NXDN®

NXDN Technical Specifications

Part 1:
Air Interface

Sub-part B:
Basic Operation

NXDN TS 1-B Version 1.3

November 2011

NXDN Forum

Contents

Figures

Tables

Disclaimer

The information presented here is intended to be for clarification and/or information purpose only, and care has been taken to keep the contents as neutral and accurate as possible.

The use or practice of contents of the information may involve the use of intellectual property rights (“IPR”), including pending or issued patents, or copyrights, owned by one or more parties. The NXDN Forum makes no search or investigation for IPR, nor the NXDN Forum makes no arrangement of licensing negotiation for IPR between the user and the owner of IPR.

All warranties, express or implied, are disclaimed, including without limitation, any and all warranties concerning the accuracy of the contents, its fitness or appropriateness for a particular purpose or use, its merchantability and its non-infringement of any third party’s IPR.

The NXDN Forum expressly disclaims any and all responsibilities for the accuracy of the contents and makes no representations or warranties regarding the content’s compliance with any applicable statute, rule or regulation.

The NXDN Forum shall not be liable for any and all damages, direct or indirect, arising from or relating to any use of the contents contained herein, including without limitation any and all indirect, special, incidental or consequential damages (including damages for loss of business, loss of profits, litigation, or the like), whether based upon breach of contract, breach of warranty, tort (including negligence), product liability or otherwise, even if advised of the possibility of such damages.

The foregoing negation of damages is a fundamental element of the use of the contents hereof, and these contents would not be published by the NXDN Forum without such limitations.

Document Copyrights

Trademarks

NXDN® is a registered trademark of JVC KENWOOD Corporation and Icom Incorporated.

AMBE+2™ is a trademark of Digital Voice Systems, Inc.

1. Introduction

This document provides the basic procedures for a Subscriber Unit and a Conventional Repeater of an NXDN System to utilize voice and data communication services and supplementary services on a traffic channel.
This document contains the following:

Usage and determination of functional channels for transaction on a traffic channel.
Packetization of data and procedure including a retry for data calls.

This document describes behaviors of a Subscriber Unit and a Conventional Repeater in a conventional system. However, sufficient descriptions that can be applicable to behavior on a traffic channel in a trunked radio system are contained in this document. Hence this document intends to provide basic operations on a traffic channel for all systems. A trunked radio system addressed in this document means Type-C trunked system of which the trunking procedure is described in REF [2].

2. References

Reference documents are listed below. This document and the references are mutually supplemented.

REF [1] Part 1-A Common Air Interface Version 1.3
REF [2] Part 1-C Trunking Procedures Version 1.3
REF [3] Part 1-D Security Version 1.3

3. Abbreviations

To help understand this document, abbreviations are listed below.

AES	Advanced Encryption Standard
BCCH	Broadcast Control Channel
CAC	Common Access Channel
CAI	Common Air Interface
CCCH	Common Control Channel
CR	Conventional Repeater
CRS	Conventional Repeater Site
DES	Data Encryption Standard
DMO	Direct Mode Operation
FACCH1	Fast Associated Control Channel 1
FACCH2	Fast Associated Control Channel 2
FS	Fixed Station
FSW	Frame Sync Word
FDMA	Frequency Division Multiple Access
L1	Layer 1
L2	Layer 2
L3	Layer 3
LICH	Link Information Channel

MS	Mobile Station
PBX	Private Branch Exchange
PSTN	Public Switched Telephone Network
RAN	Radio Access Number
RU	Repeater Unit
RCCH	RF Control Channel
RDCH	RF Direct Traffic Channel
RTCH	RF Traffic Channel
SCPC	Single Channel Per Carrier
SACCH	Slow Associated Control Channel
SU	Subscriber Unit
TC	Trunking Controller
TR	Trunking Repeater
TRS	Trunking Repeater Site
UDCH	User Data Channel
UPCH	User Packet Channel
USC	User Specific Channel
VCH	Voice Channel

4. Functional Channel Setting

This section describes the standard usage of a functional channel in each communication mode.

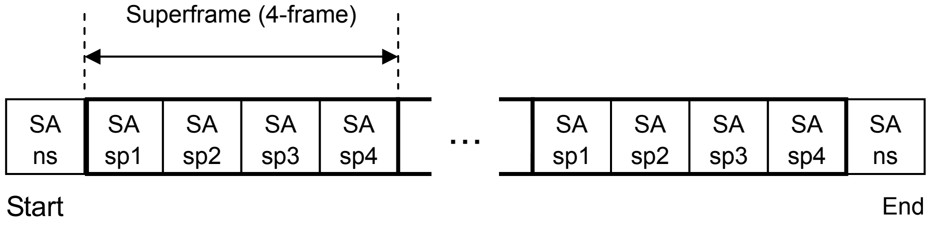
4.1. Voice Call Operation

This section outlines how a sender SU transmits functional channels and how a recipient SU determines received functional channels.

4.1.1. Superframe Structure

While communication modes other than a voice call do not require identifying the superframe structure, a voice call requires identifying two types of structure of a superframe and non-superframe structure.

In a voice call, the 1st frame and the last frame of transmission have the non-superframe structure, and all other frames during the transmission basically have the superframe structure comprised of 4 frames.



SA(ns): SACCH non-superframe

SA(sp): SACCH superframe

Figure 4.1-1 Frame Structure in a Voice Call

SACCH is the only functional channel which has the superframe structure. When a layer 3 message is sent using an FACCH1 by stealing the VCH during a voice call, the FACCH1 does not need to use a superframe structure, and the FACCH1 can be used arbitrarily on single or continuous frames.

During a normal voice call, a VCALL message is embedded in an SACCH, and when the embedded message temporarily switches to other message during a call, it shall switch in the state maintaining the superframe structure.

There are 2 exceptive states for the superframe structure. The first exception shall be applied to the response procedure when making a status call or simultaneous data call during a voice call. Following are two cases when a status call or simultaneous data call during a voice call is received and then a response to the call is sent back:

- In the event that a user presses the PTT switch to begin transmitting a talkback and the response is sent along with the Talkback.
- In the event that only response is sent without a talkback.

In the former case, a voice call is initiated, and the superframe structure shall be applied because a voice call is accompanied.

In the latter case, the exception shall be applied because a voice call is not a accompanied. The frame structure for the latter case is as illustrated in Figure 4.1-2. It does not have the superframe structure and shall have only 2 frames consisting of the 1st frame and a last frame in the non-superframe structure. Except for the second case described below, three or more consecutive non-superframe structures shall be prohibited.

SA ns	SA ns

Start End

SA(ns)：SACCH non-superframe

Figure 4.1-2　Exception 1 of Superframe Structure

The second exception of the superframe structure shall be applied to the repeat operation during the Hold Time. After a SU completes transmitting in a voice call, a CR continues the outbound transmission for a certain period. While this Hold Time elapses, the CR can continuously transmit in a non-superframe structure whose number is not limited.

...	SA ns	SA ns	SA ns	SA ns	SA ns	SA ns

Hold Time Start End

SA(ns)：SACCH non-superframe

Figure 4.1-3　Exception 2 of Superframe Structure

4.1.2. Functional Channel Allocation within a Frame

Since there are multiple types of bit rates and Vocoders available, how VCH and FACCH1 are allocated in a frame shall be determined depending on the combination of these.

The fundamental allocation is allocation which uses EHR at 4800 bps and, in addition to this, the allocation using EFR can also be used at 9600 bps. In the case of EHR, the VCH corresponding to 20 ms of voice data is 72 bits length and 4 VCHs can be allocated into the one frame. In the case of EFR, the VCH is 144 bit length that is twice as long as EHR and 2 VCHs can be allocated into one frame.

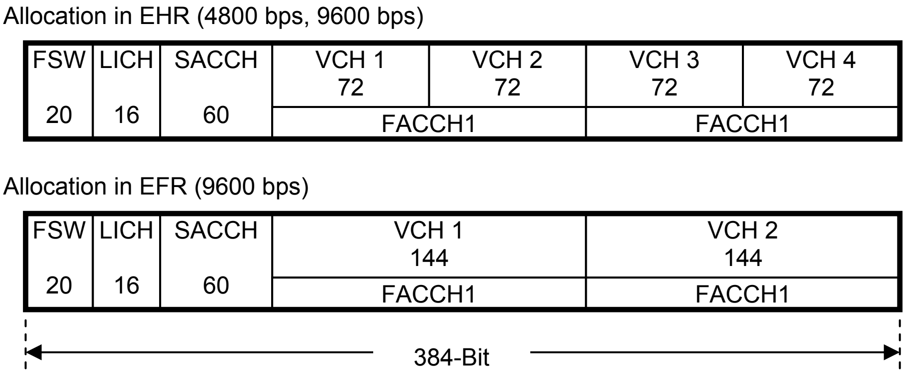


Figure 4.1-4 Allocation Method for 2 Types of VCH

4.1.2.1. Allocation of Functional Channel at 9600 bps/EHR and Intermittent Operation

Since the number of bits for one frame is common in both bit rates, the frame time of 9600 bps is 40 ms which is half that of 4800 bps. In the case EHR is used at 9600 bps, allocating 4 VCHs into one frame enables transfer of an 80 ms voice data in 40 ms frame, so that there is no voice data to be sent in the next frame. Therefore, the basic operation when EHR is used at 9600 bps should alternately send a frame including 4 VCHs and a frame including two FACCH1s. Normally, frames are alternately sent, however VCHs can be stolen by an FACCH1 in the VCH frame as necessary.

The VCH and FACCH1 frames shall be allocated into the superframe in the sequence of VCH, FACCH1, VCH and FACCH1, and the steal flags in LICH shall also be set in accordance with the sequence. While VCH may be replaced with an FACCH1, the FACCH1 frame is fixed. Hence the even-numbered frame in a superframe shall always be the FACCH1 frame.

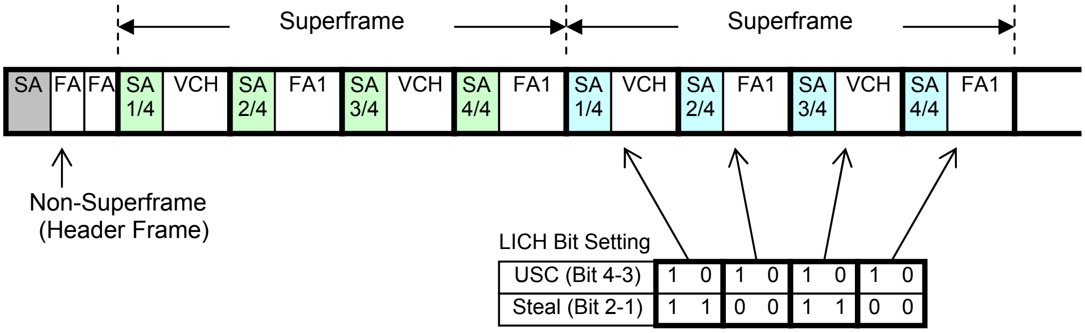
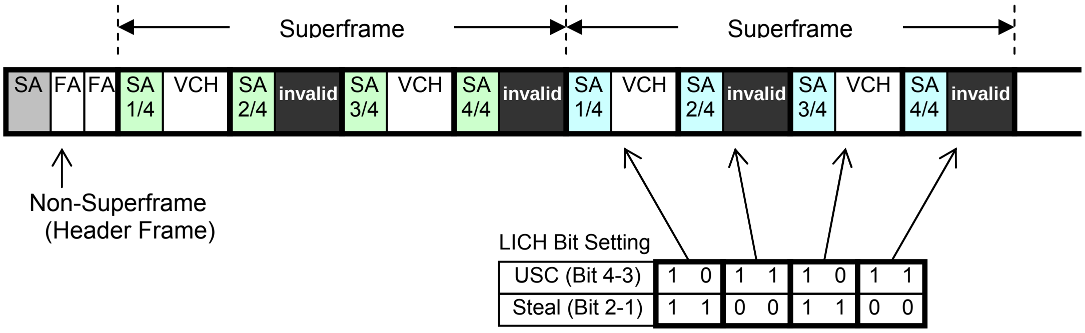


Figure 4.1-5 Allocation in Superframe for 9600 bps/ EHR

In the event that no FACCH1 needs to be sent along with VCH, intermittent operation is enabled by using a frame sending an FACCH1.

Intermittent operation is activated by notifying that 2 FACCH1s have been invalid for a transmitter and, in this case, “SACCH (Superframe)/ Idle” shall be set as the USC Type of LICH. Intermittent operation enables not only a halt operation in a receiver but also a shutdown or cut-down operation of the transmitter’s output power. When LICH indicates “SACCH (Superframe)/ Idle”, the detailed procedure of the intermittent operation in a transmitter and receiver can be arbitrarily defined.

Target	Item	Specifications
Transmitter	Transmit data	Not defined
	Transmit output control	Controlling procedures for the transmit output and ramping are not defined. However, its transmit output control should not affect the last symbol of SACCH and the first symbol of FSW in the next frame.
Receiver	Receive data	Ignore
	Reception power control	Controlling procedure for the power is not defined. However, the reception power control should be controlled so that receipt of the last symbol of SACCH and the first symbol of FSW in the next frame is possible.



USC (Bit 4-3)	1 0	1 1	1 0	1 1
Steal (Bit 2-1)	1 1	0 0	1 1	0 0

Figure 4.1-6 Superframe Structure for Intermittent Operation at 9600 bps/ EHR

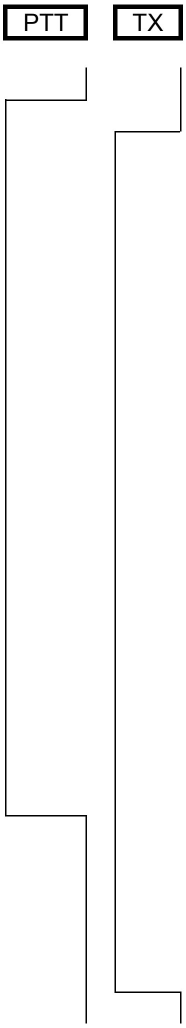
The superframe consists of 1536 bits and contains four 144-bit FACCH1, hence the maximum ratio of intermittent operation is 3/8 calculated from 576 bits/1536 bits.

4.1.3. Typical Transmission Operation of Clear Voice

In a normal voice call, a SU uses two layer 3 messages, VCALL and TX_REL.

Figure 4.1-7 shows a standard usage of functional channels from a start of transmission to the end of transmission when a SU makes a voice call on an RDCH at 4800 bps/EHR. In Figure 4.1-7, functional channels are described in the transmitting sequence in chronological order except FSW.

In the case of 9600 bps/EFR, only the number of VCH is reduced by half, and other operations are the same as shown in Figure 4.1-7. The difference in the case of 9600 bps/EHR, the VCH in even-numbered frames in a superframe are all replaced with FACCH1, and the LICH bit configuration corresponding to those frames is different.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission started
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	VCALL
	FACCH1	VCALL
5	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
6	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (2/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
7	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (3/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
8	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (4/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
9	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
10	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	TX_REL
	FACCH1	TX_REL
11		Transmission completes.

Figure 4.1-7 Voice Call Transmission on an RDCH

1. A user presses the PTT switch on a SU and the SU starts a voice call procedure.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.

4. In the first frame, a single SACCH (non-superframe) and two FACCH1s are used. An IDLE message is sent by the SACCH and a VCALL message shall be sent using at least one FACCH1. Since a single SACCH is used for the layer 3 message, the SR information is set to "00" (not shown in the figure). The receiver can determine with only an FACCH1 whether the VCALL message is suitable.

   In this frame, a single SACCH is used, FACCH1s are used in both first half and last half and the transmission is done on an RDCH. The LICH shall be set corresponding to these state.
5. From a second frame, the VCALL message is sent out using four SACCHs (superframe). This frame transmits the first division of the VCALL message divided into 4 divisions, and is a normal state transmitting the voice data in VCH. The SR information is set to "11" representing the first division in the sequence of the quartered layer 3 message (not shown in the figure).

   In this frame, the SACCH is sent using the superframe format and four VCHs are sent out. The LICH shall be set corresponding to these state.
6. Subsequent to the previous frame, the third frame remains in a normal voice call state and the second division of the divided VCALL shall be sent out. The SR information is set to "10" representing the second division in the sequence of the quartered layer 3 message (not shown in the Figure).
7. The 4th frame is also in a normal voice call state, and the third division of the divided VCALL shall be sent out. The SR information is set to "01" representing the third division in the sequence of the quartered layer 3 message (not shown in the figure).
8. The 5th frame is also a normal voice call state, and the 4th division of the divided VCALL shall be sent out. The SR information is set to "00" representing the fourth division in the sequence of the quartered layer 3 message (not shown in the figure).
9. The 6th frame is also in a normal voice call state. As described in step 5 above, the first division of the divided VCALL shall be sent out.

   In this step, it assumes a user has released the PTT switch.
10. In the 7th frame, a single SACCH (non-superframe) and two FACCH1s are used to indicate the end of transmission. An IDLE message is sent by the SACCH and a TX_REL message representing the end of transmission shall be sent out using at least one FACCH1. Since a single SACCH is used for the layer 3 message, the SR information is set to "00" (not shown in the figure). A receiver begins call termination upon reception of the TX_REL message.

    In this frame, a single SACCH is used and FACCH1s are used in both first half and last half, The LICH shall be set corresponding to these state. The VCALL sent in step 9 has not been sent completely. Even if transmission of the layer 3 message is in midstream, a last frame which contains the TX_REL may be sent out by switching the SACCH structure.

    In this figure, the TX_REL is sent out in the next frame of the frame which has been sent out when the PTT switch has been released. If vocoder data that should be sent are left, the TX_REL may be sent out after completion of the entire vocoder data transmission.
11. The transmission completes after sending out the last frame which contains the TX_REL message.

4.1.4. Typical Transmission Operation of Scrambled Encrypted Voice

In a voice call using the scramble encryption, the configuration for functional channels shall be completely equal to the configuration for functional channels of Clear Voice described in Section 4.1.3.

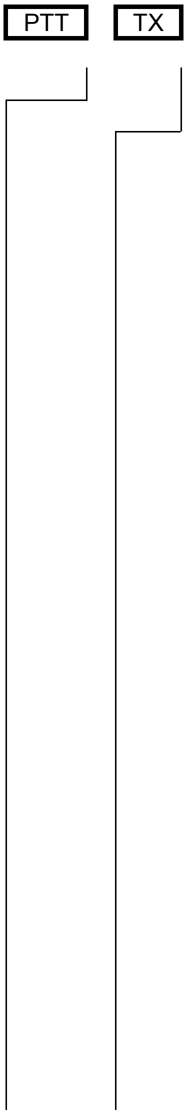
Refer to REF [3] for the detailed method to apply scramble encryption to the voice information.

4.1.5. Typical Transmission Operation of DES/AES Encrypted Voice

In a voice call using the DES and AES encryptions, a SU uses three layer 3 messages, VCALL, VCALL_IV and TX_REL.

Figure 4.1-8 shows a standard usage of functional channels from a start of transmission to the end of transmission when a SU makes a voice call on an RDCH at 4800 bps/EHR. The difference from Clear Voice of Section 4.1.3 is an addition of VCALL_IV. This section mainly explains the difference points, and describes upon simplifying Figure 4.1-8 and the operation explanation.

Refer to REF [3] for the detailed method to apply the DES and AES encryptions to the voice information.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission started
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	VCALL
	FACCH1	VCALL_IV
5	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
6		Frames for VCALL (2/4) and VCALL (3/4) are sent
7	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (4/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
8	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL_IV (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
9		Frames for VCALL_IV (2/4) and VCALL_IV (3/4) are sent
10	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL_IV (4/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
11		After that, superframe for VCALL or for VCALL_IV is sent alternately.

Figure 4.1-8 DES/AES Encrypted Voice Call Transmission on an RDCH

1. A user presses the PTT switch on a SU and the SU starts a voice call procedure.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the first frame, a single SACCH (non-superframe) and two FACCH1s are used. An IDLE message is sent by the SACCH, and a VCALL and VCALL_IV messages are sent by the FACCH1s. From the second frame, the VCALL message is sent out using four SACCHs. This frame transmits the first division of the VCALL message divided into 4 divisions.
6. The second division of the divided VCALL is sent in the third frame, and the third division of the divided VCALL is sent in the 4th frame.
7. The 4th division of the divided VCALL is sent in the 5th frame.
8. From the 6th frame, the VCALL_IV message is sent out using four SACCHs. This frame transmits the first division of the VCALL_IV message divided into 4 divisions.
9. The second division of the divided VCALL_IV is sent in the 7th frame, and the third division of the divided VCALL_IV is sent in the 8th frame.
10. The 4th division of the divided VCALL_IV is sent in the 9th frame.
11. While the PTT switch is pressed, VCALL and VCALL_IV are alternately sent in superframes.

4.1.6. Typical Reception Operation

The processing procedures of functional channels in a receiving SU are described below.

Synchronization for the FSW shall be established, and then the LICH shall be determined. Since the LICH represents an SACCH configuration and VCH stealing state, the SU shall determine the information of the LICH and switch to the decoding processing corresponding to the format which comes in subsequent to the LICH. If a CRC in the SACCH or FACCH1 passes and the Message Type indicates a message for a voice call, the SU shall carry out the voice call processing. Upon receipt of a frame which contains a VCH, the Vocoder shall start a decoding procedure.

In the event that a receiving SU succeeds in receiving the 1st frame correctly, the SU may determine the VCALL contained in the FACCH1 of the 1st frame and unmute the receiving audio if the VCALL satisfies the receiving conditions.

Since the layer 3 message can be received only in the SACCH while a SU is in the Late Entry state due to failure of receiving the 1st frame, SACCH shall be received in all 4 frames in a correct sequence and then be determined.

Since it is uncertain from which frame of quartered SACCHs that a SU can receive depending on timing to receive, the SR information contained in an SACCH shall be determined appropriately.

In the event that a TX_REL message is received in the last frame, a SU mutes the received audio at an appropriate timing. Desyncronization of FSW, failure of CRC in an SACCH may occur, for instance, in the case of low signal strength. The received audio mute except for the TX_REL reception shall be processed in accordance with the desynchronization conditions as defined in REF [1].

4.1.7. Receiver Squelch Conditions

This section describes the determination procedure for outputting a received audio signal in a SU.

The layer 3 message contains some identifiable information, and there are several options for output conditions of received audio signal on a SU. Determination criteria in the modes of reception of a SU and the received information elements are presented in Table 4.1-1.

Modes of Reception	Information Elements				
	RAN	Group ID	Destination Unit ID	Cipher Type	Key ID
RAN Only (Clear)	Valid	n/a	n/a	Valid	n/a
RAN Only (Encrypted)	Valid	n/a	n/a	Valid	Valid
Group Call (Clear)	Valid	Valid	n/a	Valid	n/a
Group Call (Encrypted)	Valid	Valid	n/a	Valid	Valid
Individual Call (Clear)	Valid	n/a	Valid	Valid	n/a
Individual Call (Encrypted)	Valid	n/a	Valid	Valid	Valid

Table 4.1-1 Squelch Condition

In the case that the receive mode is the RAN Only with unencrypted type, the received audio can be output when only the RAN matches. Basically, the received audio should be output when the Cipher Type indicates the unencrypted. However, the received audio may be output arbitrarily even if Cipher Type indicates the encrypted. In such a case, the received audio is unidentifiable. In the case of the RAN Only mode, the layer 3 message transferred in an SACCH can be arbitrarily determined in accordance with other settings in a receiving SU.

In the case that the receive mode is the RAN Only with encrypted type, the received audio can be output when only the RAN matches. Basically, the received audio should be output if the Cipher Type is the same encryption type as that of the receiving SU and the Key ID is the same value as that of the receiving SU. However, in the case that Key ID = 0 in the scramble encryption in Version 1.2 of REF [1], representing that no encryption key is specified, is received, the received audio shall be output by decrypting using a Key ID preconfigured for the receiving SU. However, the received audio may be output arbitrarily even if the Cipher Type is unencrypted. In such a case, the receiving SU does not carry out decryption and shall output an identifiable audio by use of an unencrypted processing.

In the case that the receive mode is an unencrypted Group Call, the received audio can be output when both RAN and Group ID match. The handling of the Cipher Type is the same way as the above RAN Only mode.

In the case that the receive mode is an encrypted Group Call, the received audio can be output when both RAN and Group ID match. The handling of the Cipher Type is the same way as the above RAN Only mode.

In the case that the receive mode is an unencrypted Individual Call, the received audio can be output when both RAN and Destination Unit ID match. The handling of the Cipher Type is the same way as the above RAN Only mode.

In the case that the receive mode is an encrypted Individual Call, the received audio can be output when both RAN and Destination Unit ID match. The handling of the Cipher Type is the same way as the above RAN Only mode.

4.1.7.1. RAN Setting for Carrier Squelch

In the event that a value of RAN for a receiving SU is set to all 0, the SU shall determine that any received RAN is correct RAN. This behavior is the equivalent to the carrier squelch in analog FM operation.

In the event that the receive mode is the RAN Only, the received audio can be output upon receipt of any RAN. At least if the SU establishes the frame synchronization, it may output the received audio. Or the SU may control the audio output by arbitrarily identifying the information of the LICH or contents of the layer 3 message.

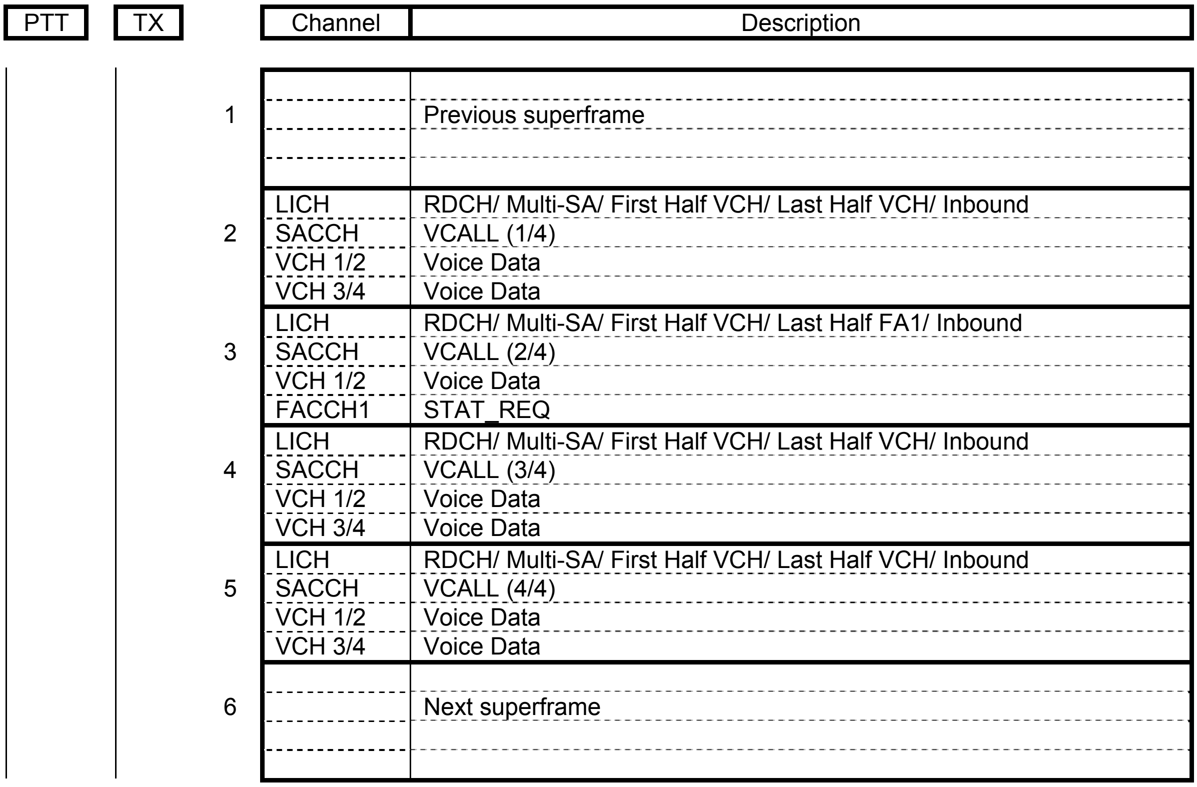
In the event that the receive mode is Group Call or Individual Call, since any RAN is judged to be correct, the received audio can be output when only a Group ID or Destination Unit ID matches.

4.1.8. Voice Call with Status Call

A status call can be carried out simultaneously during a voice call. A status call can also be initiated independently, and the procedure is described in Section 4.3.

4.1.8.1. Transmission Operation for Status Request

Figure 4.1-9 shows an example of usage of the functional channels when a SU makes a status call simultaneously during a voice call on an RDCH. This example is the case that a STAT_REQ message is sent out by replacing a VCH with a FACCH1 during a voice call in 4800 bps/EHR. Particularly in 9600 bps/EHR, dedicated frames for FACCH1 are always allocated, and status-related messages can be sent out without discarding the VCH.



	Channel	Description
1		Previous superframe
2	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
3	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half FA1/ Inbound
	SACCH	VCALL (2/4)
	VCH 1/2	Voice Data
	FACCH1	STAT_REQ
4	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (3/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
5	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (4/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
6		Next superframe

Figure 4.1-9 Voice Call with a Status Request (Example)

1. This is a normal voice call, and frames are being sent out in a sequence using the superframe structure.
2. This is the 1st frame in the arbitrary superframe, and the first division of the quartered VCALL message is sent out on an SACCH. Also, this is a normal state transmitting voice data using a VCH.
3. This is a second frame of the superframe, and subsequent to the previous frame, the second division of the divided VCALL shall be sent out on the SACCH. This is a normal voice call, however a STAT_REQ is sent by replacing the VCH in the last half of this frame with the

FACCH1.

In this frame, the SACCH is sent out in a superframe format, the VCH is used in the first half and the FACCH1 is used in the last half, and the transmission is done on an RDCH. The LICH shall be set corresponding to these state.

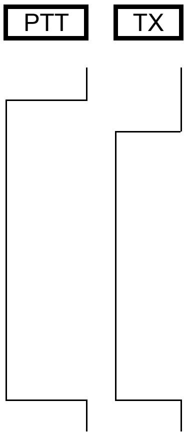
4. This is a third frame of the superframe, and the third division of the divided VCALL shall be sent out on the SACCH. This frame is a normal voice call and sends four VCHs.
5. This is a 4th frame of the superframe, and the 4th division of the divided VCALL shall be sent out on an SACCH.
6. While the PTT switch is pressed, the transmission of the voice call is continued with maintaining the superframe structure.

4.1.8.2. Transmission Operation for Status Response

Figure 4.1-10 shows an example of usage of the functional channels when a SU receives a status call together with a voice call on an RDCH and sends a response to the status call without voice calls,. This behavior is common without depending on the mode of different bit rate.

In case that a response to a status call is sent along with a voice call, the procedure is the same manner as Figure 4.1-9.

As described in Section 4.3, a status call on an RDCH basically uses an FACCH2. This example, however, is a response to a status received during a voice call, and the status response shall be sent on an FACCH1 in this case. Considering the behavior in a trunked radio system, when a status is sent along with a voice call on an RTCH which is assigned for voice calls, an FACCH1 is inevitably used since use of UDCH and FACCH2 are not permitted on an RTCH for voice calls. Hence the same behavior shall be applicable on the RDCH.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission started.
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	VCALL (or STAT_RESP)
	FACCH1	STAT_RESP
5	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	TX_REL
	FACCH1	TX_REL
6		Transmission completes.

Figure 4.1-10 Voice Call with a Status Request (Example)

1. A user does not press the PTT switch because of a non-voice call. Therefore the PTT here is only as a symbolic action of triggering a transmission start.

2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the 1st frame, a single SACCH (non-superframe) and two FACCH1s are used. An IDLE message is sent by the SACCH, a VCALL message (or STAT_RESP) is sent by the FACCH1 of the first half, and an STAT_RESP message is sent by the FACCH1 of the last half. Since a single SACCH is used for the layer 3 message, the SR information is set to "00" (not shown in the figure).

   In this frame, a single SACCH is used, FACCH1s are used in both first half and last half, and the transmission is done on an RDCH. The LICH shall be set corresponding to these state.
5. Since the superframe structure is not applicable to the non-voice call, the next 2nd frame shall be the last frame. A single SACCH and two FACCH1s are used to indicate the end of transmission in this frame. An IDLE message is sent by the SACCH and a TX_REL message representing the end of transmission shall be sent out using at least one FACCH1. A receiver begins call termination upon reception of the TX_REL message.
6. The transmission completes after sending out the last frame which contains the TX_REL message.

In this example, the STAT_RESP is sent out in the 1st frame, but one of two TX_REL messages in the 2nd frame can be STAT_RESP instead. The basic messages for this procedure are 4 types of messages: IDLE, status-related messages, TX_REL and VCALL.

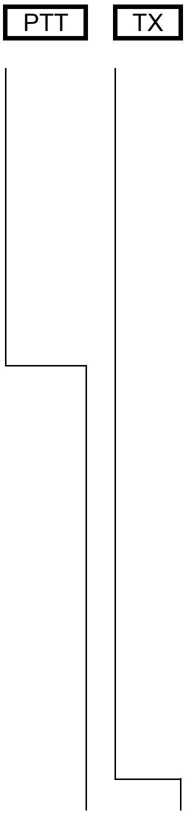
4.1.9. Simultaneous Data Call (Data with Voice)

A slow speed data call can be carried out simultaneously during a voice call. The procedure of simultaneous data call is described in Section 5 as well as data calls.

4.1.9.1. Transmission Operation for Simultaneous Data Request

Figure 4.1-11 shows an example of usage of the functional channels when a SU makes a simultaneous data call during a voice call on an RDCH. This example is the case that 4 SDCALL_REQ messages are sent on FACCH1 when the voice call is completed by releasing the PTT switch in the 4800 bps/EHR. In addition to this example, the message can be sent at any timing during a voice call, and the number of FACCH1 can be selected in accordance with the data size to send.

In the 9600 bps/EHR, the message can use dedicated frames for FACCH1 or use all frames by replacing the VCH with FACCH1.



	Channel	Description
7		Frames preceding step 7 are the same as Figure 4.1-7.
8	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (4/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
9	LICH	RDCH/ Multi-SA/ First Half VCH/ Last Half VCH/ Inbound
	SACCH	VCALL (1/4)
	VCH 1/2	Voice Data
	VCH 3/4	Voice Data
10	LICH	RDCH/ Multi-SA// First Half FA1/ Last Half FA1/ Inbound
	SACCH	VCALL (2/4)
	FACCH1	SDCALL_REQ (Header)
	FACCH1	SDCALL_REQ (User Data)
11	LICH	RDCH/ Multi-SA// First Half FA1/ Last Half FA1/ Inbound
	SACCH	VCALL (3/4)
	FACCH1	SDCALL_REQ (User Data)
	FACCH1	SDCALL_REQ (User Data)
12	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	TX_REL
	FACCH1	TX_REL
13		Transmission completes.

Figure 4.1-11 Simultaneous Data Call in Request (Example)

7. Previous frames are the same as frames from step 1 to step 7 in Figure 4.1-7.
8. This frame is the same frame as step 8 in Figure 4.1-7, and the 4th division of the divided VCALL is sent out on an SACCH. This is a normal voice call state.

9. This frame is the same frame as step 9 in Figure 4.1-7, and the first division of the divided VCALL is sent out on an SACCH.
   In this state, it assumes a user has released the PTT switch.
10. Since this is the case that 4 SDCALL_REQ messages are sent prior to completing the transmission, the next VCALL shall be sent out on an SACCH by maintaining a superframe structure. Also all VCHs are switched to FACCH1, and the SDCALL_REQ message in a Header format is sent in the first half of the FACCH1 and the SDCALL_REQ message in a User Data format is sent in the last half of the FACCH1.
    In this frame, the SACCH is sent out in a superframe structure and FACCH1s are used in both first half and last half. The LICH shall be set corresponding to these state.
11. This frame is similar to the frame in step 10, the next VCALL is sent out in the SACCH, and the remaining SDCALL_REQ message in a User Data format shall be sent out in the first half and last half of FACCH1.
12. Since the PTT switch has been released and the 4 SDCALL_REQ messages have been sent, a single SACCH (non-superframe) and two FACCH1s are used to indicate the end of transmission in the same way as step 10 of Figure 4.1-7. An IDLE message is sent by the SACCH and a TX_REL message representing the end of transmission is sent out using at least one FACCH1. Since a single SACCH is used for the layer 3 message, the SR information is set to "00" (not shown in the figure). A receiver begins call termination upon reception of the TX_REL message.
13. The transmission completes after sending out the last frame which contains the TX_REL message.

4.1.9.2. Transmission Operation in Scramble Encrypted Mode

In a simultaneous data call using the scramble encryption, the configuration for functional channels shall completely equal the configuration as described in Section 4.1.9.1.

Refer to REF [3] for the detailed method to apply the scramble encryption to the user data information.

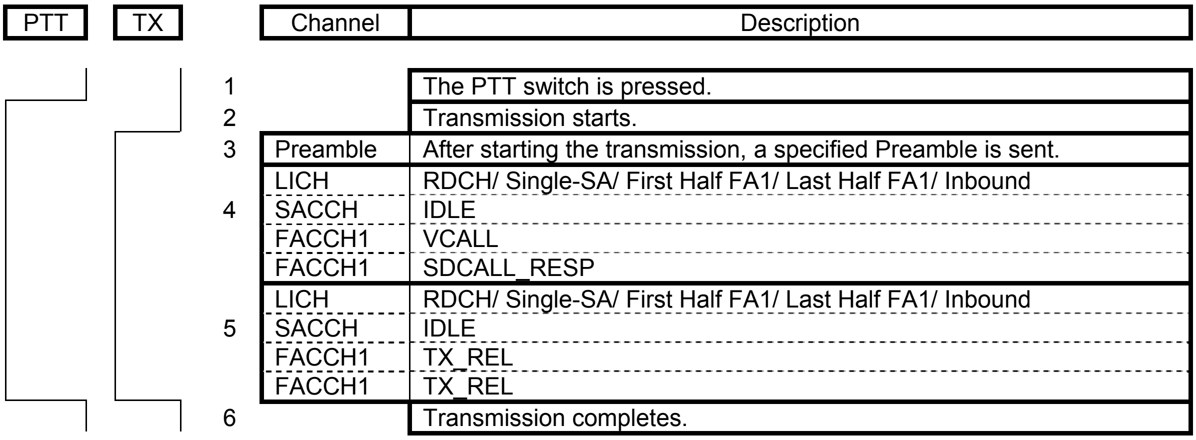
4.1.9.3. Transmission Operation of DES/AES Encrypted Mode

In a simultaneous data call using the DES and AES encryptions, the difference from Section 4.1.9.1 is use of a SDCALL_IV message besides a SDCALL_REQ message. SDCALL_IV is sent after SDCALL_REQ (Header), and the addition of SDCALL_IV needs five FACCH1s in Figure 4.1-11.

Refer to REF [3] for the detailed method to apply the DES and AES encryptions to the user data information.

4.1.9.4. Transmission Operation for Simultaneous Data Response

Figure 4.1-12 shows an example of usage of the functional channels when a SU receives a simultaneous data call together with a voice call on RDCH and responds to the simultaneous data call without a voice call,. This behavior is common without depending on the mode of different bit rate. In the case that a response to a simultaneous data call is sent along with a voice call, the procedure is the same manner as Figure 4.1-11.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission starts.
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	VCALL
	FACCH1	SDCALL_RESP
5	LICH	RDCH/ Single-SA/ First Half FA1/ Last Half FA1/ Inbound
	SACCH	IDLE
	FACCH1	TX_REL
	FACCH1	TX_REL
6		Transmission completes.

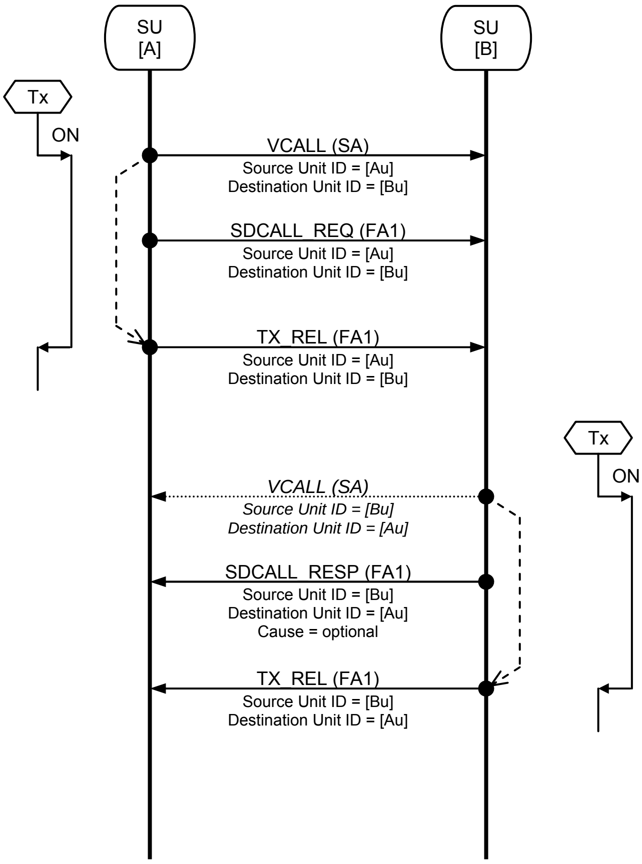
Figure 4.1-12 Simultaneous Data Call in Response (Example)

1. A user does not press the PTT switch because of a non-voice call. Therefore the PTT here is only as a symbolic action of triggering a transmission start.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the 1st frame, a single SACCH (non-superframe) and two FACCH1s are used. The SACCH carries the IDLE message and the FACCH1 of the first half carries VCALL message and the FACCH1 of the last half carries SDCALL_RESP message. Since a single SACCH is used for the layer 3 message, the SR information is set to "00" (not shown in the figure).
   In this frame, a single SACCH is used, FACCH1s are used in both first half and last half and the transmission is done on an RDCH. The LICH shall be set corresponding to these state.Since the superframe structure is not applicable to the non-voice call, the next 2nd frame shall be the last frame. A single SACCH and two FACCH1s are used to indicate the end of transmission in this frame. An IDLE message is sent by the SACCH and a TX_REL message representing the end of transmission shall be sent out using at least one FACCH1. A receiver begins call termination upon reception of the TX_REL message.The transmission completes after sending out the last frame which contains the TX_REL message.

In this example, the SDCALL_RESP is sent out in the 1st frame, but one of the two TX_REL in the 2nd frame can be SDCALL_RESP instead. The basic messages for this procedure are 4 types of messages: IDLE, SDCALL_RESP, TX_REL and VCALL.

4.1.9.5. Sequence Diagram

This figure shows the sequence of the layer 3 message base when a SU [A] (Unit ID = Au) makes a confirmed simultaneous data call to a SU [B] (Unit ID = Bu).



[Notes]

1) The sequence of an unconfirmed Simultaneous Data Call to SU [B] or a Simultaneous Data Call to Group only sends a SDCALL_REQ.
2) One arrow which expresses a SDCALL_REQ sent from an SU [A] is assumed to include the transmission of a SDCALL_REQ (Header) message and SDCALL_REQ (User Data) messages of given number.
   DES/AES encryptions use a SDCALL_IV message too (not shown in the figure).
3) It is optional that SU [B] sends a SDCALL_RESP message with the voice call.
4) Voice Call with Status Call of Section 4.1.8 uses the same sequence as this sequence and the messages are different as follows.
   SDCALL_REQ → STAT_REQ, SDCALL_RESP → STAT_RESP

Figure 4.1-13 Simultaneous Data Call (and Voice w/Status) with confirmed type

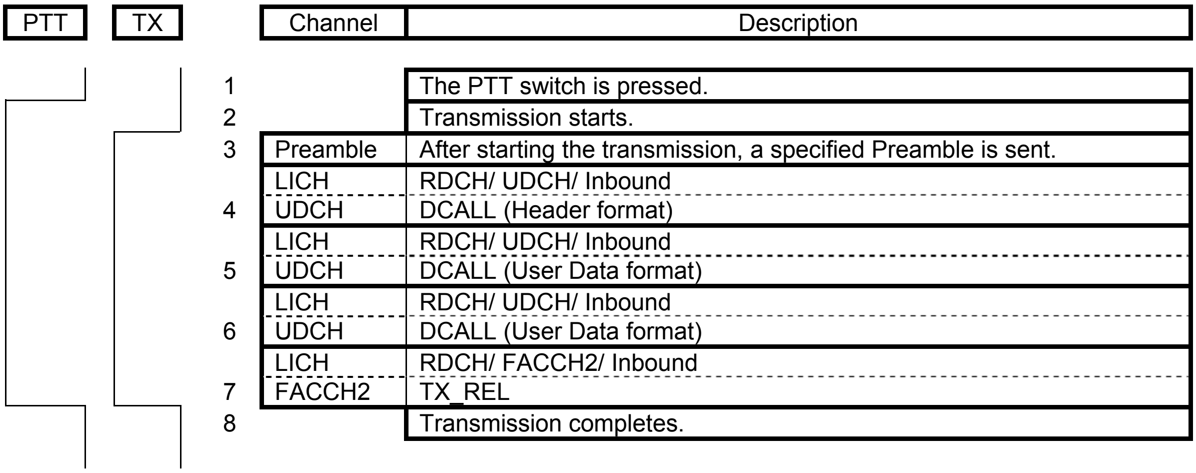
4.2. Data Call Operation

This section outlines how a sender SU transmits functional channels and how a recipient SU determines received functional channels in data calls. Refer to Section 5 for the procedure for a data call including retransmission.

4.2.1. Transmission Operation

In the data call, a SU uses two layer 3 messages of DCALL and DCALL_ACK.

Figure 4.2-1 shows a standard usage of functional channels from a start of transmission to the end of transmission when a SU makes a data call. In Figure 4.2-1, functional channels are described in the transmitting sequence in chronological order except FSW.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission starts.
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH UDCH	RDCH/ UDCH/ Inbound DCALL (Header format)
5	LICH UDCH	RDCH/ UDCH/ Inbound DCALL (User Data format)
6	LICH UDCH	RDCH/ UDCH/ Inbound DCALL (User Data format)
7	LICH FACCH2	RDCH/ FACCH2/ Inbound TX_REL
8		Transmission completes.

Figure 4.2-1 Data Transmission on an RDCH

1. A SU starts the data call procedure by activating the PTT.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the 1st frame, a DCALL message which is a format including the header information (layer 3 information) is sent by a UDCH. Since a UDCH does not use the superframe structure, the SR information for all frames is set to "00 (spare)" (not shown in the figure).

   Since a UDCH is transmitted on an RDCH, the LICH shall be set corresponding to the state.
5. In the 2nd frame, a DCALL message which is a format including only user data is sent out.
6. In the 3rd frame, it is the same as the previous frame.

   This example represents a case when volume of the user data information to be transmitted equals the data length which can be sent in three frames.
7. In the 4th frame, a TX_REL message representing the end of transmission can be arbitrarily sent out using FACCH2. A receiver begins call termination upon reception of the TX_REL message.

   In the case that no TX_REL message is sent out, since the receiver can recognize that the third frame is the last frame upon receipt of the header information in the first frame, the

receiver may begin call termination when the third frame is finished receiving.
Basically, sending a TX_REL message using FACCH2 is recommended.

8. The transmission completes after sending out the last frame.

Figure 4.2-1 shows an example of sending a DCALL message, and the usage of functional channels to send a DCALL_ACK message is the same as that for this example.

4.2.2. Transmission Operation in Scramble Encrypted Mode

In data calls using the scramble encryption, the configuration for functional channels shall completely equal the configuration as described in Section 4.2.1.

Refer to REF [3] for the detailed method to apply scramble encryption to the user data information.

4.2.3. Transmission Operation in DES/AES Encrypted Mode

In data calls using the DES and AES encryptions, the configuration for functional channels shall completely equal the configuration as described in Section 4.2.1.

Refer to REF [3] for the detailed method to apply the DES and AES encryptions to the user data information.

4.2.4. Reception Operation

The processing procedures of functional channels in a receiving SU are described below.

Synchronization for the FSW shall be established, and then the LICH shall be determined. If the information of the LICH represents UDCH, the SU determines that the UDCH will be sent out subsequent to LICH, and then it switches to the decoding processing for the UDCH format. If a CRC in the UDCH passes and the Message Type indicates a message for data calls, the SU shall carry out the data call processing.

Since the layer 3 information is contained only in the 1st frame, the Late Entry is not supported for data calls. Therefore, in order to receive the data call properly, the SU shall receive the 1st frame and determine the header information.

When all frames have been received, if the SU determines that its reception result indicated incorrect state, such as that the CRC in the UDCH did not pass or the fragment number was incorrect, the reception result using a DCALL_ACK message is sent back to a transmitting SU in a confirmed data call.

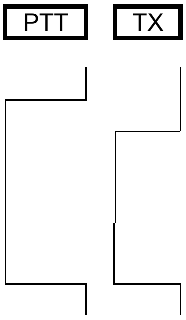
4.3. Status Call Operation

This section outlines how a sender SU transmits functional channels and how a recipient SU determines received functional channels in a status call.

4.3.1. Transmission Operation

In a normal status call, a SU uses four layer 3 messages of STAT_INQ_REQ, STAT_INQ_RESP, STAT_REQ and STAT_RESP.

Figure 4.3-1 shows a standard usage of functional channels from a start of transmission to the end of transmission when a SU makes a status call on a RDCH. In Figure 4.3-1, functional channels are described in the transmitting sequence in chronological order except FSW.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission starts.
3	Preamble	After starting the transmission, a specified Preamble is sent.
	LICH	RDCH/ FACCH2/ Inbound
4	FACCH2	STAT_REQ
	LICH	RDCH/ FACCH2/ Inbound
5	FACCH2	TX_REL
6		Transmission completes.

Figure 4.3-1 Status Notice Call Transmission on an RDCH

1. A SU starts the status call procedure by activating the PTT.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the 1st frame, a STAT_REQ message is sent out by a FACCH2. Since a FACCH2 does not use the superframe structure, the SR information for all frames is set to "00 (spare)" (not shown in the figure).

   Since an FACCH2 is transmitted on an RDCH, the LICH shall be set corresponding to the state.
5. In the 2nd frame, a TX_REL message representing the end of transmission can be arbitrarily sent out using FACCH2. A receiver begins call completion upon reception of the TX_REL message.

   In the case that no TX_REL message is sent out, since a receiver can recognize that only the first frame is valid upon reception of the STAT_REQ in the first frame, the receiver may begin call termination when the first frame is finished receiving.

   Basically, sending a TX_REL message using FACCH2 is recommended.
6. The transmission complete after sending out the last frame.

Figure 4.3-1 shows an example of sending a STAT_REQ message, and the usage of functional channels to send a STAT_RESP, STAT_INQ_REQ or STAT_INQ_RESP messages is the same as that of this example.

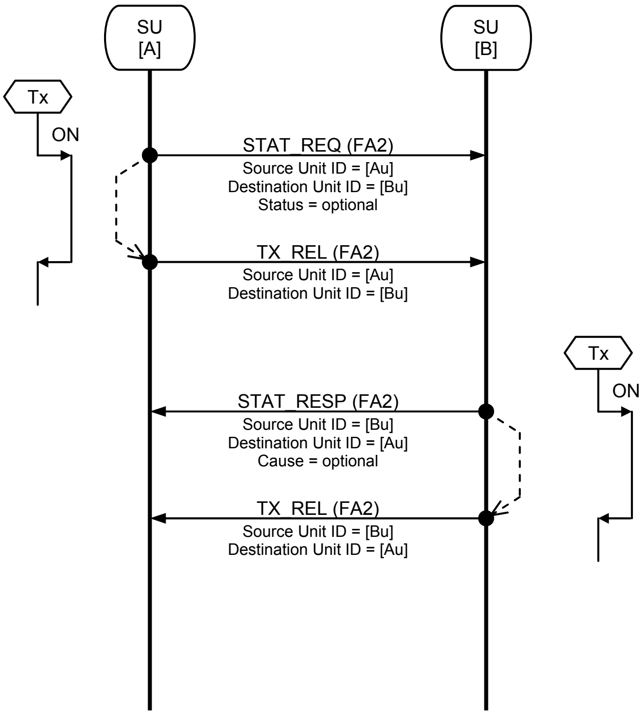
4.3.2. Reception Operation

The processing procedures of functional channels in a receiving SU are described below.

Synchronization for the FSW shall be established, and then the LICH shall be determined. If the information of the LICH represents FACCH2, the SU determines that the FACCH2 will be sent out subsequent to LICH, and then it switches to the decoding processing for the FACCH2 format. If a CRC in the FACCH2 passes and the Message Type indicates a message for status calls, the SU shall carry out the status call processing.

4.3.3. Sequence Diagram

This figure shows the sequence of the layer 3 message base when a SU [A] (Unit ID = Au) makes a confirmed status call to a SU [B] (Unit ID = Bu).



[Notes]

1) The sequence of an unconfirmed Status Call to SU [B] or a Status Call to Group only sends a STAT_REQ.
2) Status Inquiry uses the same sequence as this sequence and the messages are different as follows.
   STAT_REQ → STAT_INQ_REQ, STAT_RESP → STAT_INQ_RESP

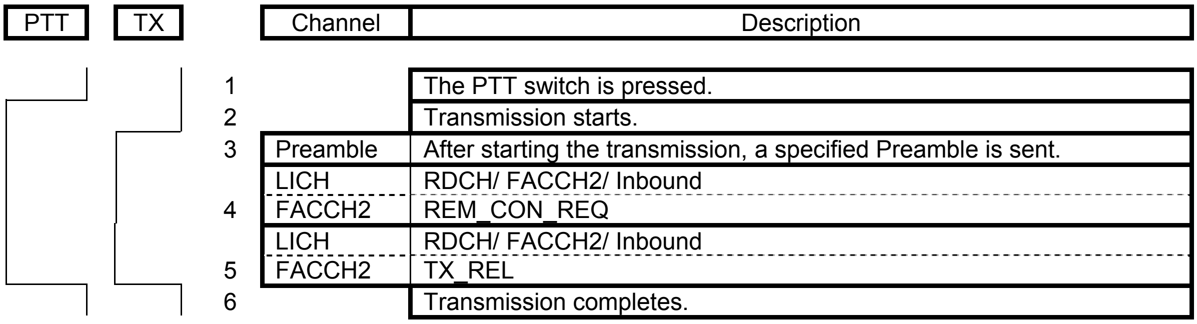
Figure 4.3-2 Status Call with confirmed type

4.4. Remote Control Command

This section outlines how a sender SU transmits functional channels and how a recipient SU determines received functional channels in a remote control command.

4.4.1. Transmission Operation

In a remote control command, a SU uses two layer 3 messages of REM_CON_REQ and REM_CON_RESP. Figure 4.4-1 shows a standard usage of functional channels from a start of transmission to the end of transmission when a SU makes a remote control command on a RDCH. In Figure 4.4-1, functional channels are described in the transmitting sequence in chronological order except FSW.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission starts.
3	Preamble	After starting the transmission, a specified Preamble is sent.
4	LICH	RDCH/ FACCH2/ Inbound
	FACCH2	REM_CON_REQ
5	LICH	RDCH/ FACCH2/ Inbound
	FACCH2	TX_REL
6		Transmission completes.

Figure 4.4-1 Remote Control Command Transmission on an RDCH

1. A SU starts the remote control command procedure by activating the PTT.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. In the 1st frame, a REM_CON_REQ message is sent out by a FACCH2. Since a FACCH2 does not use the superframe structure, the SR information for all frames is set to "00 (spare)" (not shown in the figure).
   Since an FACCH2 is transmitted on an RDCH, the LICH shall be set corresponding to the state.
5. In the 2nd frame, a TX_REL message representing the end of transmission can be arbitrarily sent out using FACCH2. A receiver begins call completion upon reception of the TX_REL message.
   In the case that no TX_REL message is sent out, since a receiver can recognize that only the first frame is valid upon reception of the REM_CON_REQ in the first frame, the receiver may begin call termination when the first frame is finished receiving.
   Basically, sending a TX_REL message using FACCH2 is recommended.
6. The transmission complete after sending out the last frame.

Figure 4.4-1 shows an example of sending a REM_CON_REQ message, and the usage of functional channels to send a REM_CON_RESP message is the same as that of this example.

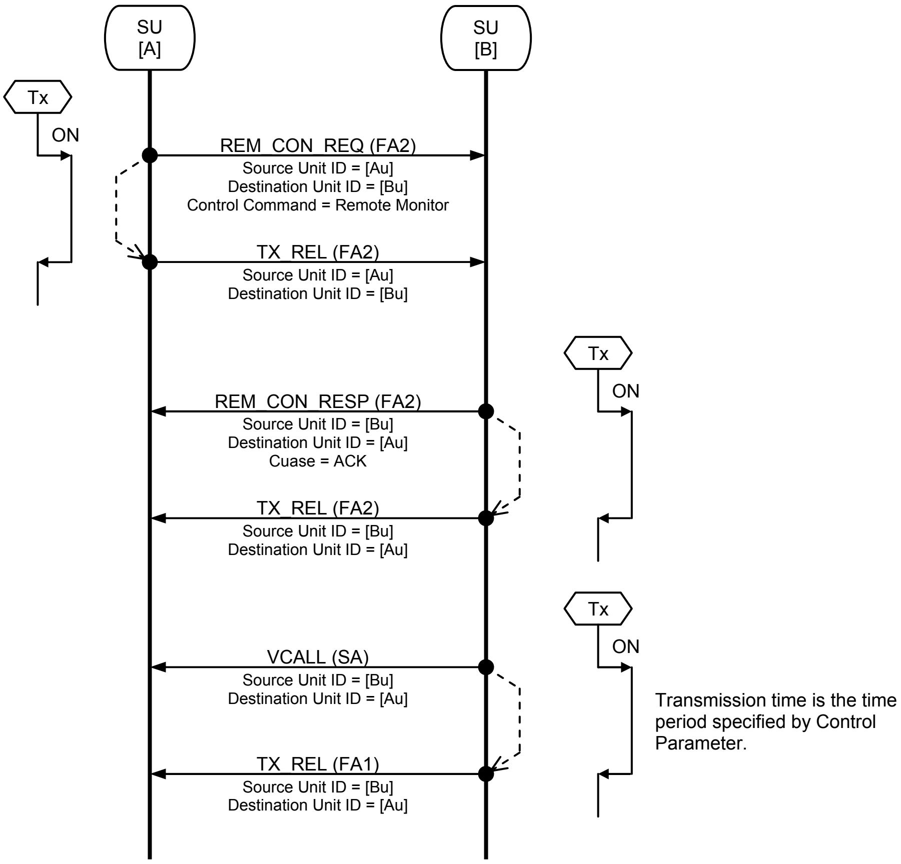
4.4.2. Reception Operation

The processing procedures of functional channels in a receiving SU are described below.

Synchronization for the FSW shall be established, and then the LICH shall be determined. If the information of the LICH represents FACCH2, the SU determines that the FACCH2 will be sent out subsequent to LICH, and then it switches to the decoding processing for the FACCH2 format. If a CRC in the FACCH2 passes and the Message Type indicates a message for remote control commands, the SU shall carry out the remote control command processing.

4.4.3. Sequence Diagram

This figure shows the sequence of the layer 3 message base when a SU [A] (Unit ID = Au) makes a remote control command (Remote Monitor) to a SU [B] (Unit ID = Bu).


SU [A]
SU [B]
Tx
ON
REM_CON_REQ (FA2)
Source Unit ID = [Au]
Destination Unit ID = [Bu]
Control Command = Remote Monitor
TX_REL (FA2)
Source Unit ID = [Au]
Destination Unit ID = [Bu]
Tx
ON
REM_CON_RESP (FA2)
Source Unit ID = [Bu]
Destination Unit ID = [Au]
Cuase = ACK
TX_REL (FA2)
Source Unit ID = [Bu]
Destination Unit ID = [Au]
Tx
ON
VCALL (SA)
Source Unit ID = [Bu]
Destination Unit ID = [Au]
Transmission time is the time period specified by Control Parameter.
TX_REL (FA1)
Source Unit ID = [Bu]
Destination Unit ID = [Au]

[Notes]

1) In the case of Control Command = Stun/Revival/Kill, SU [B] sends a REM_CON_RESP and changes to the specified state.
   The sequence of an unconfirmed Remote Control Command to SU [B] or a Remote Control Command to Group only sends a REM_CON_REQ.

Figure 4.4-2 Remote Control Command

5. Packet Data Transmission

This section describes the packetization method of user data information, the synchronization method of sequence and control method of retransmission when the user data information is transmitted using the data call procedure.

5.1. Type of Data Call

There are 3 types of data call available for NXDN, namely data call, short data call and simultaneous data call.

Short Data Call

A short data call is data calls proceeding on a control channel in a trunked radio system.
In this service, user data limited to approximately 100 bytes can be transmitted.
In this service, a UPCH is used as a functional channel, and a SDCALL_REQ and SDCALL_RESP messages are used as a layer 3 message. Additionally, SDCALL_IV message is used in DES and AES encryptions.

Simultaneous Data Call

A simultaneous data call is a low-speed data calls that proceeds on a traffic channel along with a voice call. In this service, since a voice call proceeds simultaneously, user data limited to approximately 100 bytes can be transmitted in the same manner as a short data call.
In this service, a FACCH1 is used as a functional channel and a SDCALL_REQ and SDCALL_RESP messages are used as a layer 3 message in the same manner as a short data call. Additionally, SDCALL_IV message is used in DES and AES encryptions.

Data Call

A data call is high-speed data calls proceeding on a traffic channel and it can send unlimited user data.
In this service, a UDCH is used as a functional channel, and a DCALL and DCALL_ACK messages are used as a layer 3 message.

5.2. Concept of Data Partitioning

Figure 5.2-1 shows the concept of the user data partitioning in order to actually transmit a use data on the functional channel basis.

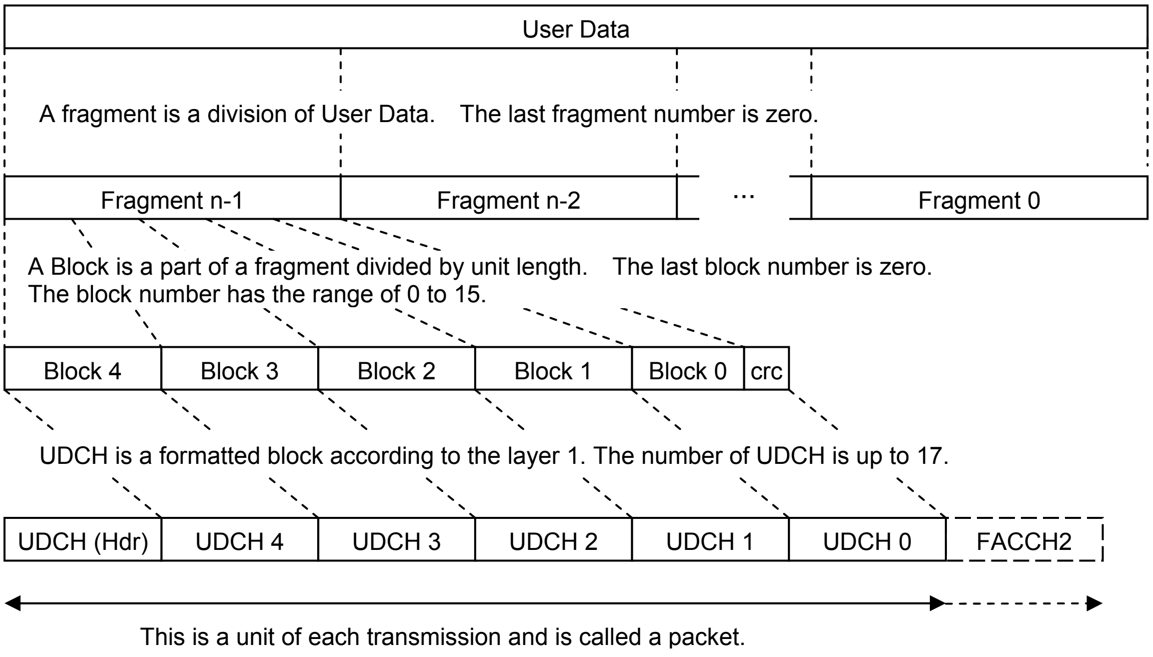

User Data
A fragment is a division of User Data. The last fragment number is zero.
Fragment n-1
Fragment n-2
…
Fragment 0
A Block is a part of a fragment divided by unit length. The last block number is zero.
The block number has the range of 0 to 15.
Block 4
Block 3
Block 2
Block 1
Block 0
crc
UDCH is a formatted block according to the layer 1. The number of UDCH is up to 17.
UDCH (Hdr)
UDCH 4
UDCH 3
UDCH 2
UDCH 1
UDCH 0
FACCH2
This is a unit of each transmission and is called a packet.

Figure 5.2-1 Concept of User Data Partitioning

Figure 5.2-1 shows the partitioning method of user data for a case when one fragment is sent out using five UDCHs as a functional channel. However, in UPCH and FACCH1, only one fragment can be used.

A fragment is data where a user data sent is partitioned, and a block is data where a fragment is partitioned in units of functional channel. Since a block is data where a user data is partitioned in units of a functional channel, the block has the same meaning as the functional channel such as UDCH. Also, one unit of transmission is called a packet, and since user data sent by one unit of transmission is in units of fragment, fragment and packet have the same meaning.

Since the available range of the Block Count field contained in the header is up to 16, the maximum number of blocks that are the partitioned fragment is 16. In the 1st frame of transmission, a DCALL (Header format) message containing a header information is used, and this message does not contain user data. Except for the 1st frame, a DCALL (User Data format) message containing mainly user data elements is used, and this message can contain 20 bytes of user data. The maximum fragment size is 316 bytes because of containing 4-byte Message CRC in the last block, and all user data will be transmitted by using multiple fragments. Transmission of one packet completes when a TX_REL message is sent using FACCH2 as the last frame of transmission.

In a short data call using UPCH, the maximum size of user data that can be sent is basically specified as approximately 100 bytes considering the traffic volume on a control channel. This limit, however, can arbitrarily be changed for each system. For an inbound signal from a SU, the 1st frame of transmission uses an SDCALL_REQ (Header format) message containing a header information, and this message does not contain user data. Except for the 1st frame, a SDCALL_REQ (User Data format) message containing mainly the user data elements is used, and this message can contain 14 bytes of user data. The maximum fragment size is 220 bytes because of containing 4-byte Message CRC in the last block, and a user data can be transmitted by using only one fragment. Since a maximum of 100 bytes user data is defined as the basic specifications, it is recommended that the maximum number of UPCHs is limited to nine (108 bytes of user data).

Similarly, in simultaneous data call using FACCH1, a SDCALL_REQ (Header format) message containing a header information is used in the 1st FACCH1, and this message does not contain user data. Except for the 1st FACCH1, an SDCALL_REQ (User Data format) message is used, and this message can contain 8 bytes of user data. The maximum fragment size is 124 bytes because of containing 4-byte Message CRC in the last block, and a user data can be transmitted by using only one fragment.

5.3. Synchronization of Fragments

In order to divide the user data into several fragments, the information elements of TX Fragment Count, Start Fragment Flag and Circular Fragment Flag shall be used. By using these 3 information elements, the both service of the limited data length format and the format allowing unlimited data transmission are provided.

In the case that a user data consists of multiple fragments, the Start Fragment Flag field in the first packet shall be set to "1". In other packets, it shall be set "0". The TX Fragment Count field represents the sequence of the fragments, and if the Circular Fragment Flag field is set to "0", the TX Fragment Count field is not circulated and the maximum fragment count will be 512. Also if the Circular Fragment Count field is "1", the TX Fragment Count field is circulated and there will be no limit to the number of fragments.

In the case that the data length is pre-determined and that the number of fragments is 512 or less, the Circular Fragment Flag field shall be set to "0" since there is no need to circulate the TX Fragment Count field. If the data length is indefinite, the Circular Fragment Flag field shall be set to "1" to represent a possibility of a circulation counter, since the number of fragments is not predictable.

In the case that the Circular Fragment Flag field is set to "0", the TX Fragment Count field in the first packet is set to (Number of fragments -1), then the counter is decremented in the subsequent packets, and the TX Fragment Count field in the final packet is set to zero.

In the case that the Circular Fragment Flag field is set to "1", the TX Fragment Count field is set to 511 in the 1st packet, and then the counter is decremented in the subsequent packets, and if the TX Fragment Count field reaches a value of 0 in the 512th packet, the TX Fragment Count field shall be reset to 511 in the next packet.

Table 5.3-1 shows configuration values for the information elements in 3 conditions: transmitting a single fragment, transmitting 5 fragments and transmitting unlimited fragments.

Conditions	Start Fragment	Circular Fragment	TX Fragment Count
1. In the case of only one packet	1	0	0
2. First packet in the case of 5 packets	1	0	4
A packet in the mid-stream	0	0	n (n = 3 to 1)
Last packet	0	0	0
3. First packet in the case of indefinite number of packets	1	1	511
A packet in mid-stream	0	1	n (n = 511 to 0)

Table 5.3-1 Usage of Information Elements for Fragments

Since a short data call using UPCH and a simultaneous data call using FACCH1 always comprise only one fragment, they shall use a configured value in the above condition-1.

5.3.1. Processing of Confirmed Type Packet

A sender embeds the TX Fragment Count (hereinafter referred to as N(S)), which represents the transmitted packet number, into a transmission packet and sends the packet out. A receiver embeds the RX Fragment Count (hereinafter referred to as N(R)), which represents the last received packet number, into a response packet and sends the packet out.

Both SU and a CR have the variables V(S) and V(R) as internal variables to maintain the synchronization of packets.

Operation in the case that the data length is pre-determined and that the number of packets to be sent has been determined, which is to say the Circular Fragment Flag is set to "0", shall be as follows. At the beginning, a sender stores the value of (Number of packets -1) to V(S), sets this V(S) to N(S), and then sends the first packet. When the next packet is ready to be sent, V(S) is decremented, that V(S) is set to N(S), and then the next packet will be sent out. In the last packet, it will be V(S) = N(S) = 0.

A receiver stores the received N(S) in the V(R), sets the V(R) to N(R), and then sends back a response packet.

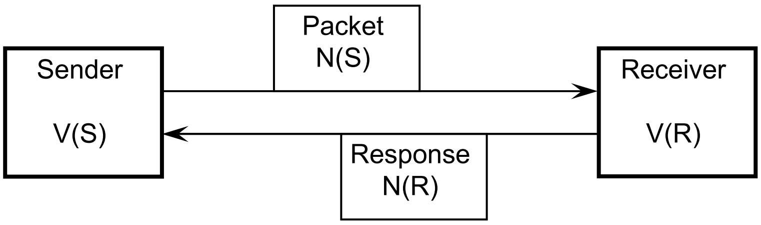


Figure 5.3-1 Conceptual Diagram of Packet Synchronization

V(S) value is equal to N(S) at the time a sender transmits the latest packet, and V(S) will be decremented when a new packet is ready to be sent. In the event that the packet needs to be retransmitted since no response packet from a receiver is received, the same V(S) shall be used without being decremented.

5.3.2. Retry Indication for Packet

Selective Retry Flag shall be used in order to indicate whether the packet has a full data or a part of data for a selective retry. The flag setting in accordance with the transmission conditions is listed in Table 5.3-2.

Since a short data call using UPCH and a simultaneous data call using FACCH1 do not support a selective retry process, they shall use the Flag = 0.

Conditions	Selective Retry Flag
Normal Packet	0
Full Retry Packet	0
Selective Retry Packet	1

Table 5.3-2 Selective Retry Flag Settings

5.3.3. Detection of Duplicated Packet

Since the N(S) is always decremented as each new packet is sent out, a receiver can detect any duplicated packet by comparing the N(S) in the received packet with the stored V(R). The latest N(S) can be retained by storing the N(S) in the received packet in V(R) when an ACK response packet is sent back as a response to the received packet.

Upon receipt of a packet, a receiver compares the N(S) in a received packet to its own variable V(R), and then has one of the following 3 states.

(1) N(S) = V(R) - 1

This represents that the packet was received in a correct sequence. A receiver arbitrarily stores one or more CRC values in UDCHs, decrements the V(R), and then sends back the ACK response packet including the N(R) equal to the N(S).

(2) N(S) = V(R)

This represents that the same packet was redundantly received. This may occur when, for example, a packet was received successfully, and a receiver has sent back an ACK response packet, however a sender fails in receiving the response packet.

To detect the duplicate packet, the Message CRC value in the received packet shall be compared with Message CRC value in the last received packet. In the case that these CRC values are same, since it proves that the same packet has been received, a receiver sends back the ACK response packet and discards the duplicate packet. In the case of different CRC values, since this represents that a new session has begun and the N(S) has been coincidentally same, the receiver checks the header information other than N(S) and carries out an appropriate procedure in accordance with the information.

(3) N(S) = anything else

This represents that a new session has begun. For example, this may occur in the case when a receiver has succeeded to receive a packet and sent an ACK response packet, however a sender has terminated the call on the way since the sender failed in receiving the ACK response packet and expired the maximum number of retransmission. The receiver checks the header information other than N(S) and carries out an appropriate procedure in accordance with the information.

Table 5.3-2 shows the state transition of Start Fragment Flag and N(S) depending on the transmitting state of packets.

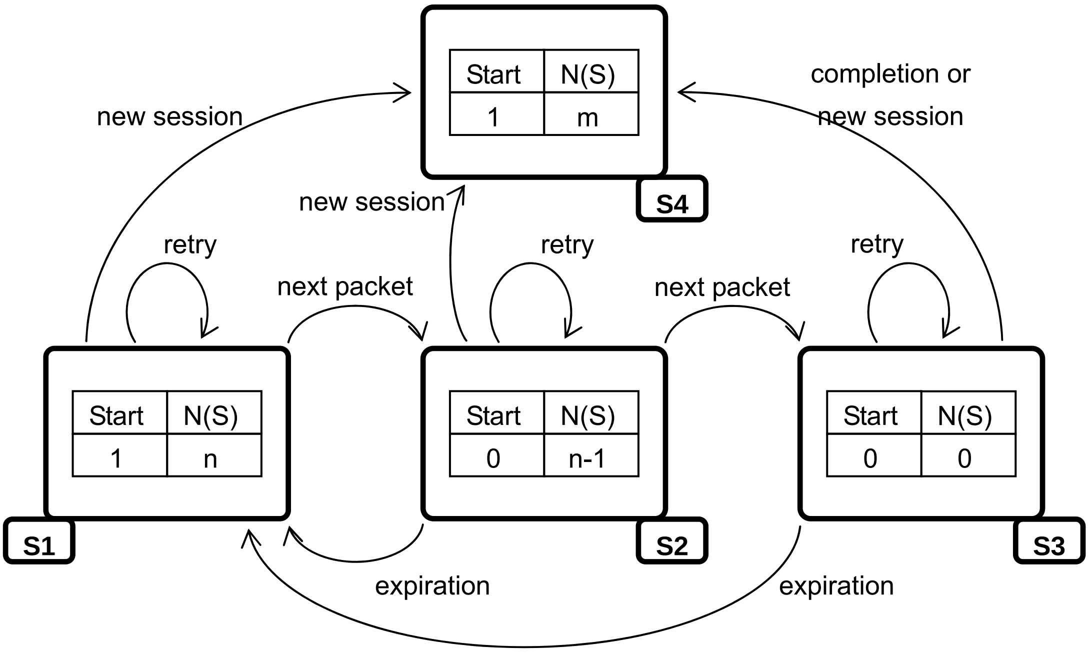


Figure 5.3-2 State Transition of Transmission Packet

In the figure, S1 to S4 represent the following states. Letters n and m are arbitrary numbers.

S1: State that the 1st packet is sent to start data calls.
S2: State the intermediate packets are sent.
S1: State the last packet is sent.
S4: State that the 1st packet is sent to start a new data calls which is different from S1.

A sender starts data calls from the S1 state, and migrates to the next state, S2, upon receipt of the ACK response packet. If no response packet is received, the sender remains at S1 state and retries. If the number of retries is exhausted at S1 state, the sender can either repeat the S1 state again, or migrate to S4 to start a new session. If no response packet is received in the S2 or S3 state and the number of retries is exhausted, the sender can either return to the first state S1 or migrate to S4 to start a new session. When an ACK response packet is received at S3 state, the data call session is completed.

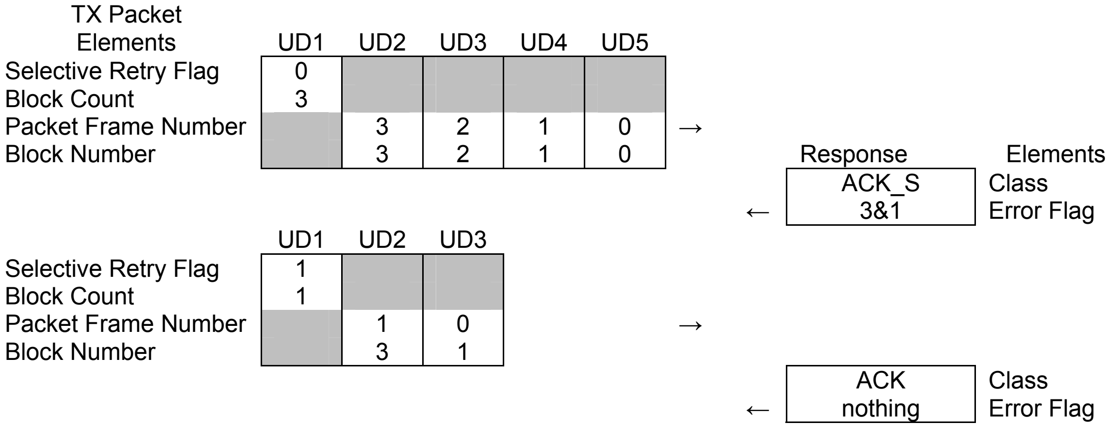
5.4. Synchronization of Block

A sender and receiver can confirm the number of blocks and the sequence of block in a packet by using the Block Count field, Block Number information element and Packet Frame Number information element.

The receiver determines the number of blocks contained in the packet from the Block Count in the header of the transmitted packet, and determines whether it receives blocks in the proper

sequence by checking the Block Number which represents the sequence of each block which constructs a fragment.
In a selective retry, since the receiver indicates a request for a selective retry using a response packet and indicates Error Block Flags corresponding to a block number that cannot be received properly, the sender can determine blocks of which a selective retry is needed. The sender, which received the response packet, sets the Selective Retry Flag in the header of the retransmitted packet to "1", and calculates the Block Count from the number of retransmitted blocks. Also, in each block other than the first block, a block number to be retransmitted is set to Block Number and a frame number in a retry packet is set to Packet Frame Number. The receiver determines by a Selective Retry Flag that a sender has been doing a Selective Retry, confirms that the Block Count matches the number of Error Block Flags specified by a response packet, and determines that the Block Number matches the number specified by Error Block Flag, so that the number of blocks and the sequence of blocks between sender and receiver can be synchronized.

Figure 5.4-1 shows the configuration for information elements in the case that a packet consisting of 5 blocks is sent and a selective retry for the second (UD2) and 4th (UD4) blocks is carried out. Description of FACCH2 is omitted here.



TX Packet Elements	UD1	UD2	UD3	UD4	UD5
Selective Retry Flag	0				
Block Count	3				
Packet Frame Number		3	2	1	0
Block Number		3	2	1	0

→

Response	Elements
ACK_S	Class
3&1	Error Flag

←

	UD1	UD2	UD3
Selective Retry Flag	1		
Block Count	1		
Packet Frame Number		1	0
Block Number		3	1

→

ACK	Class
nothing	Error Flag

←

Figure 5.4-1 Information Element Setting in Selective Retry

UDCH1 uses the Header format containing the header, and all others use the User Data format containing the user data. In the first packet transmission, the Selective Retry Flag in the header is initialized and what the Block Count indicates that the Packet Frame Number begins with No. 3 represents containing four User Data formats. In the case that it is assumed the CRCs of UDCH2 and UDCH4 result in error upon receipt of a packet, in order to request a selective retry, a receiver sends back a response packet to which the Class field is set to ACK_S and the bit in the Error Block Flag corresponding to the block number where an error has occurred is set. The sender determines which blocks shall be retransmitted by checking the Error Block Flag in the response packet. The sender recalculates the value of Block Count from the number of blocks to be retransmitted, set the Selective Retry Flag, and then re-constructs the header. By constructing a retry packet after the Block Number and the Packet Frame Number of UDCH2 and UDCH3 are respectively set to a retransmitted block number and a new frame number used for retransmission, the sender transmits the selective retry packet. When the receiver confirms that the value of Block Number matches the number specified by Error Block Flag and sends back a response packet, the selective retry procedure completes.

5.5. Procedure of Confirmed Delivery Packet

5.5.1. Transmission Procedure

This section describes the transmission procedure to send one packet in confirmed delivery. This procedure completes when the number of retry exceeds its maximum number or the ACK response packet has been received.

In this procedure, the following parameters are used:

Response Packet Wait Time:	T_resp
Number of Retry:	N_retry
Maximum Number of Retry:	N_retry_max

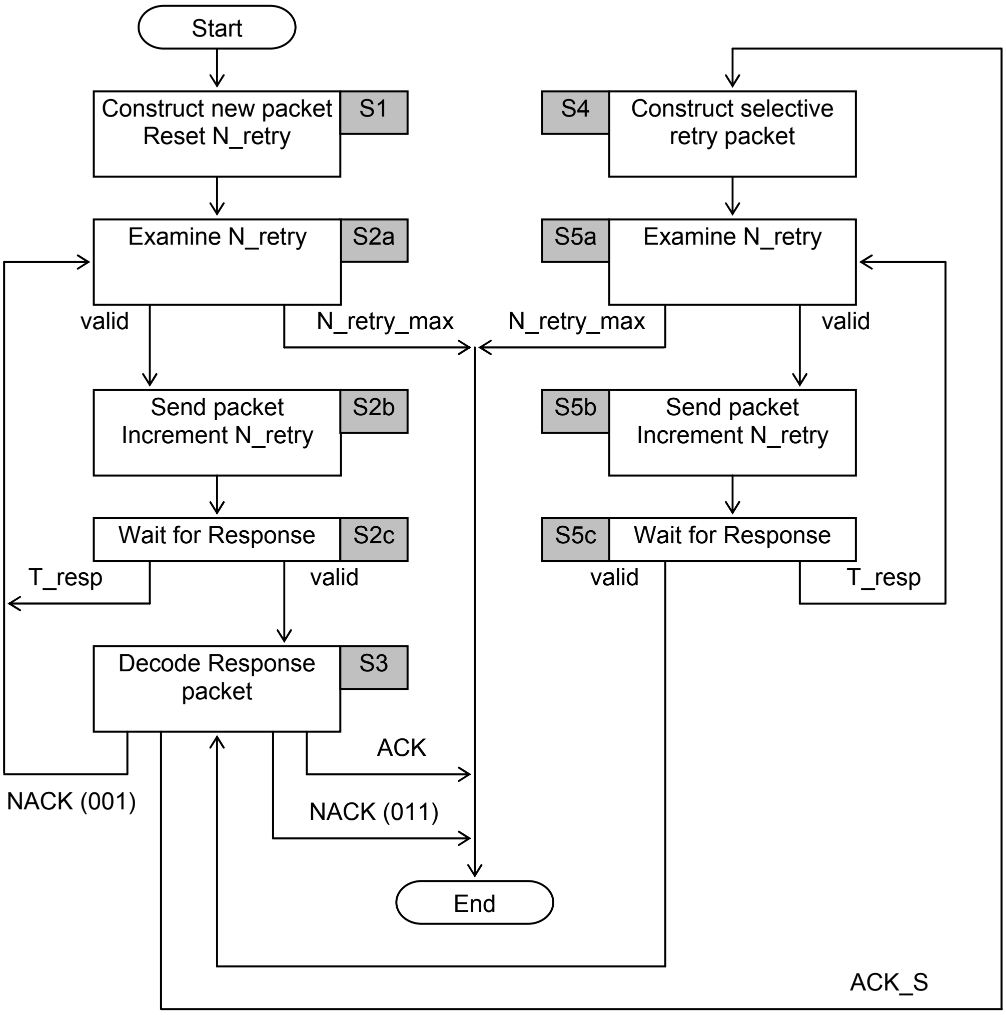


Figure 5.5-1 State Diagram of Transmission in Confirmed Delivery Packet

Step 1. Packet Construction

Step 1.1 N(S) Settings

Start Fragment Flag is set depending on the first fragment. If it is the first one, V(S) is initialized to the number of fragments and the value is set to N(S). If it is not the first one, V(S) is decremented and the value is set to N(S).

Step 1.2 Appending Message CRC

Calculate a Message CRC from the fragment and append it to the last of the fragment. If the fragment that the Message CRC was appended to is short, Null Octets shall be properly appended after the Message CRC to adjust to the specified length.

Step 1.3 Calculation of Blocks

Block Number field in each block is allocated by calculating the number of blocks after dividing a fragment into blocks, and by setting the last block to zero.

Step 1.4 Formatting

Header Format

Selective Retry Flag is set to "1".

Block Count field is set to a value of (a block number obtained in Step 1.3 minus 1).

Pad Octet Count field is set using the length of Null Octet appended in Step 1.2.

User Data Format

Packet Frame Number and Block Number information elements in each block are allocated.

In accordance with the layer 3, the Header and User Data formats are formatted. Then in accordance with the layer 1, the CRC data is appended and the FEC coding is processed.

Step 1.5 Initialization

N_retry is initialized to zero.

Step 2. First Packet Transmission

Step 2.1 Transmission

If N_retry is equal to N_retry_max, the procedure completes. Otherwise the packet is sent.

N_retry is incremented.

Step 2.2 Wait for Response

Wait for a response packet from a receiver. The response packet contains values to represent the state such as Class, Type and RX Fragment Count. Standard values are shown in Table 5.5-1.

Class	Type	RX Fragment Count	Descriptions
00	001	N(R)	ACK: Reception success
01	001	N(R)	ACK_S: Selective retry request
11	001	N(R)	NACK: CRC check failure
11	011	N(R)	NACK: Abort

Table 5.5-1 Parameters for Response Packet

If no response packet is received and T_resp period elapses, return to Step 2.1. If a valid response packet is received, proceed to Step 3.

Step 3. Determination of Response Packet

The response packet is processed in accordance with Table 5.5-2.

Contents of Received Response Packet	Next Step
Packet reception succeeded (in the case of single packet).	Complete the procedure.
Packet reception succeeded (in the case of multiple packets).	Repeat the same procedure to send the next packet. If the last packet, complete the procedure.
CRC check failed.	Return to Step 2.1.
Abort	Complete the procedure.
Selective retry required	Proceed to Step 4.

Table 5.5-2 Determination of Response Packet

Step 4. Construction of a Selective Retry Packet

Step 4.1 Determination of Block which requires retransmission

Read the Error Block Flag from the response packet and determine the Block Number to be retransmitted.

Step 4.2 Formatting

Header Format

The Selective Retry Flag is set to "1".

The Block Count is recalculated from the number of blocks to be retransmitted.

The Pad Octet Count is set to same as that of Step 1.4 regardless of whether a Message CRC is included the block to be retransmitted.

User Data Format

New Packet Frame Number in each block is allocated.

The Header and User Data formats are formatted in accordance with the layer 3, and then the CRC data is appended and the FEC coding is processed in accordance with the layer 1.

Step 5. Transmission of Selective Retry Packet

Step 5.1 Transmission

If N_retry is equal to N_retry_max, the procedure completes. Otherwise the packet is sent.

N_retry is incremented.

Step 5.2 Wait for Response

Wait for a valid response shown in Table 5.5-1. If no response packet is received and T_resp period elapses, return to Step 5.1. If a valid response packet is received, proceed to Step 3.

5.5.2. Reception Procedure

This section describes the procedures to receive packets in the confirmed delivery. 2 procedures are described in this section: one to receive the first packet and the other to receive packets except for a first packet in the case that Circular Fragment Flag = "0". The packet reception procedure starts when the following conditions are satisfied.

In the case when the Header format in the packet is decoded and its CRC passes.

In the packet reception procedure, the following parameter is used.

Retry Packet Wait Time: T_retry

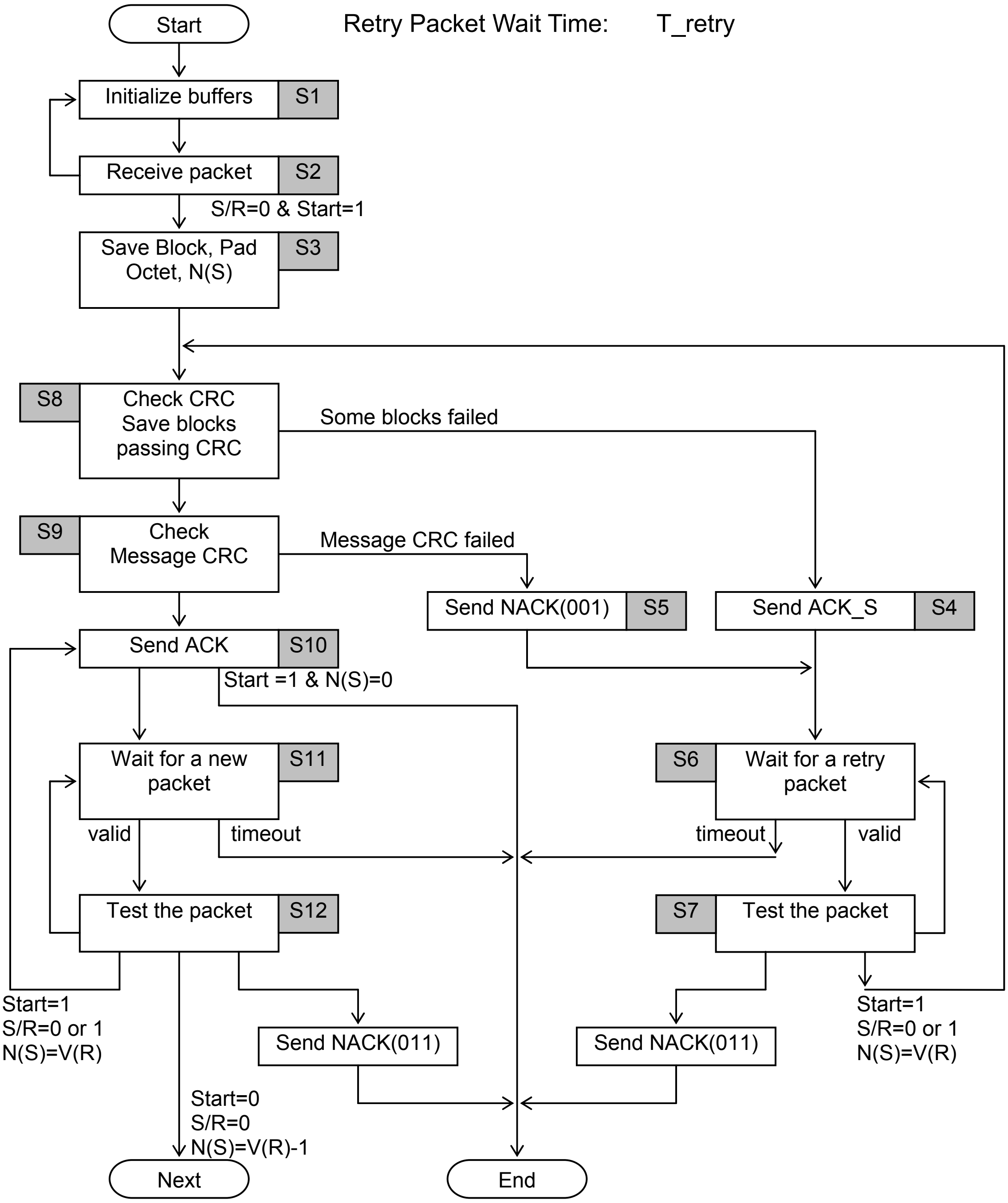


Figure 5.5-2 State Diagram of First Packet Reception in Confirmed Delivery

Step 1. Initialization

A buffer to store the packet is initialized. A receiver remains in a current state until a packet comes in.

Step 2. Reception of the First Packet

If Start Fragment Flag = "1" and Selective Retry Flag = "0", proceed to Step 3.

Otherwise since the received packet is invalid, discard the invalid packet and return to Step 1.

Step 3. Acquisition of Packet Parameters

The receiver determines the structure of received packet with the Block Count and Pad Octet Count, and the number of packets following the first packet with the Start Fragment Flag and TX Fragment Count which equals N(S).

If Start Fragment Flag = “1” and N(S) = 0, the packet is determined to be a single packet.

In order to distinguish the received packet from following packets, the receiver sets the N(S) to variable V(R).

Proceed to Step 8.

Step 4. Selective Retry

The receiver sends back a response packet of ACK_S which the Class is set to “01”, the Type is set to “001”, the RX Fragment Count is set to N(S), and the Error Block Flags are set as a Block Number needed to be retransmitted is specified.

Proceed to Step 6.

Step 5. Full Retry

The receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “001”, the RX Fragment Count is set to N(S), and all Error Block Flags are set to “1”.

Proceed to Step 6.

Step 6. Wait for Retransmission

Wait for a retry packet from a sender. If the retry packet is not received and T_retry period elapses, the procedure completes. If a valid retry packet is received, proceed to Step 7.

Step 7. Test of Retransmitted Packet

Step 7.1 Test after sending an ACK_S response

Case 1:

In the received packet, if Start Fragment Flag =1, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the same packet. This will be applied when a sender failed to receive an ACK_S response packet and sent the full retry packet after the T_resp period elapsed.

Proceed to Step 8.

Case 2:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 1 and N(S) = V(R), it means that the receiver has received the expected selective retry packet.

Proceed to Step 8.

Case 3:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 4:

Except for above cases, return to Step 6.

Step 7.2 Test after sending an NACK(001) response

Case 1:

In the received packet, if Start Fragment Flag = 1, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the expected full retry packet.

Proceed to Step 8.

Case 2:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 3:

Except for above cases, return to Step 6.

Step 8. CRC check and Storing a block passing the checks

If the CRC in the User Data format passes, the receiver updates the buffer for the received blocks according to the Block Number, and memorizes the Block Number which passes the CRC check in order not to request a Selective Retry for that block.

If there is a block that does not pass the CRC check, return to Step 4, and otherwise, proceed to Step 9.

Step 9. Message CRC check

After all blocks are received successfully, the receiver verifies the received packet by checking the Message CRC at the end. If the Message CRC does not pass, return to Step 5.

Otherwise, proceed to Step 10.

Step 10. Sending an ACK response

The receiver sends back an ACK response packet which the Class is set to “00”, the Type is set to “001” and the RX Fragment Count equals the N(S).

In Step 3, if Start Fragment Flag = 1 and N(S) = 0, in other words, if it is a single packet, the procedure completes. If there are multiple packets, proceed to Step 11.

Step 11. Wait for New Packet Transmission

Wait for transmission of the next packet by a sender. If no packet is received and T_retry period elapses, the procedure completes. If a valid packet is received, proceed to Step 12.

Step 12. Test of New Packet

Case 1:

In the received packet, if Start Fragment Flag = 1, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the same packet. This will be applied when a sender failed to receive an ACK response packet and retransmitted after the T_resp period elapsed.

Return to Step 10.

Case 2:

In the state that the receiver has sent back an ACK for a received selective retry packet during a selective retry process, if Start Fragment Flag = 1, Selective Retry Flag = 1, and N(S) = V(R) in the received packet, it means the receiver has received the same selective retry packet. This will be applied when a sender failed to receive an ACK response packet and retransmitted the selective retry packet after the T_resp period elapsed.

Return to Step 10.

Case 3:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 0 and N(S) = (V(R) - 1), it means that the receiver has received the expected new packet.

Proceed to reception procedure for non-first packets.

Case 4:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 5:

Except for above cases, return to Step 11.

Following is a description for the procedure to receive packets except the 1st packet.

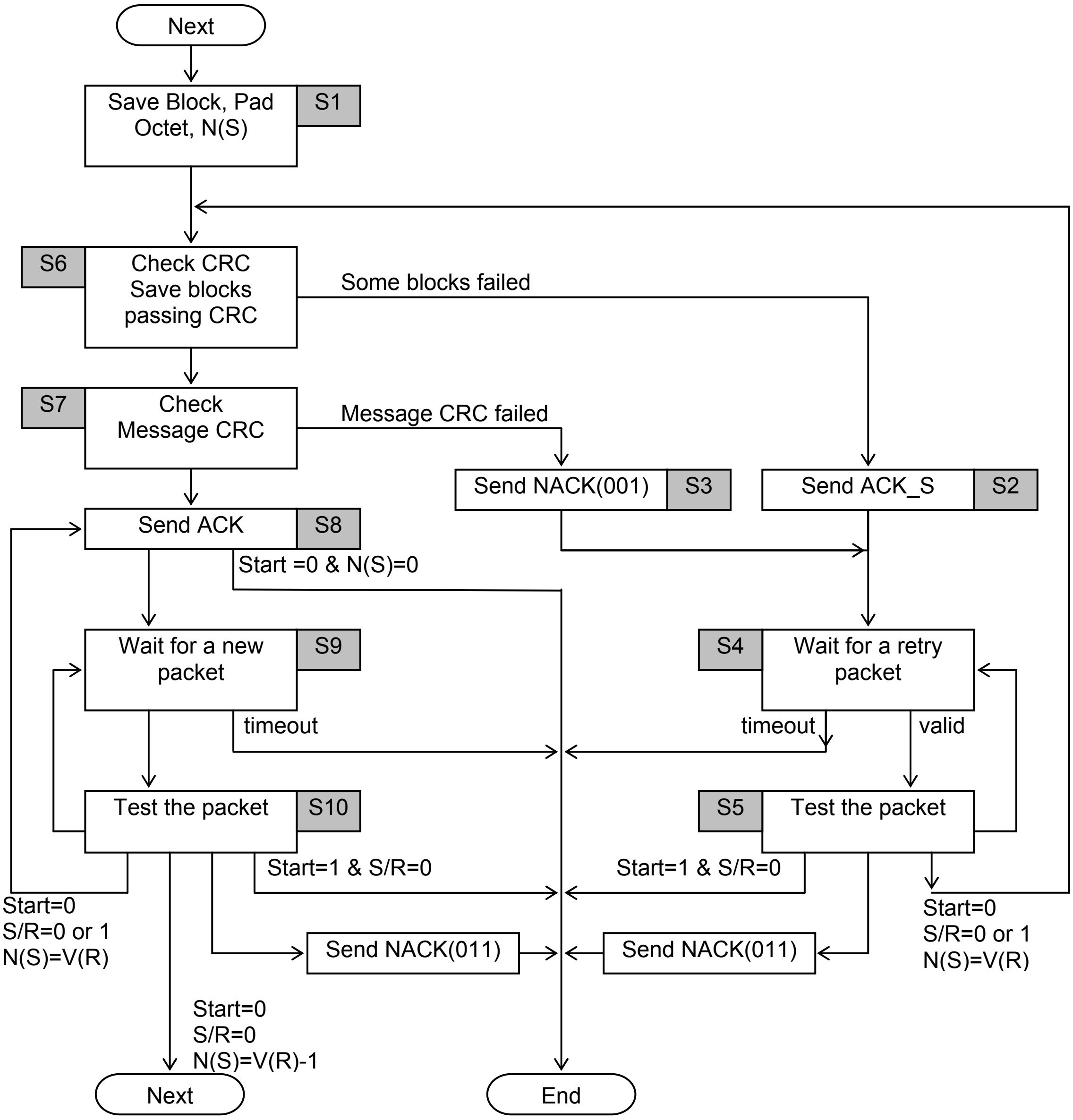


Figure 5.5-3 State Diagram for Non-First Packet Reception in Confirmed Delivery

Step1. Acquisition of Packet Parameters

The receiver determines the structure of received packets with the Block Count and Pad Octet Count, and confirms that Start Fragment Flag = 0 and N(S) = (V(R) - 1), and then recognizes as consecutive packets. If the N(S) value equals 0, it shall be determined to be the last of consecutive packets.

In order to distinguish the received packet from following packets, the receiver sets the N(S) to variable V(R).

Proceed to Step 6.

Step 2. Selective Retry

The receiver sends back a response packet of ACK_S which the Class is set to “01”, the Type is set to “001”, the RX Fragment Count is set to N(S), and the Error Block Flags are set as a Block Number needed to be retransmitted is specified.

Proceed to Step 4.

Step 3. Full Retry

The receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “001”, the RX Fragment Count is set to N(S), and all Error Block Flags are set to “1”.

Proceed to Step 4.

Step 4. Wait for Retransmission

Wait for a retry packet from a sender. If no retry packet is received and T_retry period elapses, the procedure completes. If a valid retry packet is received, proceed to Step 5.

Step 5. Test of Retransmitted Packet

Step 5.1 Test after sending an ACK_S response

Case 1:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the same packet. This will be applied when a sender failed to receive an ACK_S response packet and sent the full retry packet after the T_resp period elapsed.

Proceed to Step 6.

Case 2:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 1 and N(S) = V(R), it means that the receiver has received the expected selective retry packet.

Proceed to Step 6.

Case 3:

In the received packet, if Start Fragment Flag = 1, Selective Retry Flag = 0 it means that the receiver has received a new packet.

The receiver completes the procedure and starts a reception procedure for the first packet again.

Case 4:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 5:

Except for above cases, return to Step 4.

Step 5.2 Test after sending a NACK(001) response

Case 1:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the expected full retry packet.

Proceed to Step 6.

Case 2:

In the received packet, if Start Fragment Flag = 1, Selective Retry Flag = 0 it means that the receiver has received a new packet.

The receiver completes the procedure and starts a reception procedure for the first packet again.

Case 3:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 4:

Except for above cases, return to Step 4.

Step 6. CRC check and Storing a block after passing the checks

If the CRC of User Data format passes, the receiver updates the buffer for the received blocks according to the Block Number, and memorizes the Block Number which passes the CRC check in order not to request a Selective Retry for that block.

If there is a block that does not pass the CRC check, return to Step 2. Otherwise proceed to Step 7.

Step 7. Message CRC check

After all blocks are received successfully, the receiver verifies the received packet by checking the Message CRC at the end. If the Message CRC does not pass or retransmission is requested for other reasons, return to Step 3.

Otherwise, proceed to Step 8.

Step 8. Sending an ACK response

The receiver sends back an ACK response packet which the Class is set to “00”, the Type is set to “001” and the RX Fragment Count equals the N(S).

In Step 1, if Start Fragment Flag = 0 and N(S) = 0, in other words, if it is the last packet, the procedure completes. If it is a intermediate packet, proceed to Step 9.

Step 9. Wait for New Packet Transmission

Wait for transmission of the next packet by a sender. If no packet is received and T_retry period elapses, the procedure completes. If a valid packet is received, proceed to Step 10.

Step 10. Test of New Packet

Case 1:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 0 and N(S) = V(R), it means that the receiver has received the same packet. This will be applied when a sender failed to receive an ACK response packet and retransmitted after the T_resp period elapsed.

Return to Step 8.

Case 2:

In the state that the receiver has sent back an ACK for a received selective retry packet during a selective retry process, if Start Fragment Flag = 0, Selective Retry Flag

= 1, and N(S) = V(R) in the received packet, it means that the receiver has received the same selective retry packet. This will be applied when a sender failed to receive an ACK response packet and retransmitted the selective retry packet after the T_resp period elapsed.

Return to Step 8.

Case 3:

In the received packet, if Start Fragment Flag = 0, Selective Retry Flag = 0 and N(S) = (V(R) - 1), it means that the receiver has received the expected new packet.

Return to Step 1 and restart the reception procedure for non-first packets.

Case 4:

In the received packet, if Start Fragment Flag = 1, Selective Retry Flag = 0 it means that the receiver has received a new packet.

The receiver completes the procedure and starts a reception procedure for the first packet again.

Case 5:

If a fragment counter gets out of synchronization or a receiver aborts the reception procedure, the receiver sends back a NACK response packet which the Class is set to “11”, the Type is set to “011”, the RX Fragment Count is set to N(S) and the procedure completes.

Case 6:

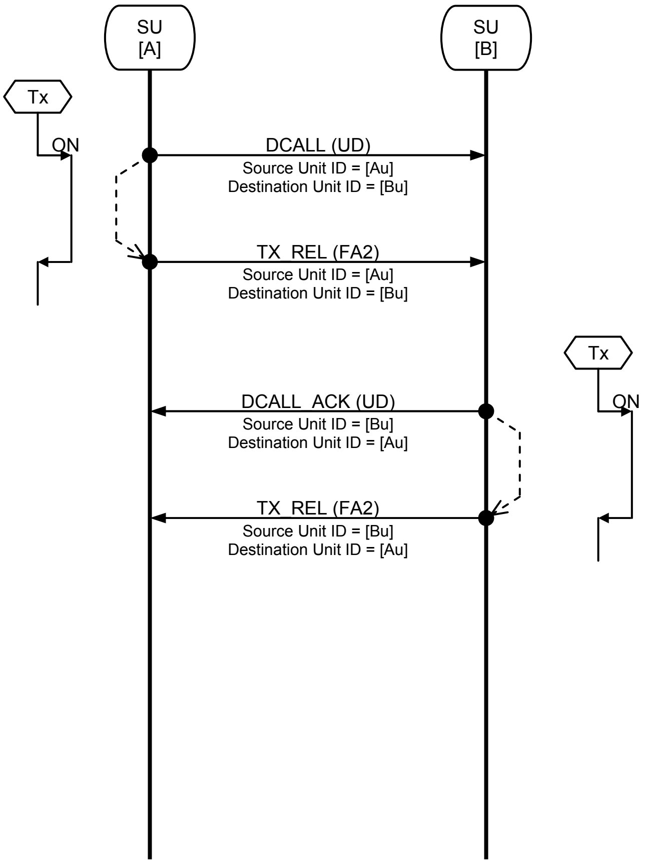
Except for above cases, return to Step 9.

5.6. Sequence Diagram

This section describes the sequence of the layer 3 message base in data calls

5.6.1. Confirmed Delivery Type

This figure shows the sequence when a SU [A] (Unit ID = Au) makes a confirmed individual data call to a SU [B] (Unit ID = Bu).



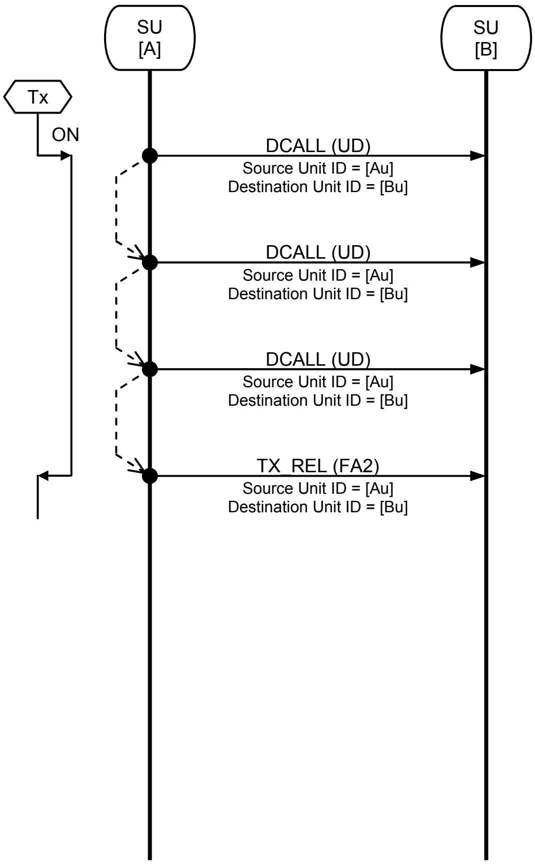
Note:

1) This is a sequence in the case of 1 packet. In the case that multiple packets are sent or in the case of retransmission, this sequence is repeated.
2) Although the DCALL message sent from SU [A] is represented by one arrow, it is considered that this includes a DCALL (Header) and given DCALL (User Data).

Figure 5.6-1 Data Call with confirmed type

5.6.2. Unconfirmed Delivery Type

This figure shows the sequence when a SU [A] (Unit ID = Au) makes an unconfirmed individual data call to a SU [B] (Unit ID = Bu).



Note:

1) This is a sequence in the case of three packets. Since no response exists in a unconfirmed delivery, a sender may send multiple packets consecutively. In the case that a sender returns to a receive mode after transmitting one packet such as Figure 5.6-1, the period of reception can be determined arbitrarily.
2) This sequence represents a case that multiple packets are sent consecutively, and in this case a TX_REL message shall be transmitted only at the end of transmission.
3) The same sequence shall be also used even for a broadcast data call addressed to a group using an unconfirmed delivery.

Figure 5.6-2 Data Call with unconfirmed type

6. Authentication

This section describes the authentication procedure in a conventional system.

In a conventional system, the following two authentication processes are available depending on an authenticating unit.

- Authentication by a CR for a SU.
- Authentication by a SU for another SU.

6.1. Authentication by a Conventional Repeater

If a CR (or a facility connected to a CR) has an authentication processing capability, it can apply an authentication procedure to a SU. Figure 6.3-1 shows the sequence of authentication process initiated by a CR.

6.1.1. Repeater Inquiry

An originating CR sends an AUTH_INQ_REQ to activate an authentication process of a SU.

6.1.2. Repeater Actions

The CR shall take the following actions as valid responses to the message from the SU.

a) If an AUTH_INQ_RESP is received, the CR:
   i) transfers Authentication Value of SU to a authentication facility.
   ii) Validation method is system-dependent, and the subsequent actions are not defined.

b) A optional timer can be configured, and if an AUTH_INQ_RESP is not received before the timer expires,
   i) subsequent actions are system-dependent and are not defined.

6.1.3. Unit Actions

A SU in idle shall take the following actions if Unit ID of the SU matches a Unit ID contained in a AUTH_INQ_REQ.

a) If an AUTH_INQ_REQ which matches Destination Unit ID is received, the SU:
   i) calculates the Authentication Value using the Authentication Parameter,
   ii) sends an AUTH_INQ_RESP to which the calculation result is embedded, and
   iii) returns to the idle state.

6.2. Authentication by a Subscriber Unit

If a SU (or a facility connected to a SU) has an authentication process capability, it can apply an authentication procedure to another SU. Figure 6.3-2 shows a sequence of an authentication process for another SU initiated by a SU.

6.2.1. Originating Unit Inquiry

An originating SU sends an AUTH_INQ_REQ to activate an authentication process of a target SU.

6.2.2. Repeater Actions

A CR only relays a message transmitted from SUs and does not control particularly.

6.2.3. Originating Unit Actions

The originating SU sending an AUTH_INQ_RESP shall take the following actions as valid responses to the message from the target SU.

a) If an AUTH_INQ_RESP is received, the originating SU:
   i) transfers the Authentication Value of a target SU to a authentication facility,
   ii) Validation method is system-dependent and are not defined.
b) A optional timer can be configured, and if an AUTH_INQ_RESP is not received before the timer expires:
   i) subsequent actions are system-dependent and are not defined .

6.2.4. Target Unit Actions

A SU in idle shall take the following actions if Unit ID of the SU matches a Unit ID contained in an AUTH_INQ_REQ.

a) If an AUTH_INQ_REQ which matches a Destination Unit ID is received, the SU:
   i) calculates the Authentication Value using the Authentication Parameter,
   ii) sends an AUTH_INQ_RESP to which the calculation result is embedded, and
   iii) returns to the idle state.

6.3. Sequence Diagrams

6.3.1. Authentication Inquiry by a Conventional Repeater

This figure shows a sequence when a CR (Unit ID = CRu) makes an authentication inquiry to a SU [A] (Unit ID = Au).

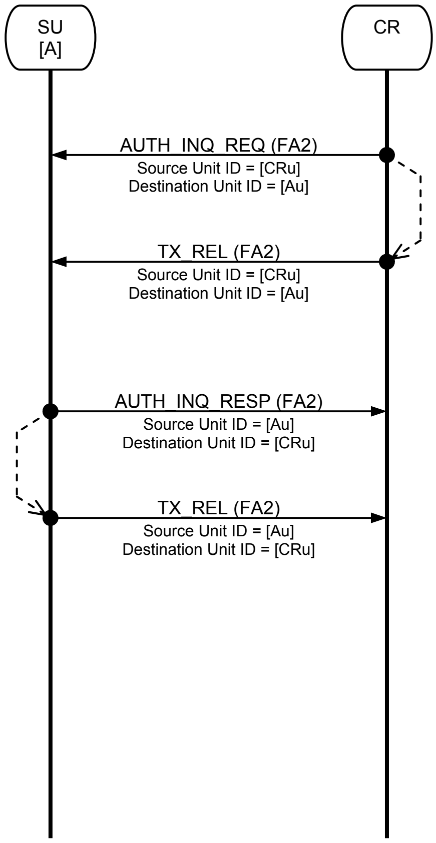
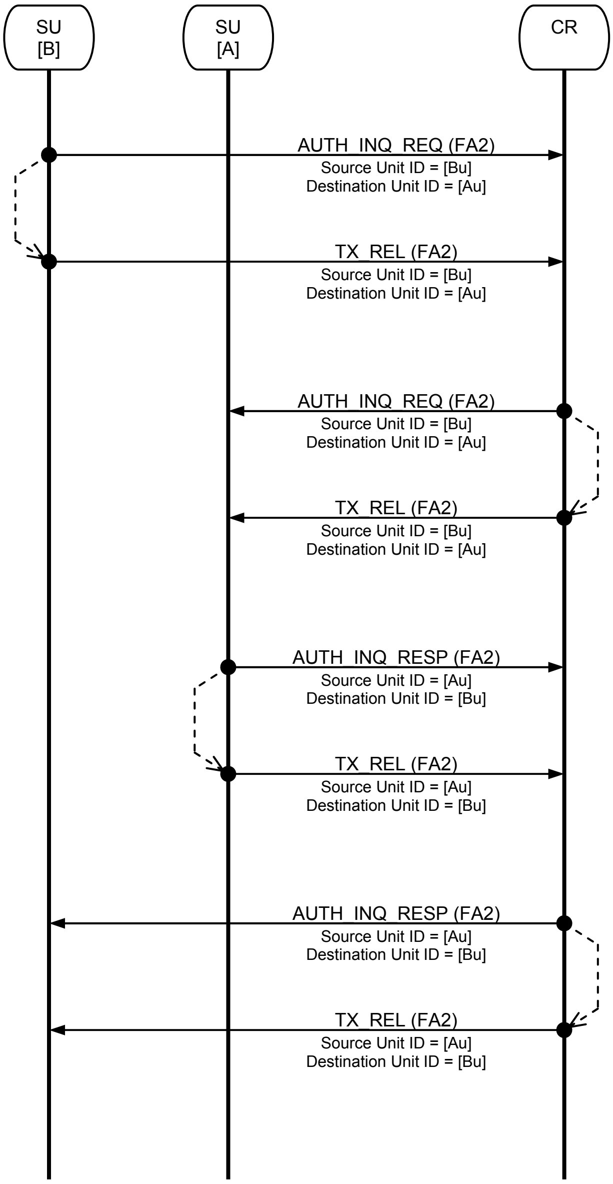


Figure 6.3-1 Authentication Inquiry by a CR

6.3.2. Authentication Inquiry by a Subscriber Unit

This figure shows a sequence when a SU [B] (Unit ID = Bu) makes an authentication inquiry with a SU [A] (Unit ID = Au).



Note:

1) A Hold Timer value of a CR is arbitrary. The sequence represents only a relay procedure of a message, and behavior during the Hold Time is omitted.

Figure 6.3-2 Authentication Inquiry by a SU

7. Repeater Operation

This section describes the basic operation of communication via a CR.

There are 2 types of repeater operation: a communication between SUs via a CR, and a communication between a console connected to a CR and a SU. Since this section is intended to describe how a CR responds to signals sent by a SU, it focuses on the communication between SUs via a CR. Since communication between a console and a SU is basically similar to direct mode operation between SUs, even though a CR is used, this section does not describe it specifically.

There are an inbound-driven repeater which starts an outbound transmission when a signal from a SU is received and an outbound-driven repeater which always or periodically transmits outbound signals as configuration of a CR. Since most CRs are the former (inbound-driven repeater), this section describes the operation for such an inbound-driven repeater.

The following is 3 typical modes of relay operation of repeater:

- RF relay
- Bit recovery relay
- Error correction relay

The RF relay system is a relay method that a received RF signal, or a received signal which is down-converted to an IF signal, is simply converted and relayed to the transmission frequency. The RF relay system is supposed to be used, for instance, to extend the coverage range in a mountainous area by relaying a signal with a bucket brigade-like operation. This method is just a frequency conversion, and does not involve any processing in relation to the protocol. Therefore, this relay system is not described here.

The bit recovery relay system is a relay method that a bit or symbol sequence detected and recovered from a received signal, which is handled as a bit or symbol sequence for reception, is relayed after simply a transmit baseband filtering. The bit recovery relay system is supposed to be used for the same applications as the RF relay system. Compared with the RF relay system, although the procedure is more complicated, the reliability is improved. Since this method does not also involve any processing in relation to the protocol, this relay system is not described here.

The error correction relay system is a relay method which an upper layer's data is restored from a recovered bit sequence by a FEC decoding and then the upper layer's data or the modified data is relayed after a series of processing necessary for transmission such as a FEC encoding. This method requires processing in relation to a protocol. Also, the error correction relay system is needed since a RAN contained in the upper layer's data is utilized to determine activation of relay operation and a LICH is utilized to specify the direction of inbound and outbound according to the REF [1]. Therefore, this method is one of the subjects to be described in this document.

7.1. Relay Conditions

This section describes how a CR determines to activate of relay operation.

In general, CTCSS and CDCSS are used by a conventional analog FM repeater to determine activation of relay operation. When a repeater decodes a CTCSS/CDCSS transmitted by a SU and the received CTCSS/CDCSS matches a CTCSS/CDCSS for a reception of the repeater, the repeater determines that relay operation can be activated and starts its outbound transmission. A CTCSS/CDCSS for a reception and transmission can be individually configured for a repeater, and a CTCSS/CDCSS used in a relay transmission by a repeater might differ from a CTCSS/CDCSS transmitted by a SU in some situations.

In an NXDN System, the Radio Access Number (RAN), which has a similar meaning as CTCSS/CDCSS in analog FM systems, is available. A maximum of 64 RANs (6 bits length) can be identified.

The relay operation is similar to that of a repeater for an analog FM system. When a CR decodes a RAN transmitted by a SU and the decoded RAN matches a RAN for reception of the CR, the CR determines that relay operation can be activated and starts its outbound transmission. A RAN for a reception and transmission can be individually configured for a CR, and a RAN used in a relay transmission by a CR might differ from a RAN transmitted by a SU in some situations.

Determination whether a CR activates its relay operation shall be basically subject to use of a RAN. The determination of relay activation using upper layer information such as Group ID and Unit ID for a selective call as well as a RAN is arbitrary. The RAN code is the only information which can be different between an inbound signal and an outbound signal, and upper layer information such as Group ID is relayed without modification.

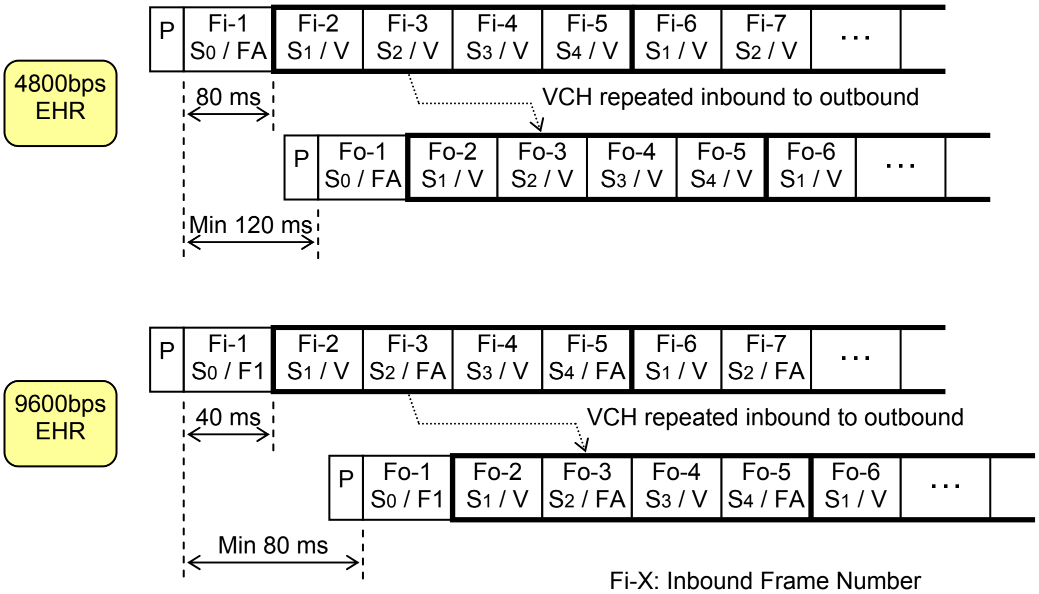
7.2. Timing

This section outlines timing of start and end of relay in a CR.

7.2.1. Relay Start Timing

As described in the layer 2 of REF [1], there is an offset of 40 ms at 4800 bps or 0 ms at 9600 bps between inbound and outbound frames, and a frame timing is generated based on an inbound signal. As shown in Figure 7.2-1, the shortest delay time of transporting data such as a VCH from an inbound frame to an outbound frame is 120ms at 4800bps or 80ms at 9600bps. Also, as described in REF [3], since an encrypted voice call have a relationship between the encryption cycle and the VCH placement, contents of an inbound frame shall be transported to an outbound frame without losing the relationship.

As described in Section 4.1, a given preamble prior to a 1st frame is sent as an inbound signal from a SU. Similarly, a given preamble prior to a 1st frame shall be sent as an outbound signal from a CR. Even if the CR is in state of Late Entry, the given preamble shall be sent.



Fi-X: Inbound Frame Number
Fo-X: Outbound Frame Number
P: Preamble
S_0: Single SACCH in non-superframe
S_1 - S_4: Four SACCH in superframe
FA: FACCH1
V: VCH

Figure 7.2-1 Timing of Repeat Operation

7.2.2. Relay Termination Timing

As presented in Section 4.1, a SU terminates its transmission after sending out a TX_REL in the last frame. A CR starts a termination procedure of a relay operation by receiving a TX_REL as a trigger. A CR can arbitrarily configure a Hold Time and it terminates the outbound transmission when this Hold Time has elapsed.

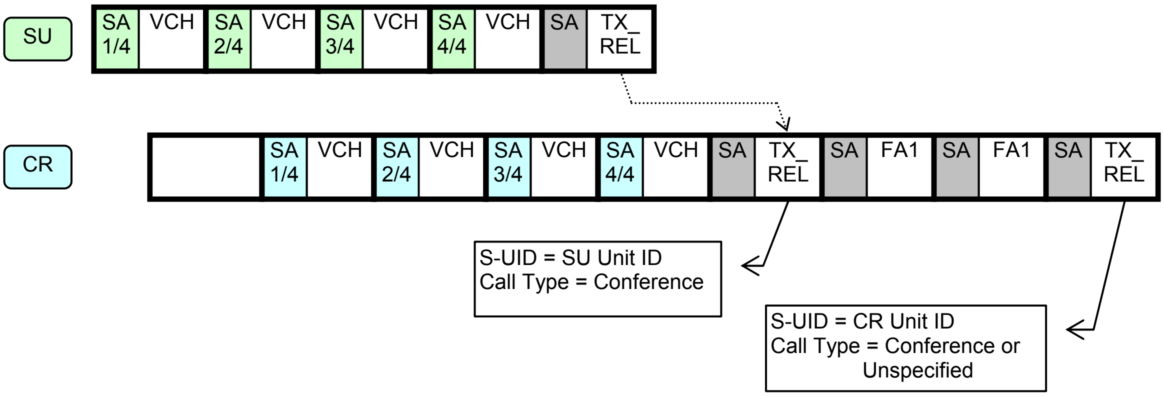


Figure 7.2-2 Repeat Termination Operation

Figure 7.2-2 shows an operation example when terminating a relay of a Group Voice Call at 4800bps. This is a recommended operation when the call is terminated properly, and different operation may be employed for an abnormal case or even for a normal case.

A CR keeps relaying until the frame which contains a TX_REL sent by a SU, and then it terminates its relay operation after sending out a TX_REL when the Hold Time arbitrarily configured elapses. As a setting for a TX_REL sent by CR, the Source Unit ID element uses Unit ID of a CR and the Call Type element uses either the same value relayed or “Unspecified Call”. A message to be sent during a Hold Time is not particularly specified.

7.2.3. SU Timing during Outbound Signal Synchronization

As described above, a SU normally starts the transmission at its own timing and a CR starts relaying by synchronizing the inbound signal from a SU. As a different state from above state, when a SU starts transmitting during a Hold Time, or when a SU overrides and transmits while other SU is transmitting, the SU shall start transmitting by synchronizing the outbound signals from a CR. As described in layer 2 of REF [1], since these cases consider an outbound signal of a CR as reference, a SU shall transmit the inbound frame with an offset of 40 ms at 4800 bps or 0 ms at 9600 bps to the outbound frame.

8. Appendix

8.1. Setting Guide for Conventional Operation

This section describes the configuration of a message in various communications on an RDCH.

8.1.1. Setting Guide for Voice Call

This section describes the configuration of a message in various voice calls.

8.1.1.1. Unaddressed Group Call

This section describes the configuration of the information elements of a transmitter during a voice call using only a RAN.

Item	Calling Unit Setting		
	4800bps/EHR	9600bps/EHR	9600bps/EFR
RAN	Any	←	←
VCALL			
CC Option	00000000	←	←
Call Type	001	←	←
Voice Call Option	00 000	00 010	00 011
Source ID	Calling Unit ID	←	←
Destination ID	Null Group ID	←	←
Cipher Type	00	←	←
Key ID	000000	←	←

Table 8.1-1 Settings for Unaddressed Group Call

8.1.1.2. Group Call

This section describes the configuration of the information elements of a transmitter during a group voice call.

Item	Calling Unit Setting		
	4800bps/EHR	9600bps/EHR	9600bps/EFR
RAN	Any	←	←
VCALL			
CC Option	00000000	←	←
Call Type	001	←	←
Voice Call Option	00 000	00 010	00 011
Source ID	Calling Unit ID	←	←
Destination ID	Any Standard Group ID		
Cipher Type	00	←	←
Key ID	000000	←	←

Table 8.1-2 Settings for a Group Call

8.1.1.3. All Call

This section describes the configuration of the information elements of a transmitter during a voice call to all units.

Following information element is changed in Table 8.1-2.

Destination ID	1111 1111 1111 1111 (All Group)

8.1.1.4. Individual Call

This section describes the configuration of the information elements of a transmitter during an individual voice call.

Item	Calling Unit Setting		
	4800bps/EHR	9600bps/EHR	9600bps/EFR
RAN	Any	←	←
VCALL			
CC Option	00000000	←	←
Call Type	100	←	←
Voice Call Option	00 000	00 010	00 011
Source ID	Calling Unit ID	←	←
Destination ID	Any Standard Unit ID		
Cipher Type	00	←	←
Key ID	000000	←	←

Table 8.1-3 Settings for an Individual Call

8.1.1.5. Emergency Call

This section describes the configuration of the information elements of a transmitter during an emergency voice call.

Following information element is changed in Table 8.1-1, in Table 8.1-2, and in Table 8.1-3.

CC Option	10000000

8.1.1.6. Encrypted Call

This section describes the configuration of the information elements of a transmitter during a voice call using encryption.

Following information elements are changed in Table 8.1-1, in Table 8.1-2, and in Table 8.1-3.

Cipher Type	01 (Scramble) / 10 (DES) / 11 (AES)
Key ID	Any

8.1.2. Settings Guide for Status Call

This section describes the configuration of the message in various status calls.

8.1.2.1. Broadcast Status Message

This section describes the configuration of the information elements of a transmitter when sending a status message to a group.

Item	Calling Unit Setting
STAT_REQ	
CC Option	00000000
Call Type	001
Status Call Option	00000
Source ID	Calling Unit ID
Destination ID	Any Standard Group ID
Status	Any valid User-definable Status

Table 8.1-4 Setting for Broadcast Status Message

8.1.2.2. Status Message

This section describes the configuration of the information elements of a transmitter when sending a status message to a unit.

Item	Calling Unit Setting	
	Unconfirmed	Confirmed
STAT_REQ		
CC Option	00000000	←
Call Type	100	←
Status Call Option	00000	01000
Source ID	Calling Unit ID	←
Destination ID	Any Standard Unit ID	←
Status	Any valid User-definable Status	

Table 8.1-5 Settings for a Status Message

This section describes the configuration of the information elements of a transmitter when sending back a status response in response to a confirmed status message.

Item	Called Unit Setting	
	ACK	NACK
STAT_RESP		
CC Option	00000000	←
Call Type	100	←
Source ID	Called Unit ID	←
Destination ID	Calling Unit ID	←
Cause (SS)	000 0001	000 1000

Table 8.1-6 Settings for a Status Response

8.1.2.3. Emergency Status Message

This section describes the configuration of the information elements of a transmitter when sending an emergency status message.

Following information elements are changed in Table 8.1-4, in Table 8.1-5, and in Table 8.1-6.

CC Option	10000000
Status	1110 0000 (Emergency)
	Note: Refer to REF [1] for values of other Statuses in relation to Emergency.

8.1.2.4. Status Inquiry

This section describes the configuration of the information elements of a transmitter when sending a status inquiry to a Unit.

Item	Calling Unit Setting
STAT_INQ_REQ	
CC Option	00000000
Call Type	100
Status Call Option	01000
Source ID	Calling Unit ID
Destination ID	Any Standard Unit ID

Table 8.1-7 Settings for a Status Inquiry

This section describes the configuration of the information elements of a transmitter when sending back a status inquiry response in response to a status inquiry.

Item	Called Unit Setting
STAT_INQ_RESP	
CC Option	00000000
Call Type	100
Source ID	Called Unit ID
Destination ID	Calling Unit ID
Cause (SS)	000 0001
Status	Any valid User-definable Status

Table 8.1-8 Settings for a Status Inquiry Response

8.1.3. Setting Guide for Remote Control Command

This section describes configuration of the message in various remote control commands.

8.1.3.1. Broadcast Remote Control Command

This section describes the configuration of the information elements of a transmitter when sending the remote control command to a group.

Item	Calling Unit Setting
REM_CON_REQ	
CC Option	00000000
G/U	0
D	0
Control Command	Any valid Control Command for Group
Source ID	Calling Unit ID
Destination ID	Any Standard Group ID
Control Parameter	Depend on Control Command

Table 8.1-9 Settings for a Broadcast Remote Control Command

8.1.3.2. Remote Control Command

This section describes the configuration of the information elements of a transmitter when sending a remote control command to a unit.

Item	Calling Unit Setting	
	Unconfirmed	Confirmed
REM_CON_REQ		
CC Option	00000000	←
G/U	1	←
D	0	1
Control Command	Any valid Control Command for Unit	
Source ID	Calling Unit ID	←
Destination ID	Any Standard Unit ID	←
Control Parameter	Depend on Control Command	

Table 8.1-10 Settings for a Remote Control Command

This section describes the configuration of the information elements of a transmitter when sending back a remote control response in response to a remote control command.

Item	Called Unit Setting
REM_CON_RESP	
CC Option	00000000
G/U	1
Control Command	Received Control Command
Source ID	Called Unit ID
Destination ID	Calling Unit ID
Cause (SS)	000 0001

Table 8.1-11 Setting for Remote Control Response

8.1.4. Setting Guide for Data Call

This section describes the configuration of the message in various data calls.

8.1.4.1. Broadcast Long Data Call

This section describes the configuration of the information elements of a transmitter during data calls to a group.

Item	Calling Unit Setting	
	4800bps	9600bps
DCALL		
CC Option	00000000	←
Call Type	001	←
Data Call Option	00 000	00 010
Source ID	Calling Unit ID	←
Destination ID	Any Standard Group ID	←
Cipher Type	00	←
Key ID	000000	←
Packet Information	0000 xxxx xxxx xxxx xxxx xxxx	

Note:
“X” areas in Packet Information depend on contents of data call.

Table 8.1-12 Settings for Broadcast Long Data Call

8.1.4.2. Individual Long Data Call

This section describes the configuration of the information elements of a transmitter during data calls to a unit.

Item	Calling Unit Setting			
	Unconfirmed		Confirmed	
	4800bps	9600bps	4800bps	9600bps
DCALL				
CC Option	00000000	←	←	←
Call Type	100	←	←	←
Data Call Option	00 000	00 010	00 000	00 010
Source ID	Calling Unit ID	←	←	←
Destination ID	Any Standard Unit ID		←	←
Cipher Type	00	←	←	←
Key ID	000000	←	←	←
Packet Information	00x0 xxxx xxxx xxxx xxxx xxxx		10x0 xxxx xxxx xxxx xxxx xxxx	

Note:
“X” areas in Packet Information depend on contents of data call.

Table 8.1-13 Settings for Individual Long Data Call

This section describes the configuration of the information elements of a transmitter when sending back a data call acknowledge in response to a confirmed long data call.

Item	Called Unit Setting			
	ACK		NACK	
	4800bps	9600bps	4800bps	9600bps
DCALL_ACK				
CC Option	00000000	←	←	←
Call Type	100	←	←	←
Data Call Option	00 000	00 010	00 000	00 010
Source ID	Called Unit ID	←	←	←
Destination ID	Calling Unit ID	←	←	←
Resp. Information	0000 001x xxxx xxxx		0011 001x xxxx xxxx	
Error Block Flag	0000 0000 0000 0000		xxxx xxxx xxxx xxxx	

Note:
“X” areas in Response Information and Error Block Flag depend on contents of data call.

Table 8.1-14 Setting for Long Data Call Acknowledge

8.1.5. Simultaneous Broadcast Data Call

This section describes the configuration of the information elements of a transmitter during a simultaneous data call to a group.

Item	Calling Unit Setting	
	4800bps	9600bps
SDCALL_REQ		
CC Option	00000000	←
Call Type	001	←
Data Call Option	00 000	00 010
Source ID	Calling Unit ID	←
Destination ID	Any Standard Group ID	←
Cipher Type	00	←
Key ID	000000	←
Packet Information	0000 xxxx xxxx x100	

Note:
“X” areas in Packet Information depend on contents of data call.
Normally, Source & Destination IDs match the contents of VCALL.

Table 8.1-15 Settings for a Simultaneous Broadcast Data Call

8.1.5.1. Simultaneous Data Call

This section describes the configuration of the information elements of a transmitter during a simultaneous data call to a unit.

Item	Calling Unit Setting			
	Unconfirmed		Confirmed	
	4800bps	9600bps	4800bps	9600bps
SDCALL_REQ				
CC Option	00000000	←	←	←
Call Type	100	←	←	←
Data Call Option	00 000	00 010	00 000	00 010
Source ID	Calling Unit ID		←	←
Destination ID	Any Standard Unit ID		←	←
Cipher Type	00	←	←	←
Key ID	000000	←	←	←
Packet Information	0000 xxxx xxxx x100		1000 xxxx xxxx x100	

Note:
“X” areas in Packet Information depend on contents of data call.
Normally, Source & Destination IDs match the contents of VCALL.

Table 8.1-16 Settings for Simultaneous Data Call

This section describes the configuration of the information elements of a transmitter when sending back a simultaneous data call response in response to a confirmed simultaneous data call.

Item	Called Unit Setting			
	ACK		NACK	
	4800bps	9600bps	4800bps	9600bps
SDCALL_RESP				
CC Option	00000000	←	←	←
Call Type	100	←	←	←
Data Call Option	00 000	00 010	00 000	00 010
Source ID	Called Unit ID	←	←	←
Destination ID	Calling Unit ID	←	←	←
Cause (SS)	000 0001	←	000 1000	←

Table 8.1-17 Settings for a Simultaneous Data Call Response

8.1.5.2. Encrypted Call

This section describes the configuration of the information elements of a transmitter during a data call using encryption.

Following information elements are changed in Table 8.1-12 and Table 8.1-13 for Long Data Call.

Cipher Type	01 (Scramble) / 10 (DES) / 11 (AES)
Key ID	Any
Initialization Vector	Any valid values (this element is added only in DES/AES)

Following information elements are changed in Table 8.1-15 and Table 8.1-16 for Simultaneous Data Call.

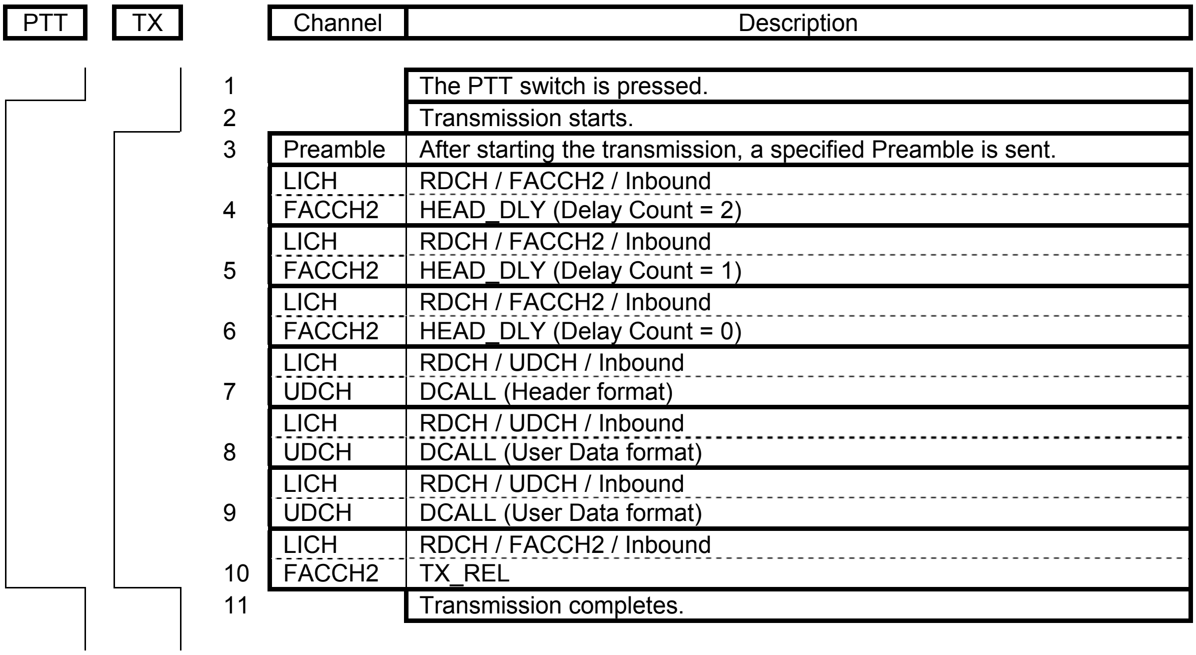
Cipher Type	01 (Scramble) / 10 (DES) / 11 (AES)
Key ID	Any

8.2. Delayed Header Operation

In data calls, since ID information is sent in the first frame, a data call is accomplished when the first frame is able to be received. Therefore if a receiver is in a scan operation, the probability that the receiver can acquire the first frame gets lower. The certainty of a data call is improved by sending a HEAD_DLY message containing ID information as a dummy frame. A HEAD_DLY message can be used in non-voice communications including a status call.

8.2.1. Transmission Operation

This section describes how to send a HEAD_DLY message in a data call. Figure 8.2-1 shows the operation in the case where 3 HEAD_DLY messages are appended to a normal data call procedure described in Figure 4.2-1.



	Channel	Description
1		The PTT switch is pressed.
2		Transmission starts.
3	Preamble	After starting the transmission, a specified Preamble is sent.
	LICH	RDCH / FACCH2 / Inbound
4	FACCH2	HEAD_DLY (Delay Count = 2)
	LICH	RDCH / FACCH2 / Inbound
5	FACCH2	HEAD_DLY (Delay Count = 1)
	LICH	RDCH / FACCH2 / Inbound
6	FACCH2	HEAD_DLY (Delay Count = 0)
	LICH	RDCH / UDCH / Inbound
7	UDCH	DCALL (Header format)
	LICH	RDCH / UDCH / Inbound
8	UDCH	DCALL (User Data format)
	LICH	RDCH / UDCH / Inbound
9	UDCH	DCALL (User Data format)
	LICH	RDCH / FACCH2 / Inbound
10	FACCH2	TX_REL
11		Transmission completes.

Figure 8.2-1 Data Call Transmission on an RDCH

1. A SU starts the data call procedure by activating the PTT.
2. A SU starts a transmission.
3. After starting a transmission, a specified Preamble is sent out.
4. A HEAD_DLY message is sent using a FACCH2. Since the value of a Delay Count information element is set to “2”, this message indicates that there are two remaining frames until the DCALL (Header) is sent.

5. The 2nd HEAD_DLY is sent. By decrementing the Delay Count, this message indicates that there is one remaining frame until the DCALL (Header) is sent.
6. The 3rd HEAD_DLY is sent. By decrementing the Delay Count, this message indicates that there is no remaining frame until the DCALL (Header) is sent.
7. This step or later has a frame configuration for a normal data call and is the same procedure as that of Figure 4.2-1.

9. Revision History

Version	Date	Revised Contents
1.0	Oct 26 2007	Original release
1.1	Feb 26 2008	Descriptions about RAN code were added to Section 3. Descriptions about SACCH were deleted in Section 4.1.7. Sequential diagram for data call was added to Section 5. Sequential diagram for authentication in Section 6 was modified. Operational example for relay termination was added to Section 7. Status Call Option were deleted from START_RESP and STAT_INQ in Section 4.1.7.
1.2	Jul 7 2008	Copyright was added. Packet procedure in section 5 was modified Procedure for HEAD_DLY was added as section 8.2
1.3	Nov 11 2011	Section 1: Add the description of Type-C. Section 4: Add sequence diagrams of Simul. Data Call and Status Call and the section for Remote Control. Modify the description of Rcvr SQ Condition. Section 5: Modify descriptions of Section 5.3.2 and Step 1.2 in Section 5.5.1. Section 8: Modify the configuration of Simul. Data Call. Descriptions of DES and AES are added in the related sections. (Section 3, 4.1.5, 4.1.9.3, 4.2.3, 5.1, 8.1.1.6 and 8.1.5.2)