(5,5,5,1,2,3,4))
R: dt/oc, dt/rel, dt/ans
```

Basic PBX requires ringing and ring-back instead i.e.:

```
RQNT 4002 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
Q: loop
S: bl/rg,bl/rt@34738A
R: bl/oc,bl/hd
```

In this case ringback will come from the gateway and will start immediately with the signal request for rt@connectionID. It will end as soon as an event occurs (off-hook representing answer event) In the case of other PBX's, the ringback tone comes from the PBX so does not have to be generated by the gateway.

Note that these requests could be done as easily at the same time as the connection request (B3) saving some post-dial delay time.

Step C2 The gateway responds with an ack:

```
200 4002 OK
```

Step C3 Except for the basic PBX, case (where digits are not outpulsed) when the digits have completed being sent out, the gateway will notify the fact by indicate that the operation is complete.

```
NTFY 1001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
O: ms/oc(ms/sup) (or dt/oc(dt/sup))
```

Step C4 The Call Agent acks the notify

```
200 1001 OK
```

In the case of the BL package, steps C3 and C4 will not exist.

Step C5 When an answer is obtained from the other end (off-hook from the PBX), the gateway sends a notify to indicate:

```
NTFY 1002 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
O: ms/ans (or dt/ans or bl/hd)
```

Step C6 The Call Agent acks

```
200 1002 OK
```

Step C7 The Call Agent now sends a request to make the connection full duplex and indicates that the other end has answered the phone.

```
MDCX 2004 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
C: A7453949499
X: 45375842
I: 34738A
M: sendrecv
S: ms/ans ( or dt/ans but S: not included in the case where the
originating gateway uses the BL package)
```

Step C8 The Gateway acks the request

```
200 2004 OK
```

Step C9 The Call Agent sends a notification request to be told when the trunk to be released.

```
RQNT 4003 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375842
R: ms/rel,ms/sus (or R: dt/rel,dt/sus or R: bl/hu)
```

Step C10 The gateway acks the request

```
200 4003 OK
```

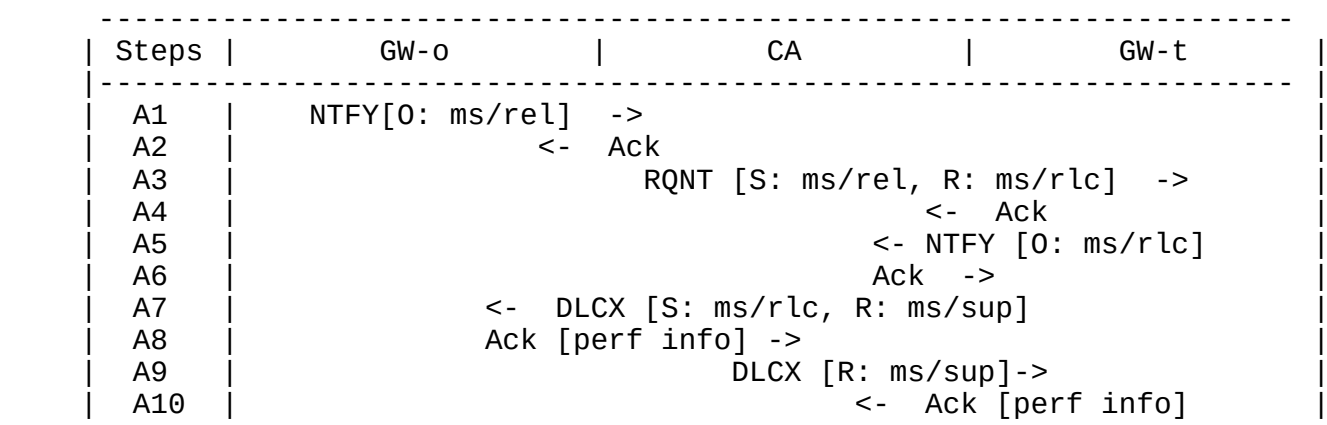
The call is now setup.

5.1.2. Call Tear-Down

Two cases are included here, one where the origination end initiates the release (section 5.1.2.1) and one where the termination end initiates the release (section 5.1.2.2).

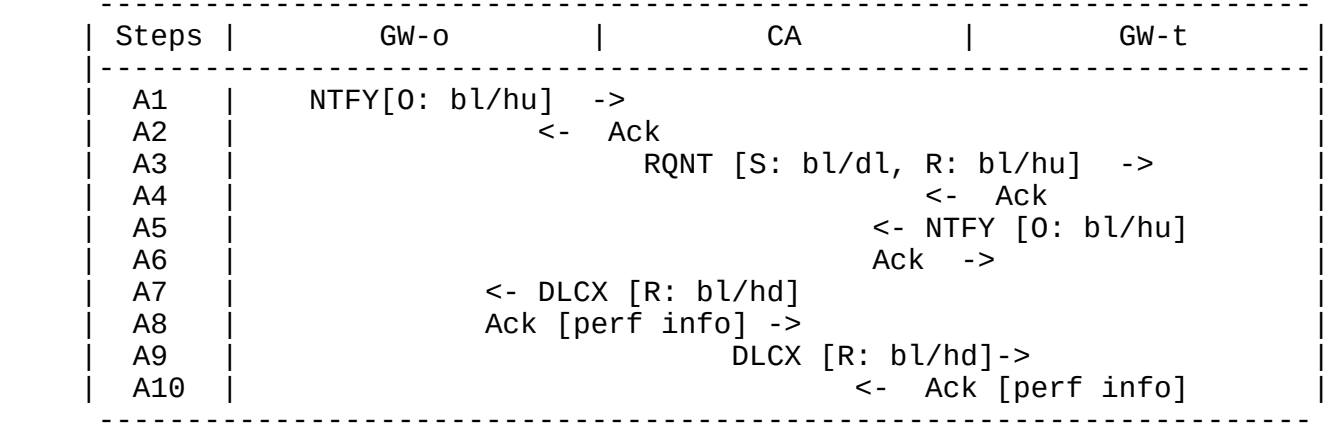
5.1.2.1. Origination End Initiates the Release

The following call flow shows an example where the origination end initiates the release for the "MS" package (similar for "DT" Package).



Steps	GW-o	CA	GW-t
A1	NTFY[O: ms/rel] ->		
A2	<- Ack		
A3		RQNT [S: ms/rel, R: ms/rlc] ->	
A4		<- Ack	
A5		<- NTFY [O: ms/rlc]	
A6		Ack ->	
A7	<- DLCX [S: ms/rlc, R: ms/sup]		
A8	Ack [perf info] ->		
A9		DLCX [R: ms/sup]->	
A10		<- Ack [perf info]	

The same call flow for the "BL" package is shown below



Steps	GW-o	CA	GW-t
A1	NTFY[O: bl/hu] ->		
A2	<- Ack		
A3		RQNT [S: bl/dl, R: bl/hu] ->	
A4		<- Ack	
A5		<- NTFY [O: bl/hu]	
A6		Ack ->	
A7	<- DLCX [R: bl/hd]		
A8	Ack [perf info] ->		
A9		DLCX [R: bl/hd]->	
A10		<- Ack [perf info]	

Step A1 The originating user goes on-hook resulting in a Notify from the gateway to indicate that the trunk is being released (reason indicating normal release)

```
NTFY 3005 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
X: 45375842
O: ms/rel(0) (or dt/rel(0) or bl/hu)
```

Step A2 The Call Agent Acks the Notify

```
200 3005 OK
```

Step A3 The Call Agent sends a request to release the trunk. (For all but Basic PBX.)

```
RQNT 4004 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375843
S: ms/rel (or dt/rel)
R: ms/rlc (or dt/rlc)
```

For the Basic PBX ("BL" package), dial-tone is played

```
RQNT 4004 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375843
S: bl/dl
R: bl/hu
```

Step A4 The Gateways acks the request

```
200 4004 OK
```

Step A5 The other end releases the call by going on-hook and a Notify is sent to the Call Agent to indicate that release is complete.

```
NTFY 1004 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375843
O: ms/rlc (or dt/rlc)
```

In the case of Basic PBX, this is:

```
NTFY 1004 ds/ds1-5/3@gw-o.whatever.net MGCP 1.0
X: 45375843
O: bl/hu
```

Step A6 The Call Agent returns an Ack

```
200 1004 OK
```

Step A7 The Call Agent sends a delete connection to the originating gateway with a request to do a release complete (which results in sending on-hook to the PBX).

```
DLCX 4005 ds/ds1-5/3@gw-o.whatever.net MGCP 1.0
X: 45375844
I: 34738A
S: ms/rlc (or dt/rlc)
R: ms/sup (or dt/sup)
```

Or in the case of Basic PBX ("BL" package):

```
DLCX 4005 ds/ds1-5/3@gw-o.whatever.net MGCP 1.0
X: 45375844
I: 34738A
R: bl/hd
```

Step A8 The Gateway Acks and provides performance information.

```
250 4005 OK
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

Step A9 The Call Agent sends a DLCX to the terminating gateway.

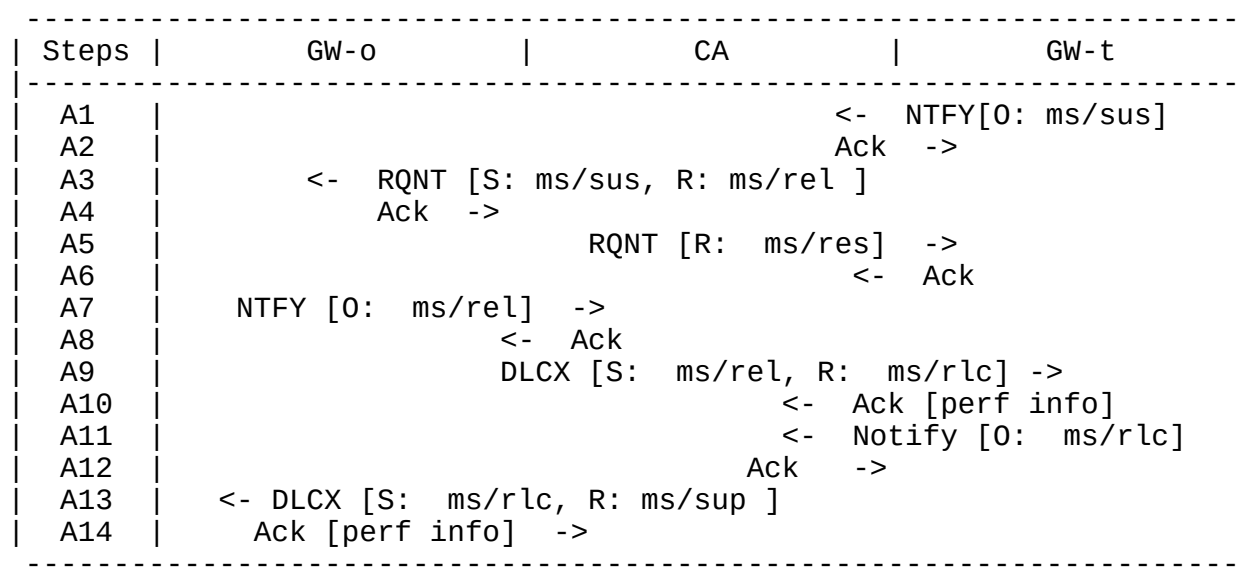
```
DLCX 2004 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 0123456789B3
I: 23474FE
R: ms/sup (or dt/sup or bl/hd)
```

Step A10 The gateway acks with performance information

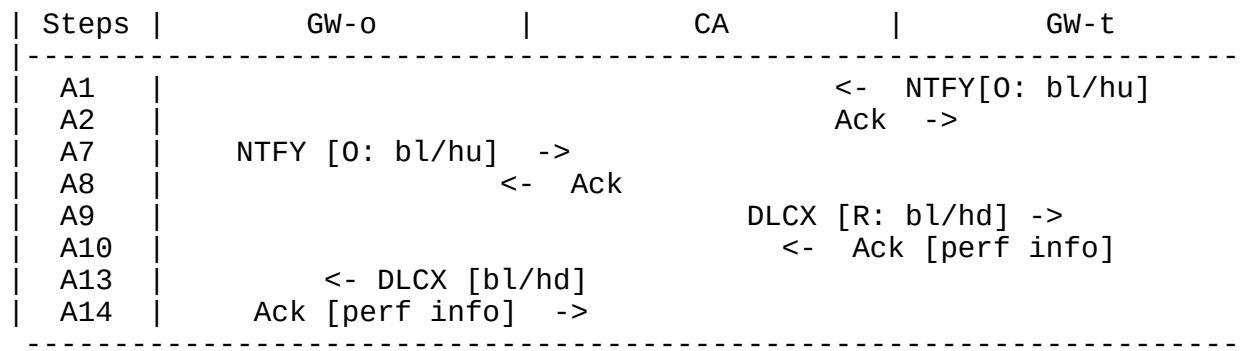
```
250 2004 OK
P: PS=1245, OS=62345, PR=0, OR=0, PL=0, JI=0, LA=48
```

5.1.2.2. Termination End Initiates the Release

The following call flow gives an example of the terminating end releasing a call for all but Basic PBX ("MS" package - "DT" package is similar).



The following shows the same call flow but for Basic PBX. There is no equivalent to steps A3-A6 and A11-A12 - so these are not included.



Step A1 An on-hook is received from the PBX. In the case of all but the "BL" package, this results in a notify with event "sus" for suspend.

Step A2 The Call Agent returns an acknowledge

The Call Agent starts a timer at this point (typically 10 seconds). If an off-hook is received from the PBX connected to GW-t before the origination side releases, the call is continued (this would appear as a "res" event or "hd" in the case of Basic PBX interface). If the origination side goes on-hook or the timer expires, then the call is torn down.

Note that for Basic PBX (the "BL" package), steps A3 - A6 are missing (these steps do not exist for basic PBX).

Step A3 A "sus" signal is sent to the originating side resulting in a on-hook being sent to the originating PBX.

Step A4 GW-o acks the request.

Step A5 The Call Agent sends a request to see off-hook or resume ("res") events.

Note: this depends on whether the Call Agent wants to do suspend/resume processing. If not, the Call Agent may simply send "rel" along with DLCX to both ends.

Step A6 GW-t acks the request.

Step A7 An on-hook is received from the originating PBX resulting in a notify from GW-o with event "rel" ("hu" for Basic PBX interface).

Step A8 The Call Agent "acks"

Step A9 A delete connection is sent to the terminating gateway with signal "rel" which results in on-hook being sent to the terminating PBX (except for basic PBX - where there is no such signal)

Step A10 GW-t acks the DLCX and provides performance information

Steps A11 and A12 do not exist for the basic PBX case.

Step A11 GW-t returns an "rlc" event

Step A12 The Call Agent "acks" the notify

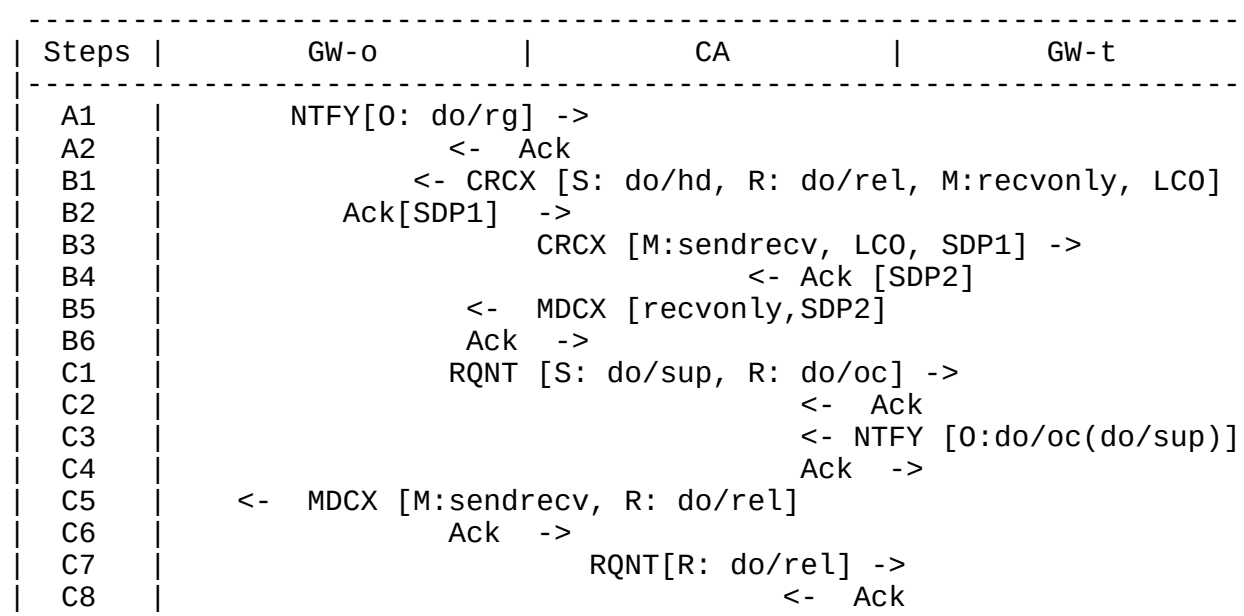
Step A13 A delete connection is sent to the originating side (with signal "rlc" except in the case of the "BL" package).

Step A14 GW-o returns an "ack" with performance information.

5.2. Example Call Flows - "DO" package

5.2.1. Call Setup Flows

The following describes some PBX to PBX call. The table gives an overview of the initial part of the call flow with details to follow.



Step A1 PBX rings results in a notify to the Call Agent indicating the start of a call setup:

```
NTFY 3001 aaln/0@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
O: do/rg
```

Step A2 The Call Agent sends an Ack:

Step B1 The Call Agent now requests that a receive-only connection be made.

> If the endpoint is running FXO ground-start, the call would also request detection of disconnect supervision from the PBX (R: do/rel) and should send an off-hook (S: do/hd) in response to ringing.

Step B2 The Gateway acks with a connection ID and provides the SDP information.

Step B3 The Call Agent passes this SDP information to the terminating gateway (GW-t) as part of the connection request.

Step B4 The terminating gateway, responds with an ack and its SDP information.

Step B5 Call Agent sends a modify connection request with connection mode receive-only to the origination gateway and includes the SDP information with the selected profile from the termination gateway.

Step B6 The Gateway Acks the modify connection request

Step C1 The Call Agent does a setup request to the terminating gateway The setup request will be the following:

```
RQNT 4002 aaln/0@gw-t.whatever.net MGCP 1.0
X: 45375841
S: do/sup(addr(5,5,5,1,2,3,4))
R: do/oc
```

Note that the result of the "sup" signal is the following sequence on the interface to the PBX:

* off-hook -> PBX
* tip-ground <- PBX (for loop-start this step does not apply)
* digits sent to PBX

Step C2 The gateway responds with an ack:

```
200 4002 OK
```

Step C3 When the digits have been completely sent out, the gateway will notify the fact by indicate that the operation is complete.

```
NTFY 1001 aaln/0@gw-t.whatever.net MGCP 1.0
X: 45375841
O: do/oc(do/sup)
```

Step C4 The Call Agent acks the notify

```
200 1001 OK
```

Step C5 The Call Agent now sends a request to make the connection full duplex and indicates that the other end has answered the phone.

If the endpoint is running FXO ground-start, the call would also requests detection of disconnect supervision from the PBX (R:do/rel)

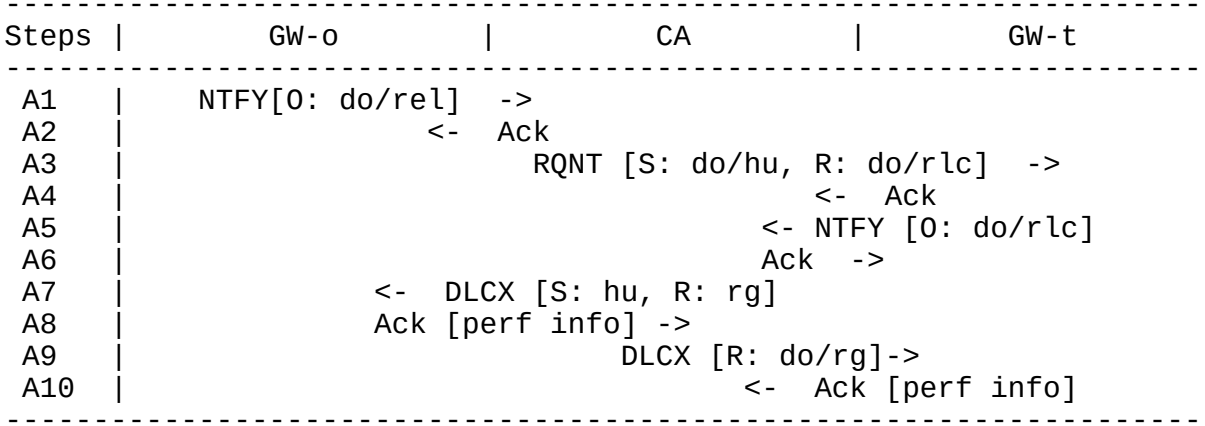
Step C6 The Gateway acks the request

Step C7 If the endpoint is running FXO ground-start, the Call Agent sends a notification request to be told when the trunk to be released (R: do/rel). This step and step C8 are not needed if the endpoint is running FXO loop-start.

Step C8 The gateway acks the request and the call is now setup.

5.2.2. Call Tear-Down

If the endpoint is running FXO loop-start, the PBX cannot initiate call release. In this case, call release is always initiated by the Media Gateway by going onhook. Disconnect supervision from the PBX is provided only for FXO ground-start. However, it does not matter whether the origination end or the termination end initiates the release. The call flows for either case are the same. Therefore, only the case where the origination end initiates the release is illustrated in this section.



Step A1 The originating PBX goes on-hook resulting in a Notify from the gateway to indicate that the trunk is being released (reason indicating normal release).

```
NTFY 3005 aaln/0@gw-o.whatever.net MGCP 1.0
X: 45375842
O: do/rel(0)
```

Step A2 The Call Agent Acks the Notify

```
200 3005 OK
```

Step A3 The Call Agent sends a request to release the trunk.

Step A4 The Gateways acks the request

Step A5 PBX at the terminating end releases the call by releasing tip-ground and a Notify is then sent to the Call Agent to indicate that release is complete.

Note that there is no ground signal in case of loop-start. However, this NTFY message is still generated as soon as hu signal is applied.

Step A6 The Call Agent returns an Ack

Step A7 The Call Agent sends a delete connection to the originating gateway with a request to go onhook.

Step A8 The Gateway Acks and provides performance information.

Step A9 The Call Agent sends a DLCX to the terminating gateway.

Step A10 The gateway acks with performance information

5.3. Example Call Setup - "MD" Package

The following describes Feature Group D call setup using the "MD" package. The table gives an overview of the initial part of the call flow with details to follow.

Steps	GW-o	CA	GW-t
A1	NTFY[O:md/sup] ->		
A2	<- Ack		
A3	NTFY[O:md/inf(<id>)] ->		
A4	<- Ack		
A5	NTFY[O:md/inf(<addr>)] ->		
A6	<- Ack		
B1	<- CRCX [M:recvonly, LCO, R: md/rel]		
B2	Ack[SDP1] ->		
B3		CRCX [M:sendrecv, LCO, SDP1] ->	
B4		<- Ack [SDP2]	
B5	<- MDCX [recvonly,SDP2]		
B6	Ack ->		

The assumption is that prior to the initial "notify", the Call Agent has sent a request to be informed of "sup" and "inf" events using quarantine handling "Q: loop".

Step A1 Trunk seizure results in a notify to the Call Agent indicating the start of a call setup:

```
NTFY 3001 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/sup
```

Step A2 The Call Agent sends an Ack.

Step A3 Once the digits for the identification field are collected the gateway notifies the call agent:

```
NTFY 3002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/inf(k0,0,0,4,0,8,5,5,5,1,2,3,4,s0)
```

Step A4 The Call Agent responds with an ack.

Step A5 When the digits are collected for the address field, another notify is sent:

```
NTFY 3003 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/inf(k0,5,1,2,5,5,5,4,5,6,7,s0)
```

Step A6 The Call Agent "acks"

Step B1 Create connection - receive only:

```
CRCX 2002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
C: A3C47F21456789F1
L: p:10, a:PCMU
M: sendrecv
X: 0123456789B1
R: md/rel
```

Step B2 The Gateway "acks" the request and provides connection ID and SDP information.

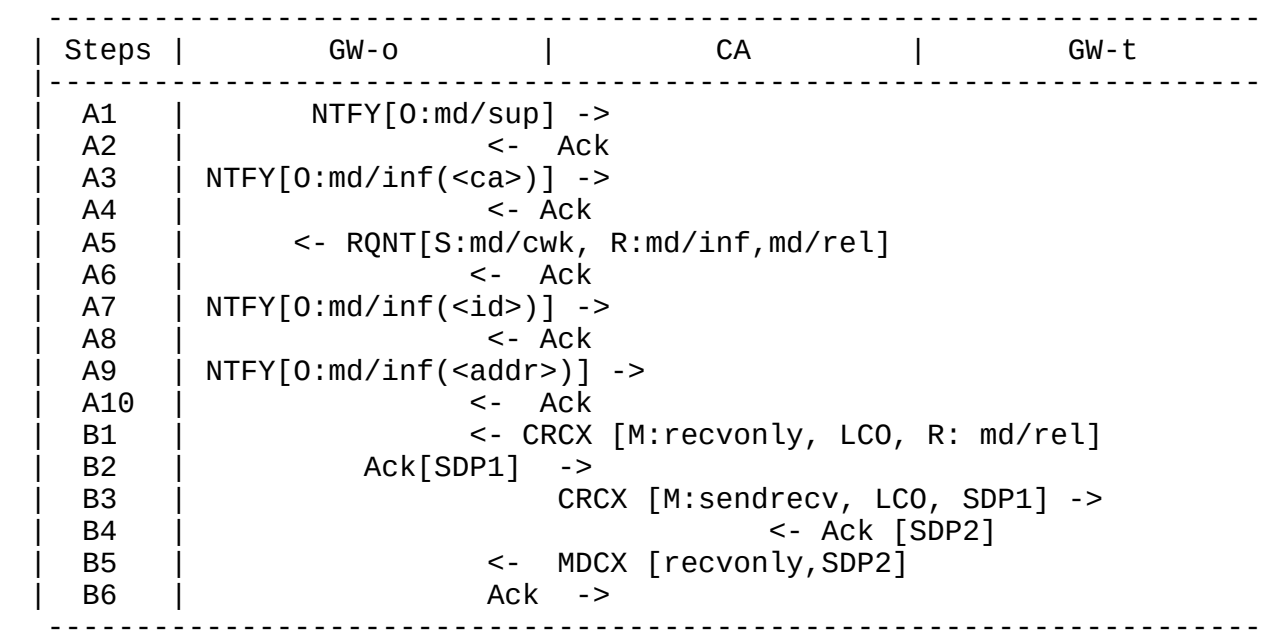
Step B3 The Call Agent passes this SDP information to the terminating gateway (GW-t) as part of the connection request.

Step B4 The terminating gateway, responds with an ack and its SDP information.

Step B5 Call Agent sends a modify connection request with connection mode receive-only to the origination gateway and includes the SDP information with the selected profile from the termination gateway.

Step B6 The Gateway Acks the modify connection request.

In the case of EAIN signaling there is some additional information provided so that this initial part of the call setup looks slightly different:



The assumption is that prior to the initial "notify", the Call Agent has sent a request to be informed of "sup" and "inf" events using quarantine handling "Q: loop".

Step A1 Trunk seizure results in a notify to the Call Agent indicating the start of a call setup:

```
NTFY 3001 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/sup
```

Step A2 The Call Agent sends an Ack

Step A3 The initial digit string contains the country address field:

```
NTFY 3002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/inf(k0,1,3,8,9,9,0,0,1,9,s0)
```

Step A4 The Call Agent responds with an ack

Step A5 The Call Agent does processing on the country address field and sends a request to initiate further input (sends a continue wink):

```
RQNT 2002 ds/*@mgw45.whatever.net MGCP 1.0
X: 0123456789B1
Q: loop
R: md/inf,md/rel
S: md/cwk
```

Step A6 The Gateway "acks" the request.

Step A7 Once the digits for the identification field are collected the gateway notifies the call agent:

```
NTFY 3003 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/inf(k0,0,0,4,0,8,5,5,5,1,2,3,4,s0)
```

Step A8 The Call Agent responds with an ack

Step A9 When the digits are collected for the address field, another notify is sent:

```
NTFY 3004 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/inf(k0,5,1,2,5,5,5,4,5,6,7,s0)
```

Step A10 The Call Agent "acks"

Step B1 Create connection - receive only:

```
CRCX 2002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
C: A3C47F21456789F1
L: p:10, a:PCMU
M: sendrecv
X: 0123456789B1
R: md/rel
```

Step B2 The Gateway "acks" the request and provides connection ID and SDP information

Step B3 The Call Agent passes this SDP information to the terminating gateway (GW-t) as part of the connection request.

Step B4 The terminating gateway, responds with an ack and its SDP information

Step B5 Call Agent sends a modify connection request with connection mode receive-only to the origination gateway and includes the SDP information with the selected profile from the termination gateway.

Step B6 The Gateway Acks the modify connection request

The following table shows the remainder of the call flow to set up the call for FGD EANA.

Steps	GW-o	CA	GW-t
C1	RQNT [S:sup, R:md/swk,md/oc,	md/rel,md/awk, md/ans] ->	
C2		<- Ack	
C3		<- NTFY [O:md/swk)]	
C4		Ack ->	
C5		<- NTFY [O:md/oc(md/sup)]	
C6		Ack ->	
C7		<- NTFY [O:md/awk)]	
C8		Ack ->	
C9	<- RQNT[S:md/awk]		
C10	Ack ->		
C11		<- NTFY [O: md/ans]	
C12		Ack ->	
C13	<- MDCX [M:sendrecv, S: md/ans, R: md/rel]		
C14	Ack ->		
C15		RQNT [R: md/sus, md/rel] ->	
C16		<- Ack	

Step C1 The Call Agent does a setup request to the terminating gateway The setup request for an MF EANA FGD interface will be the following:

```
RQNT 2001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
Q: loop
S:
md/sup(ct(nda),addr(k0,5,5,5,5,2,2,1,2,3,4,s0),id(k0,0,5,5,5,1,
2,3,4,s2))
R: md/swk,md/oc,md/rel,md/awk,md/ans
```

Note that the result of the "sup" signal is the following sequence on the interface to the PBX:

* off-hook -> SCN
* wink <- SCN
* caller ID digits sent to SCN
* address digits sent to SCN

Step C2 The gateway responds with an ack

Step C3 "Notify" the CA that the start of signaling has occurred (incoming wink start has occurred) i.e.:

```
NTFY 3000 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/swk
```

Step C4 The Call Agent "acks".

Step C5 "Notify" that out-pulsing is complete:

```
NTFY 3001 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/oc(md/sup)
```

Step C6 The Call Agent "acks".

Step C7 "Notify" that the acknowledgement wink has been received:

```
NTFY 3002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/awk
```

Step C8 The Call Agent "acks".

Step C9 The acknowledge wink is passed to the originating gateway:

```
RQNT 2001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375842
S: md/awk
R: md/rel
```

Step C10 GW-o "acks".

Step C11 "Notify" off-hook event (the person at the other end has answered):

```
NTFY 3003 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B0
O: md/ans
```

Step C12 The Call Agent "acks".

Step C13 The Call Agent now sends a request to make the connection full duplex and indicates that the other end has answered the phone (S: ans sent)

Step C14 The Gateway acks the request

In the case of FGD EAIN, there is an additional digits string (country address and/or carrier access code that has to be included so that step C1 looks like the following in a case where there is no overlapped sending:

```
RQNT 2001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
X: 45375841
Q: loop
S:md/sup(ct(nta),ca(k0,1,3,8,9,9,0,0,1,0,s0),id(k0,
0,5,5,5,1,2,3,4,s0),addr(ko,0,1,1,3,8,1,2,3,4,7,6,5,s0))
R: md/swk,md/oc,md/rel,md/awk,md/ans
```

If overlapped sending is done, only the country address and caller ID digit strings are sent out in step C1:

```
RQNT 2001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0

X: 45375841
Q: loop
S:md/sup(ct(nta),ca(k0,1,3,8,9,9,0,0,1,0,s0),id(k0,0,
5,5,5,1,2,3,4,s0))
R: md/swk,md/oc,md/rel,md/ans
```

Then after these digits go out indicated by event "oc(sup)" in step C5, and as soon as the Call Agent has the address information, it sends it out using the "inf" signal:

```
RQNT 2002 ds/ds1-3/6@mgw45.whatever.net MGCP 1.0
X: 0123456789B1
Q: loop
R: md/oc,md/rel,md/awk,md/ans
S: md/inf(ko,0,1,1,3,8,1,2,3,4,7,6,5,s0)
```

The Call Agent will then get a further "md/oc(md/sup)" event when these digits have gone out.

Step C15 The Call Agent requests to be told of on-hook ("sus") events
or abnormal release ("rel") events.

Step C16 The gateway "acks" the request.

5.4. Example Call Setup - "MO" Package

The following describes Feature Group D operator services signaling call setup (911 call) using the "MO" package. The table gives an overview of the initial part of the call flow with details to follow. In this case GW-o is a residential gateway using the line package and GW-t connects to the E911 tandem.

Steps	GW-o	CA	GW-t
A1	NTFY[O:hd] ->		
A2	<- Ack		
A3	<- RQNT S: dl, R: [0-9*#T](D)		
A4	Ack ->		
A5	NTFY[O: 9,1,1] ->		
A6	<- Ack		
B1	<- CRCX [M:recvonly, R: hu]		
B2	Ack[SDP1] ->		
B3	CRCX [M:sendrecv, LCO, SDP1, S: mo/sup] ->		
B4		<- Ack [SDP2]	
B5		<- NTFY [O: oc(sup)]	
B6		Ack ->	
B5	<- MDCX [sendrecv,SDP2]		
B6	Ack ->		

Note: the originating side in this case is a line-side gateway.

Step A1 The user goes off-hook:

```
NTFY 3001 aaln/1@gw-o.whatever.net MGCP 1.0
X: 0123456789AF
O: l/hd
```

Step A2 The Call Agent sends an Ack:

```
200 3001 OK
```

Step A3 The Call Agent sends dial-tone and requests that digits be collected:

```
RQNT 2001 aaln/1@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
S: l/dl
R: d/[0-9#*T](D), hu
```

Step A4 The gateway responds with an ack:

```
200 2001 OK
```

Step A5 Once the digits are collected the gateway notifies the Call Agent. In this case, it is a 911 call

```
NTFY 3002 aaln/1@gw-o.whatever.net MGCP 1.0
X: 0123456789B0
O: d/9,d/1,d/1
```

Step A6 The Call Agent responds with an ack:

```
200 3002 OK
```

Step B1 The Call Agent now requests that a receive-only connection be made.

```
CRCX 2002 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
C: A7453949499
L: a:PCMU,s:off,e:on
M: recvonly
X: 0123456789B1
R: l/hu.
```

Step B2 The Gateway acks with a connection ID and provides the SDP information:

```
200 2002 OK
I: 23474FE

v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B3 The Call Agent passes this SDP information to the terminating gateway (GW-t) as part of the connection request and does a call setup request at the same time:

```
CRCX 4001 ds/ds1-5/3@gw-t.whatever.net MGCP 1.0
C: A7453949499
X: 45375840
L: a:PCMU,s:off,e:on
M: sendrecv
Q: loop
R: oc, rel, orbk
S: sup(addr(k0,9,1,1,s2),id(k0,0,8,3,4,5,6,7,8,s0))

v=0
o=- A7453949499 0 IN IP4 128.96.41.1
s=-
c=IN IP4 128.96.41.1
t=0 0
m=audio 3456 RTP/AVP 0
```

As a result of this request, the following signaling interactions will occur between GW-t and the Switched Circuit Network (SCN - in this case, the E911 tandem):

* Off-hook -> SCN
* Wink <- SCN
* k0,9,1,1,s2 -> SCN
* Off-hook <- SCN
* k0,0,8,3,4,5,6,7,8,s0

Note that off-hook from the SCN is part of the protocol (a request for the caller ID) and does not provide an indication of whether the operator answered or not.

Step B4 The terminating gateway, responds with an ack and its SDP information:

```
200 4001 OK
I: 34738A

v=0
o=- A7453949499 0 IN IP4 47.123.34.33
s=-
c=IN IP4 47.123.34.33
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B5 The Call Agent will get a further notify when outpulsing of all of the digits is complete.

```
NTFY 3003 aaln/1@gw-o.whatever.net MGCP 1.0

X: 45375840
O: oc(sup)
```

Step B6 The Call Agent returns an "ack"

```
200 3003 OK
```

Step B7 Call Agent sends a modify connection request with connection mode receive-only to the origination gateway and includes the SDP information with the selected profile from the termination gateway.

```
MDCX 2003 ds/ds1-3/6@gw-o.whatever.net MGCP 1.0
C: A7453949499
I: 34738A
M: sendrecv

v=0
o=- A7453949499 0 IN IP4 47.123.34.33
s=-
c=IN IP4 47.123.34.33
t=0 0
m= audio 3456 RTP/AVP 0
```

Step B8 The Gateway Acks the modify connection request

```
200 2003 OK
```

The call is now established between the user and the operator.

Acknowledgements

The source for some these packages are Flemming Andreasen, Wai-Tak Siu - Cisco Systems, and Don Stanwyck - IP Unity. Special thanks to Joe Clark, Telcordia Technologies for his CAS interface expertise. Also thanks to all the reviewers for all their comments, including but not limited to the following people: Charles Eckel, Cisco Systems; Jerry Kamitses, Sonus Networks.

References

[1] Arango, M., Dugan, A., Elliott, I., Huitema, C. and S. Pickett, "Media Gateway Control Protocol (MGCP) Version 1.0", RFC 2705, October 1999.

[2] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[3] Bellcore, Compatibility Information for Feature Group D Switched Access Service, TR-NPL-000258, Issue 1, October 1985.

[4] Bellcore, Interoffice LATA Switching Systems Generic Requirements (LSSGR): Verification Connections (25-05-0903), TR-TSY-000531, Issue 2, July 1987.

[5] Bellcore, LSSGR: Signaling for Analog Interfaces, GR-506-CORE, Issue 1, June 1996.

[6] PacketCableTM PSTN Gateway Call Signaling Protocol Specification, Pkt-SP-TGCP-I01-991201

Author’s Address

Bill Foster
170 West Tasman Dr
San Jose, CA, 95134

Phone: 408-527-8791
EMail: bfoster@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.