# Color Ring Back Tone — Building Feature-Rich Wireless Applications with Dialogic® Signaling Solutions

## Executive Summary

This application note describes the implementation of a Color Ring Back Tone (CRBT) wireless service. Focusing on the technologies and signaling involved, this document demonstrates how service providers can integrate CRBT into an existing Signaling System 7 (SS7) wireless application using Dialogic® Signaling Distributed Architecture (SigDiA) enabled building blocks.

# Table of Contents

## Introduction

Instead of hearing the familiar "ring" tone when calling someone, Color Ring Back Tone (CRBT) lets wireless service subscribers play their favorite tune, sound, jingle, or phrase to incoming callers. With the ability to choose the ring tone a caller will hear, subscribers have an instant way to express their own individuality — and enterprises have the option to provide a customized message to the calling party.

An important distinction between CRBTs and other services is that this network-based solution is independent of both handset and Subscriber Identity Modules (SIMs). Users are not required to have the latest phone or a specific brand of phone. Therefore, operators have the flexibility to offer personalized ring back services to all types of subscribers (for example, postpaid or prepaid, fixed or mobile). Wireless service providers can generate revenue from service management, using a wealth of existing subscriber data.

## Objectives

This application note discusses three CRBT scenarios, and presents the signaling information to develop such services using call control and Intelligent Network (IN) SS7 protocols. It is assumed that the reader has some knowledge of Intelligent Networking (IN) and SS7-based applications.

The section *Information for System Design* describes ways to implement a CRBT service using SS7 protocols running on Dialogic® board and system-level platforms.

The following CRBT call scenarios are discussed in the next section:

- The first scenario discusses an ISDN User Part (ISUP) call control method, highlighting how an existing SS7 ISUP platform could be used as a starting point to develop such a service

- The second scenario builds upon the implementation in Scenario 1 and discusses IN signaling, which gives the service provider a method of controlling which incoming calls are able to access the CRBT service. Here, an IN-based CRBT implementation is described, as well as how this signaling approach can be integrated with an existing ISUP call control implementation.

- The third scenario shows how — and why — more complex IN signaling can be used. This scenario

overcomes "tromboning" issues in a network (inherent in an ISUP call control solution) and lets network operators offer additional high-value IN services during the duration of the call.

## CRBT Service Scenarios

### Scenario 1 — ISUP Call Control Implementation

In this scenario, the network is configured to route incoming calls for a specific called party to the CRBT tone server. The CRBT tone server is a network element that can:

- Answer an incoming SS7 call
- Play a specific CRBT to the caller (based upon the calling and called party number)
- Make an outgoing call to the called party number while the tone plays
- Disconnect the CRBT as the called party answers
- Connect the incoming and outgoing calls together

**Note:** The design of the tone server itself is outside the scope of this application note.

As shown in Figure 1, the caller dials a called party who has subscribed to the service (1). The incoming call reaches a Service Switching Point (SSP), which routes it via an SS7 ISUP trunk to the tone server. The tone server answers the incoming call, and plays the correct custom CRBT to the calling party (2), based upon the calling and called party number received in the incoming SS7 Initial Address Message (IAM).

As the tone server plays the custom ring tone to the calling party, it makes an outgoing call (3) to the called party (from the called party number information in the received IAM). This call is monitored by the tone server, and once the call is answered, the server stops playing the custom ring tone and connects the incoming and outgoing calls (4).

Note that two trunks have been used to route the call to and from the tone server; this is known as "tromboning," which can occur as a call is routed through a circuit-switched network. If tromboning happens during routing, a network pathway between two or more points will be unnecessarily used (looped) in both the incoming and outgoing directions. Tromboning is generally undesirable in the network, as it wastes trunk resources between network nodes. It also reduces the total number of calls that could be serviced simultaneously by the tone server.
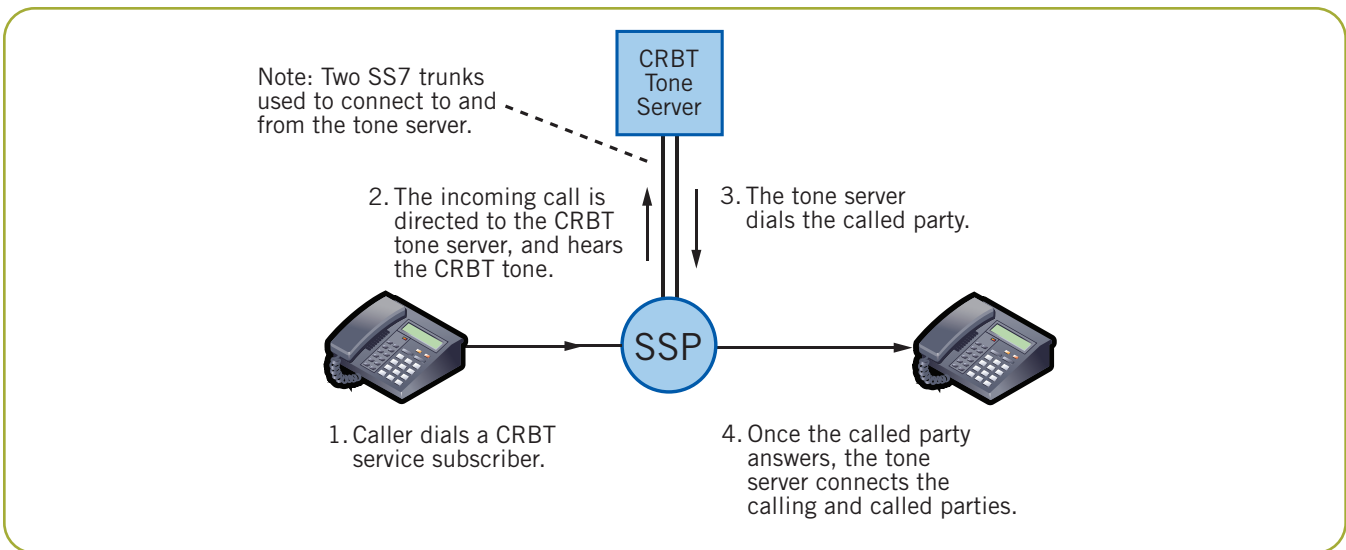
*Figure 1. ISUP Call Control Implementation for CRBT*

For more information on tromboning, see Appendix A.

### Scenario 2 — Introducing IN Signaling

Scenario 1 provides no explanation as to which incoming calls should be routed to the CRBT tone server. It is logical that incoming calls should be routed to the tone server *only* if the called party's number is an active subscriber to the CRBT service. In Scenario 1, in order to route calls (to the subscriber) via the tone server, datafill changes can be made to the network SSP for every CRBT service subscriber. This is time-consuming and costly for network operators to implement.

Using Intelligent Networking (IN) [ETS 300 374-1], it is possible to centralize subscriber information on an external database away from the core switching platform, allowing a CRBT service provider to maintain a record of current subscribers.

In Scenario 2, Figures 2 and 3 show how IN signaling can be integrated into the ISUP CRBT solution (that is, Scenario 1) to determine how incoming calls should be handled.

As in Scenario 1, the caller dials a called party who has subscribed to the service. The mobile subscriber's incoming call to the SSP, in this case, hits a Trigger Detection Point (TDP) datafilled within the SSP. The SSP detects the TDP request and sends an INAP **InitialDP** message to a Service Control Point (SCP) requesting instructions to complete the call [ETS 300 374-1].

**Note:** For AIN (Advanced Intelligent Network) implementations, similar call logic could be achieved using the AIN messages **Info_Analyzed** or **Info_Collected** instead of the message INAP **InitialDP**.

**Note:** The SSP can be datafilled to cause incoming calls to trigger towards the IN platform (SCP) or only calls to a specific range of mobile numbers. This involves far less datafill changes to implement on an SSP than in Scenario 1, and could be a "once only" operation for the network operator when the service is initially implemented.
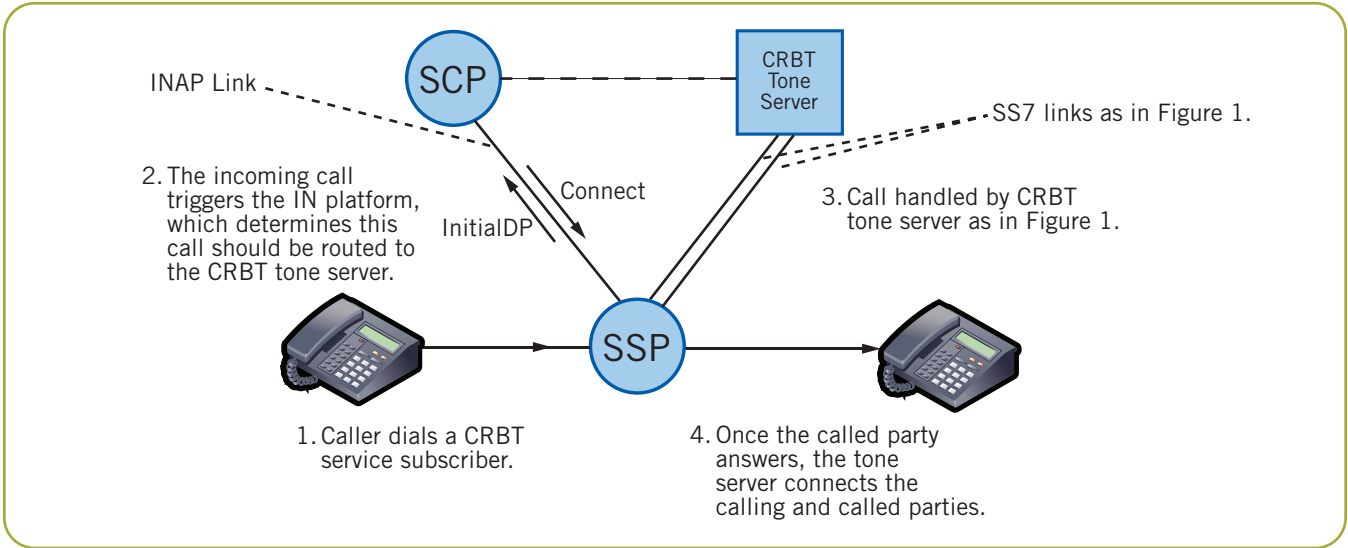
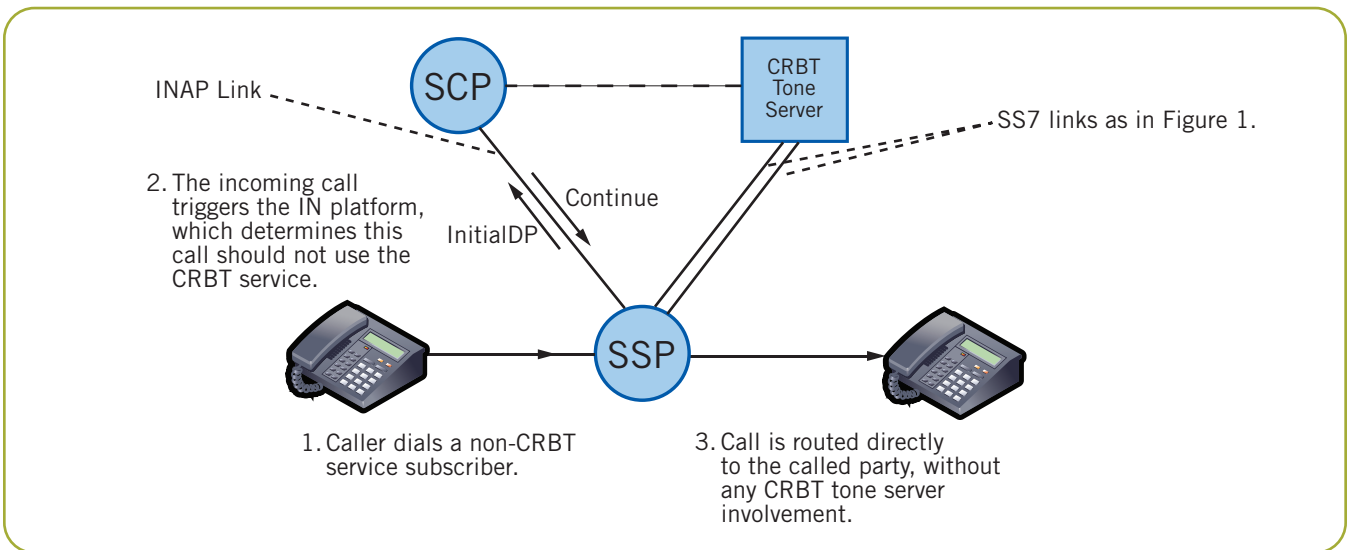*Figure 2. IN Signaling Used to Route Calls to CRBT Subscribers*



*Figure 3. IN Signaling Used to Route Calls to Non-CRBT Subscribers*

The SCP uses onboard service logic to determine how the call should be handled. Using the called party number parameter contained in the **InitialDP** message, the CRBT database (shared with the CRBT tone server) is checked to determine if the called party is a current CRBT service subscriber. The two ways to how to handle the call are:

1.  Figure 2 shows that if the called party is a CRBT service subscriber, the SCP responds to the SSP with an INAP **Connect** message [ETS 300 374-1]. Since there are various parameters defined in the INAP specifications that can be included in the **Connect** message, there are various possible ways to implement routing to the tone server.

**Note:** The AIN message **Analyze_Route** could be used instead of the INAP **Connect** message.

The call is then treated as in Scenario 1, and the tone server plays the custom ring tone to the calling party as the called party is dialed.

2.  Figure 3 shows that if the called party is not a CRBT service subscriber, an INAP **Continue** message [ETS 300 374-1] is returned to the SSP. The INAP **Continue** message causes the SSP to continue analyzing the originally received digits and route the call onward, without involving the CRBT tone server.

As shown in Figures 2 and 3, the IN SCP and CRBT tone server can be implemented separately using a
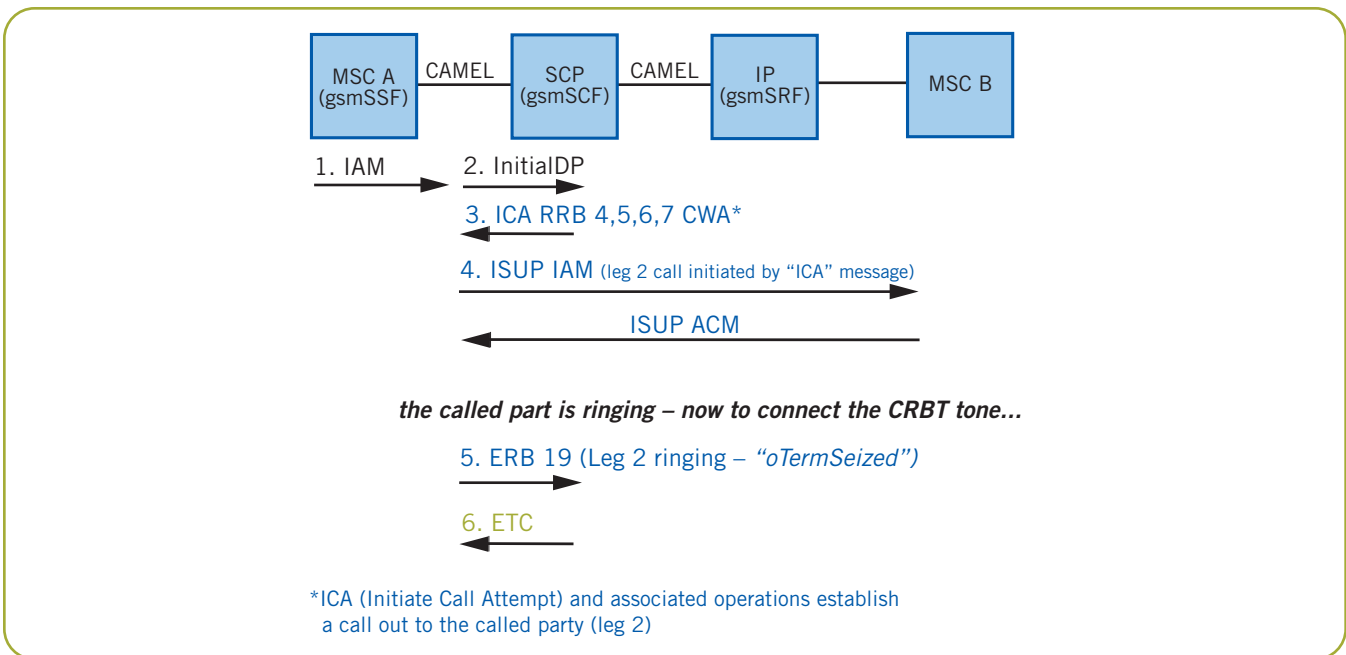
*Figure 4. CRBT with IN Call Party Handling (Part 1 of 3)*

distributed architecture. It is also possible for these two functions to be merged as a single unit.

As in Scenario 1, tromboning of circuits to and from the tone server remains an issue. However, the impact on the number of calls that could be handled is reduced, as only calls to subscribers actually using the CRBT service will be routed via the tone server.

**Note:** The section, *Information for System Design,* describes how to add IN signaling to an existing call control implementation.

**Scenario 3 — CRBT with IN Call Party Handling**

Scenarios 1 and 2 show two SS7 circuits are needed for every call accessing the CRBT tone server. This is inefficient and can make the service unnecessarily expensive. Scenario 3 proposes a solution using CAMEL Application Part (CAP) Phase 4 signaling [ETSI TS 129 078] and overcomes the need for tromboning when implementing a CRBT service.

**Note:** In order to implement this method, the connected MSC (or SSP) must be able to support CAMEL Phase 4 (or INAP CS2) Call Party Handling operations. Figures 4 through 6 show how IN messaging is involved in this implementation.

In Figures 4 through 6, an incoming call is received at an MSC, destined for a mobile handset served by Mobile

Switching Center B (**MSC B**). The called mobile subscriber has the CRBT feature activated.

IN Signaling is used in this CRBT application example as follows:

1.  In Figure 4, an incoming call is received at MSC A.

2.  This incoming call triggers towards the SCP as in the previous examples.  The CRBT service running at the SCP determines how the call should now be handled. *(The SCP conducts a database check to determine if the called party is a subscriber to the CRBT service.)*

3.  The SCP instructs MSC A to make an outgoing call leg directly to the called party.  An IN **Initiate Call Attempt** [ETSI TS 129 078] message is used.

4.  Figure 4 shows the outgoing call leg 2 (highlighted in blue) made from MSC A to the called party due to the ICA message. This call leg is monitored by the SCP to detect network events such as the call ringing and being answered.

5.  Once the called party starts to ring, the SCP is notified with an IN **Event Report BCSM (Basic Call State Machine) Event** (ERB) message indicating event "oTermSeized."

6.  Once it is known that the called party is ringing, an IN **Establish Temporary Connection** (ETC) operation is sent to MSC A, to connect the calling
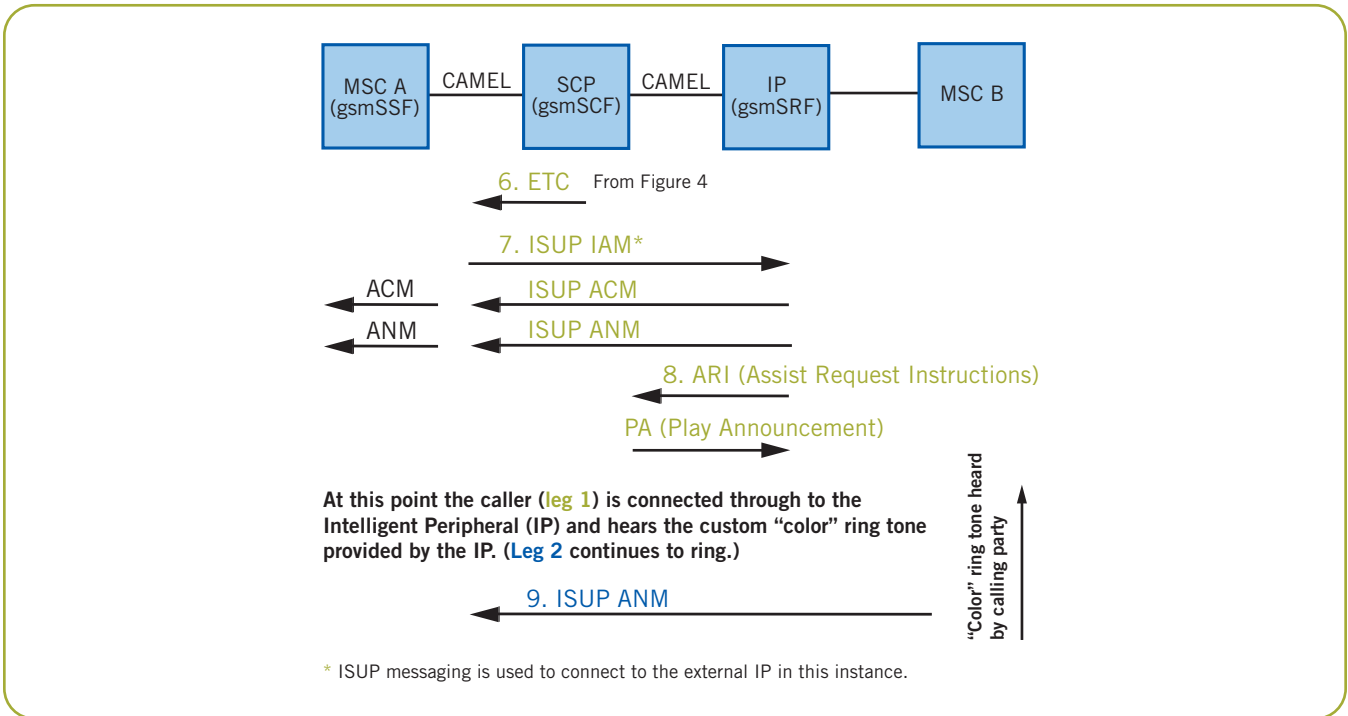
Figure 5. CRBT with IN Call Party Handling (Part 2 of 3)

party to a CRBT tone server. (In this example, an IN Intelligent Peripheral [IP] behaves as the tone server.)

7.  The signaling sequence in Figure 5 (highlighted in green) shows ISUP signaling used to connect to the CRBT tone server (IP).

8.  Once the IP has answered the incoming call, the SCP is notified using an IN **Assist Request Instructions** (ARI) message. The IP is directed by the SCP to play the correct CRBT tone to the caller via a **Play Announcement** (PA) message.

    At this point, one call (highlighted in green) is the original calling party listening to the custom ring tone. A second call (highlighted in blue in Figure 4) has been made directly from MSC A, ringing the called party.

9.  The called party answers, and now the custom ring tone should be removed and the two separate call legs joined.

10. As shown in Figure 6, once the called party answers the second call leg, the SCP is notified with the IN **Event Report BCSM (Basic Call State Machine) Event** (ERB) message.

11. This notification causes the SCP to disconnect the call between the calling party and the CRBT tone server. (The IN message **Disconnect Forward Connection With Argument** (DFCWA) [ETSI TS 129 078] is used to achieve this.)

12. Finally, the IN message MoveLeg [ETSI TS 129 078] is sent to MSC A, causing the two separate call legs to be joined. Hence, the called and calling parties are directly connected together in speech state, with no tromboning of ISUP circuits. (For INAP CS2 implementations **Merge Call Segments** [ETS 301 140-1] can be used in place of MoveLeg.)

Such an implementation could be adopted to make full use of the ISUP circuits provisioned to the CRBT tone server. This implementation could also use a centralized subscriber database shared by the IN platform (SCP) and the tone server, holding details of all the current service subscribers.
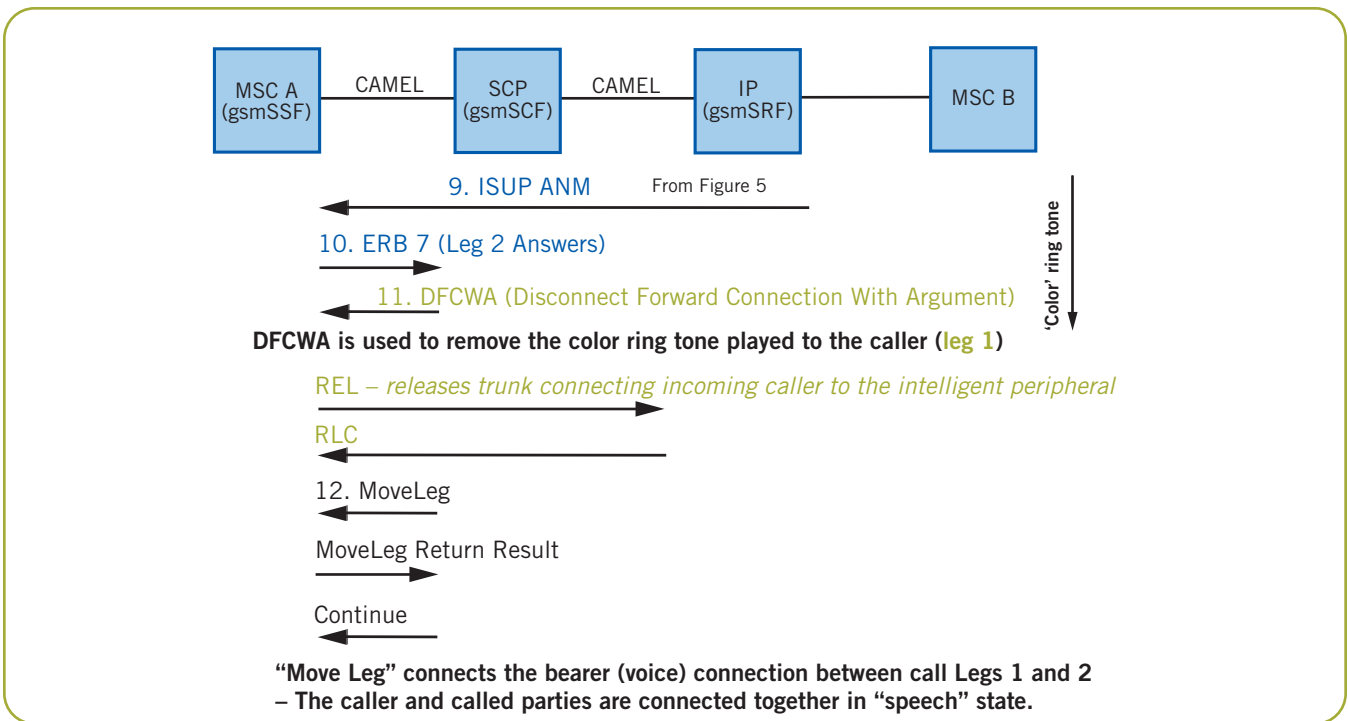
*Figure 6. CRBT with IN Call Party Handling (Part 3 of 3)*

## Information for System Design

### Working with ISUP Signaling – Notes on Implementing Scenario 1

The Call Test Utility (CTU) from Dialogic® *Call Test Utility (CTU) User Guide* (see the *For More Information* section) demonstrates how an application can be built which receives and answers an incoming ISUP call, and follows this by making an outgoing call. This functionality could be noted when developing a CRBT tone server operating over ISUP trunks.

Further development (outside the scope of this application note) would be needed to create a complete ISUP tone server solution that would carry out the following tasks:

- Answer an incoming SS7 call

- Play a specific CRBT to the caller (based upon the called party number in the received ISUP IAM message)

- As this tone is playing, make an outgoing call to the called party number (CTU demonstrates an outgoing call being made)

- Disconnect the CRBT and connect the incoming and outgoing calls together, as the called party answers

### Introducing IN Signaling – Notes on Implementing Scenario 2

The example INAP Test Utility (INTU) in the *Dialogic® INAP Test Utility (INTU) User Guide* (see the *For More Information* section) demonstrates how to implement an application that can accept and respond to an INAP or CAMEL query, as shown in Figure 7. This signaling example is similar to that outlined in *Scenario 2 — Introducing IN Signaling.*

INTU can be used as a basis to create an application that detects which incoming calls should be routed to the CRBT server. The INTU application checks the Called Party Number parameter of an incoming INAP **InitialDP** message, and simulates a database lookup, checking if the incoming called party number matches that of a known CRBT service subscriber. If the called party number doesn't match that of a known CRBT subscriber, the call should continue as usual and not be routed to the tone server.

Figure 8 shows the required INAP signaling where an incoming call is bound for an existing CRBT subscriber (Figure 8a) and where an incoming call is for a non-subscriber (Figure 8b).
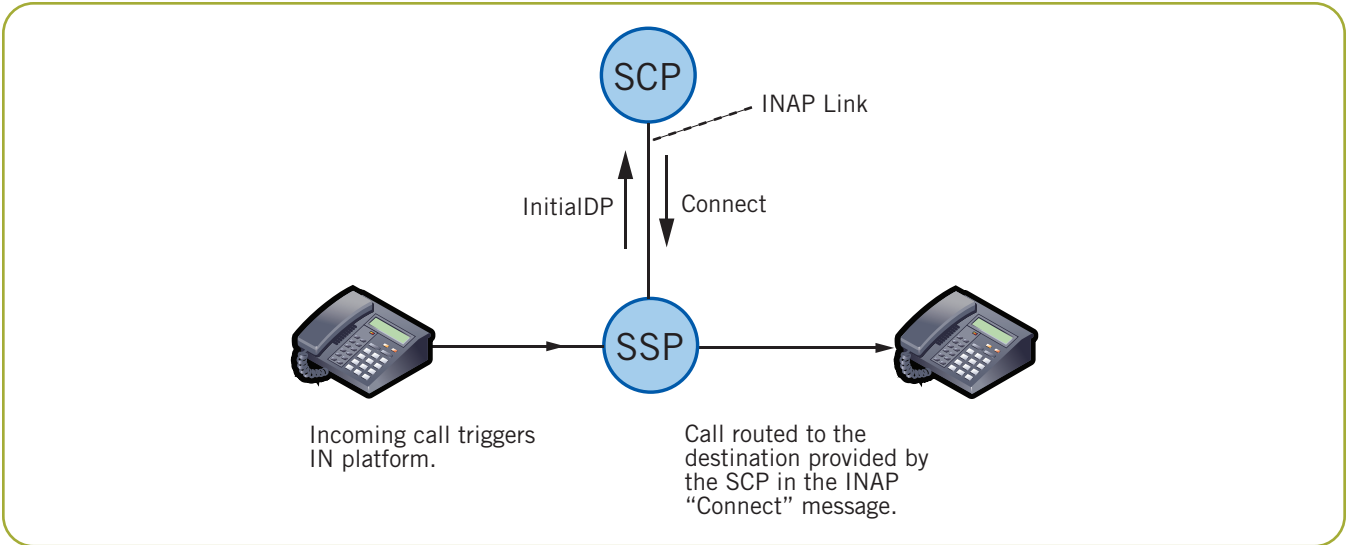
*Figure 7. INAP Test Utility (INTU) Example (see the INAP Test Utility (INTU) User Guide)*
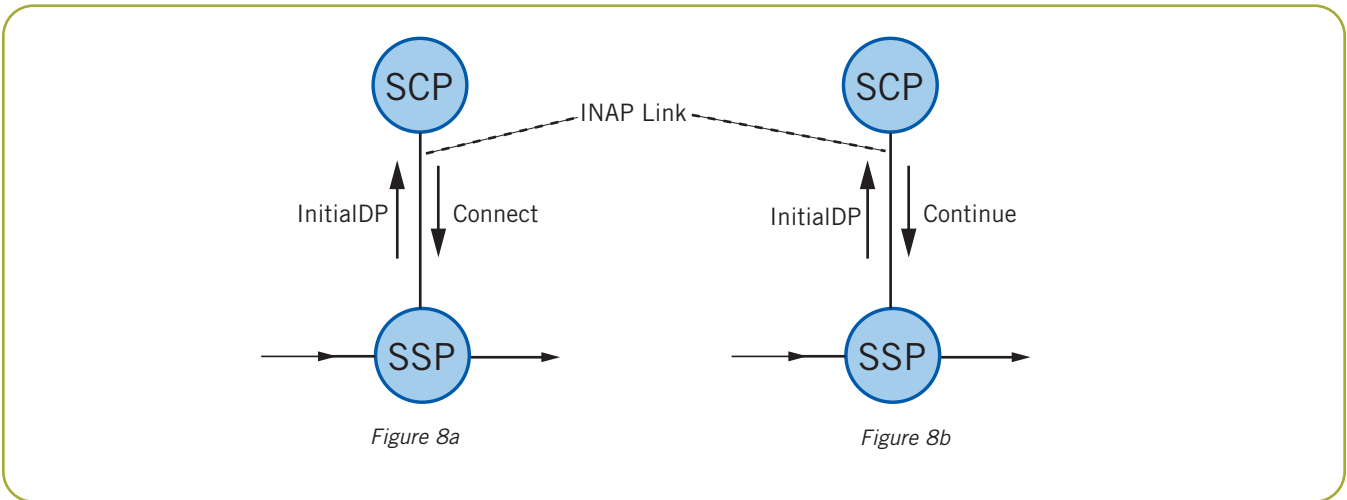


*Figure 8. INAP Signaling for a Call to an Existing CRBT Subscriber (Figure 8a) and a Call to a Non-Subscriber (Figure 8b)*

To enable the case where an INAP **Continue** message is returned (if the call is for a non-CRBT subscriber, shown in Figure 8b), the source code for the INTU application should be modified. This is necessary because the INTU tool, by default, is programmed to reject calls to non-subscriber numbers using the INAP **ReleaseCall** operation instead of the **Continue** operation. The following section describes the code changes needed to the INTU source code (*Dialogic® INAP Test Utility (INTU) User Guide*, Section 2.2.3, "Invalid Parameter").

**Code Modifications to the INTU Application**

The source code for the INTU example application is available as part of the *SS7 User Part Development Package* available at the following web site:

http://www.dialogic.com/support/helpweb/signaling/software3.htm

If a received **InitialDP** message contains a called party number that does not match that of a known CRBT subscriber, the code changes detailed here allow the INTU application to respond with an INAP **Continue** message.

Code changes are needed to the file `intu.c`. The following section shows extracts of the `intu.c` file, showing the original code (`intu.c` in black) and changed lines (`intu_continue_edition.c`) in blue.

(The line numbers shown are for approximate reference.)

This is the process:

1. Copy the `intu.c` file and rename it to maintain a backup of the original file.

2. Modify the `intu.c` file, changing the lines as detailed in this section.

3. Recompile the modified INTU program.

4. Run the INTU test cases as detailed in the *INTU User Guide.* Check that the INAP **Continue** operation is correctly returned in case the **InitialDP** Called Party Number parameter does not match the test number 0800123456, as detailed in `intu.c` code file.

```
Excerpt from intu.c code file:
/*
 * Example numbers to use in number translation
 * e.g. 0800123456 translated to -> 01425651300
 */
static u8 example_freephone_num[]    = {0,8,0,0,1,2,3,4,5,6};
static u8 example_dest_routing_num[] = {0,1,4,2,5,6,5,1,3,0,0};
```

### intu.c – Code Changes (Part 1 of 3)

```
***** intu.c
 1047:          * We couldn't translate, but the dialogue itself is still successful.
 1048:          * Now build and send a ReleaseCall...
 1049:          */
 1050:
 1051:         IN_set_component_param (INPN_Cause, sizeof(example_release_cause),
 1052:                                          example_release_cause, &og_cpt);
 1053:         IN_set_operation(INOP_ReleaseCall, INTU_ReleaseCall_timeout, &og_cpt);
 1054:


***** intu_continue_edition.c
 1047:          * Couldn't find the CRBT subscriber, but the dialogue itself is
 1048:          * still successful.  Now build and send a Continue message..
 1049:          */
 1050:
 1051:         IN_set_operation(INOP_Continue, INTU_Connect_timeout, &og_cpt);
 1052:
```

**intu.c – Code Changes (Part 2 of 3)**

```
***** intu.c
 1061:
 1062:          INTU_send_close(ic_dlg_id, INAPRM_normal_release);
 1063:          INTU_close_dialogue(ic_dlg_id, dlg_ptr, h, INTU_DLG_SUCCESS);
 1064:          return(0);

***** intu_continue_edition.c
 1061:
 1062:          INTU_send_delimit(ic_dlg_id);
 1063:          INTU_send_close(ic_dlg_id, INAPRM_prearranged_end);
 1064:          INTU_change_state(ic_dlg_id, dlg_ptr, INTU_STATE_CLOSING);
 1065:          return(0);
```

**intu.c – Code Changes (Part 3 of 3)**

```
***** intu.c
 1291:          if (intu_options & INTU_OPT_TR_NUM_TRANS)
 1292:            printf("], Releasing Call\n");
 1293:

***** intu_continue_edition.c
 1291:          if (intu_options & INTU_OPT_TR_NUM_TRANS)
 1292:            printf("], Call continues with no translation\n");
 1293:
```

## Dialogic® Building Blocks for an SS7 CRBT Service

Dialogic® Signaling Distributed Architecture (SigDiA) enabled building blocks can be used to create network elements, such as the SCP or IP, that are required to develop an IN-based CRBT service.

For SS7 ISUP call control and INAP/CAMEL/WIN Intelligent Networking, a number of building blocks are available, including the following:

- Dialogic® SS7AM1 SS7 Interface Board
  (http://www.dialogic.com/products/signalingip_ss7components/Signaling_Boards_SS7AM1.htm)
- Dialogic® Signaling Server with SIU Option (Dialogic® SS7G21 and SS7G22 Signaling Servers)
  (http://www.dialogic.com/products/signalingip_ss7components/signaling_servers_and_gateways.htm)
- Dialogic® SS7HDC SS7 Interface Board (cPCI)
  (http://www.dialogic.com/products/signalingip_ss7components/Signaling_Boards_SS7HDC.htm)
- Dialogic® SS7HDP SS7 Interface Board (PCI)
  (http://www.dialogic.com/products/signalingip_ss7components/Signaling_Boards_SS7HDP.htm)
- Dialogic® SPCI4 SS7 Interface Board
  (http://www.dialogic.com/products/signalingip_ss7components/Signaling_Boards_SPC.htm)

A choice of products is available based upon the level of integration needed, the circuit density, and the level of fault tolerance required. The same Application Programming Interface (API) is used across the entire SS7 product family, allowing easy migration from board-based to server-based platforms as service needs change. Further information on the range of Dialogic® Signaling product building blocks is available on the web at http://www.dialogic.com/products/signalingip_ss7components/signaling_software.htm.

The following Dialogic® boards can be used to build the media-handling functionality of a CRBT tone server:

- Dialogic® Multimedia Platform (MMP) for AdvancedTCA (ATCA)
  - MMP for ATCA is a powerful product that can be used to deliver applications, such as voice and video mail, color ring back tones, unified messaging, and audio conferencing over IP and PSTN interfaces in wireline and wireless environments using standard protocols for session and media control (http://www.dialogic.com/products/ip_enabled/MM_ATCA.htm)

- Dialogic® Host Media Processing (HMP) Interface Boards (T-1/E-1) for Dialogic® HMP Software
  - Dialogic® DNI/300TEPHMP Digital Network Interface Board
  - Dialogic® DNI/601TEPHMP Digital Network Interface Boards
  - Dialogic® DNI/1200TEPHMP Digital Network Interface Boards
  - Dialogic® DNI/2410TEPEHMP Digital Network Interface Boards (http://www.dialogic.com/products/ip_enabled/DNI_HMP_DM3.htm)

- Dialogic® Signaling Boards (T-1/E-1)
  - Dialogic® SS7HDC SS7 Interface Board, a CompactPCI line interface board providing 16 T1/E1 ports (provided by a companion Rear Transition Module) (http://www.dialogic.com/products/signalingip_ss7components/Signaling_Boards_SS7HDC.htm)

- Dialogic® Digital Telephony Interface Boards
  - Dialogic® DM/N160TEC Digital Telephony Interface Board, a CompactPCI digital telephony interface board (16 T1/E1 trunks per slot) (http://www.dialogic.com/products/tdm_boards/signaling/DMN_160_Boards_DM3.htm)
  - Dialogic® DM/N960-4T1 Digital Telephony Interface Board
  - Dialogic® DM/N1200-4E1 Digital Telephony Interface Board (PCI or CompactPCI) (http://www.dialogic.com/products/tdm_boards/media_processing/DMN_4T1_Boards_DM3.htm)

- Dialogic® DM3 Media Boards
  - Dialogic® DM/V3600BP Media Board
  - Dialogic® DM/V4800BC Media Board available in H.100 (PCI) and H.110 (CompactPCI) compliant universal form factors (http://www.dialogic.com/products/tdm_boards/media_processing/DMVB_Boards_DM4.htm)

The Dialogic® Global Call API can be used to develop a CRBT service incorporating signaling and media control. A manual entitled *Global Call SS7 Technology Guide* is available on the web (see the *For More Information* section), detailing how SS7 signaling can be used with Dialogic® media products to enable a system designer to build an application platform (such as a CRBT tone server) where media is controlled by SS7 signaling.

## Summary

This application note described the SS7 Color Ring Back Tone (CRBT) service and how such a service can be provided using call control (ISUP) or Intelligent Network (CAMEL or INAP) signaling.

The application note detailed how IN signaling can be introduced to a call control service implementation, letting such an implementation use available ISUP links (connecting to a tone server) more efficiently.

Finally, a complete IN signaling scheme was discussed, removing the issues of ISUP trunk usage (tromboning) from the service implementation. Using such a scheme should be considered, with the potential costs (that is, availability of network nodes supporting the necessary IN protocols and a longer development time for the complete IN signaling application) weighed against the overall service benefit of efficient ISUP trunk usage for CRBT calls.
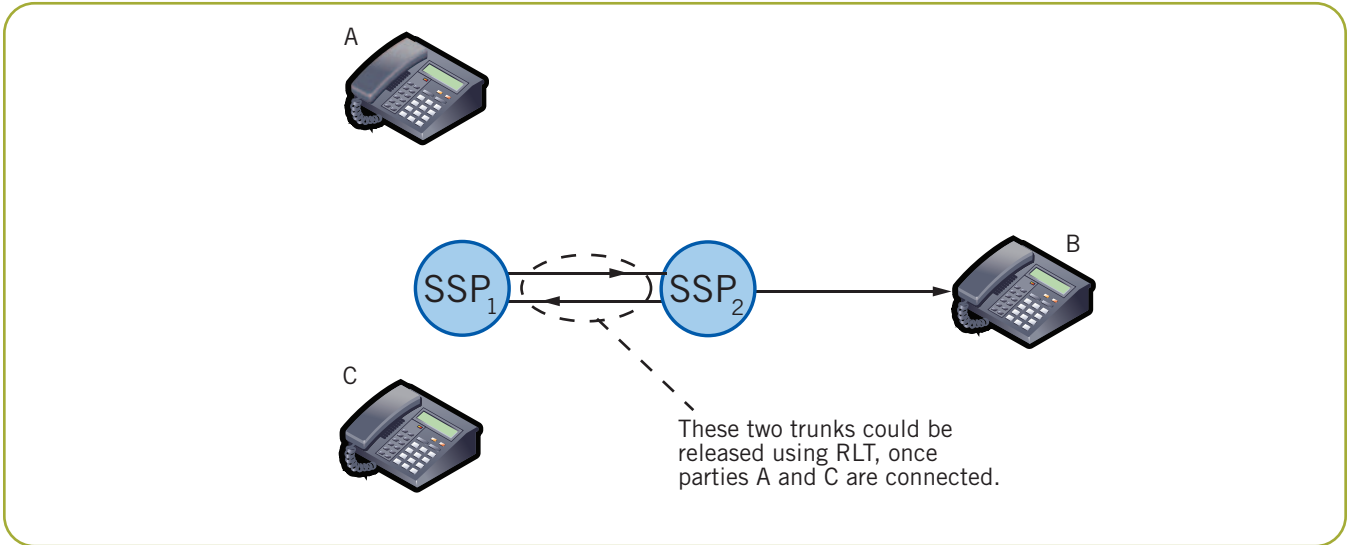
A

B

C

These two trunks could be
released using RLT, once
parties A and C are connected.

*Figure 9. SS7 Network "Tromboning" Example*

## References

[ETS 300 374-1] ETSI Intelligent Network Capability
Set 1 (CS1) ETS 300 374-1, *INAP Programmer's Manual*,
available at http://webapp.etsi.org/key/queryform.asp

[ETS 301 140-1] ETSI Intelligent Network Capability
Set 2 (CS2) ETS 301 140-1, available at
http://webapp.etsi.org/key/queryform.asp

[ETSI TS 129 078] Customized Applications for Mobile
network Enhanced Logic (CAMEL) Phase X; CAMEL
Application Part (CAP) specification (3GPP TS 29.078),
available at http://webapp.etsi.org/key/queryform.asp

## Appendix A: Tromboning

It is possible in certain SS7 networks to overcome the
tromboning issue by the use of a network feature known
as "Release Link Trunking." RLT lets trunk resources be
used efficiently after scenarios such as call forwarding or
call transfer have occurred.

In Figure 9, caller "A" dials caller "B" and is then
transferred to caller "C." The figure shows that the two
trunks connecting SSP1 and SSP2 are unnecessary once
"A" is answered by "C." The RLT feature could release
these trunk resources, which would then be free for other
calls. However, the SS7 signaling for this feature is not
standardized and individual implementations will vary
depending upon the SS7 equipment manufacturer
involved.

## Acronyms

| | |
|---|---|
| **ACM** | Acknowledgement Message |
| **ANM** | Answer Message |
| **CAMEL** | Customized Applications for Mobile network Enhanced Logic |
| **CRBT** | Color Ring Back Tone |
| **DFCWA** | Disconnect Forward Connection (with Argument) |
| **ERB** | Event Report BCSM (Basic Call State Machine) Event |
| **ETC** | Establish Temporary Connection |
| **GSMSCF** | GSM Service Control Function |
| **GSMSRF** | GSM Specialized Resource Function |
| **GSMSSF** | GSM Service Switching Function |
| **HLR** | Home Location Register |
| **HMP** | Host Media Processing |
| **IAM** | Initial Address Message |
| **ICA** | Initiate Call Attempt |
| **INAP** | Intelligent Network Application Part |
| **InitialDP** | Initial Detection Point |
| **IP** | Intelligent Peripheral |
| **ISUP** | ISDN User Part |
| **MAP** | Mobile Application Part |
| **MgCS** | Merge Call Segments |
| **MSC** | Mobile Switching Center |
| **PA** | Play Announcement |
| **RRB** | Request Report BCSM (Basic Call State Machine) Event |
| **RLT** | Release Link Trunking |
| **SCP** | Service Control Point |
| **SS7** | Signaling System 7 |
| **SSP** | Service Switching Point |
| **VLR** | Visitor Location Register |

## For More Information

Dialogic® SS7 Protocols: *Call Test Utility (CTU) User Guide* — http://www.dialogic.com/support/helpweb/signaling/software3.htm

Dialogic® SS7 Protocols: *INAP Test Utility (INTU) User Guide* — http://www.dialogic.com/support/helpweb/signaling/software3.htm

Dialogic® SS7 Protocols: *INAP Programmer's Manual* — http://www.dialogic.com/support/helpweb/signaling/software3.htm

Dialogic® Signaling Building Blocks — http://www.dialogic.com/products/signalingip_ss7components/signaling_software.htm

Dialogic® SS7 Products Software and Documentation — http://www.dialogic.com/support/helpweb/signaling/

Dialogic® SS7 and Sigtran Signaling technical discussion forum — http://www.dialogic.com/forums/forums/forum-view.asp?fid=12

Dialogic® hardware and software building blocks for integration into high performance communications solutions — http://www.dialogic.com/products

*Global Call SS7 Technology Guide* — http://www.dialogic.com/support/helpweb/signaling/

**Dialogic**

To learn more, visit our site on the World Wide Web at **http://www.dialogic.com**

Dialogic Corporation
9800 Cavendish Blvd., 5th floor
Montreal, Quebec
CANADA H4M 2V9

**www.dialogic.com**