Technical Document - Confidential

GSM General Packet Radio Services

Message Sequence Charts

SNDCP

Document Number:	8441.206.99.004
Version:	0.5
Status:	Draft
Approval Authority:	
Creation Date:	1999-Mar-12
Last changed:	2015-Mar-08 by XINTEGRA
File Name:	Sndcp.doc

Important Notice

Texas Instruments Incorporated and/or its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products, software and services at any time and to discontinue any product, software or service without notice. Customers should obtain the latest relevant information during product design and before placing orders and should verify that such information is current and complete.

All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment. TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI products, software and/or services. To minimize the risks associated with customer products and applications, customers should provide adequate design, testing and operating safeguards.

Any access to and/or use of TI software described in this document is subject to Customers entering into formal license agreements and payment of associated license fees. TI software may solely be used and/or copied subject to and strictly in accordance with all the terms of such license agreements.

Customer acknowledges and agrees that TI products and/or software may be based on or implement industry recognized standards and that certain third parties may claim intellectual property rights therein. The supply of products and/or the licensing of software does not convey a license from TI to any third party intellectual property rights and TI expressly disclaims liability for infringement of third party intellectual property rights.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products, software or services are used.

Information published by TI regarding third–party products, software or services does not constitute a license from TI to use such products, software or services or a warranty, endorsement thereof or statement regarding their availability. Use of such information, products, software or services may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Change History

Date	Changed by	Approved by	Version	Status	Notes
1999-Mar-12	Harald Krug		0.1		1
2000-Apr-28	Harald Krug		0.2		2
2000-Jun-02	Harald Krug		0.3		3
2001-Sep-14	Harald Krug		0.4		4
2003-Jun-03	XINTEGRA		0.5	Draft	

Notes:

1. Initial version
2. Corrected errors in 3.6.12, 3.6.13
3. S3, omitted deprecated sub clauses on block formats and deprecated MSCs
4. Fixed errors and cleared assumptions

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000] International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

[1] GSM 05.02 version 8.0.0 Release 1999
Digital cellular telecommunications system (Phase 2+);
Multiplexing and multiple access on the radio path

[2] GSM 04.60 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station (MS) - Base Station System (BSS) interface;
Radio Link Control/ Medium Access Control (RLC/MAC) protocol

[3] GSM 04.08 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface layer 3 specification

[4] GSM 03.64 version 6.1.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Overall description of the GPRS radio interface; Stage 2

[5] GSM 03.60 version 6.3.1 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Service description; Stage 2

[6] GSM 04.07 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface signalling layer 3; General aspects

[7] GSM 04.64 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station - Serving GPRS Support Node (MS-SGSN)
Logical Link Control (LLC) layer specification

[8] GSM 05.08 version 6.4.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem link control

[9] GSM 05.10 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem synchronization

[10] GSM 03.20 TS 100 929: July 1998 (GSM 03.20 version 6.0.1)
Security related network functions, ETSI

[11] Draft GSM 03.22: August 1998 (GSM 03.22 version 6.1.0)
Functions related to Mobile Station (MS) in idle mode and group receive mode, ETSI

[12] GSM 04.65 V6.3.0: Subnetwork Dependant Convergence Protocol
ETSI, March 1999

[13] ITU-T V42bis ITU-T, Recommendation V.42 bis 1990

[14] GSM 09.60 GPRS Tunneling Protocol (GTP) across the Gn and Gp Interface

[15] RFC 1661 IETF STD 51 July 1994
The Point-to-Point Protocol (PPP)

[16] RFC 1662 IETF STD 51 July 1994
PPP in HDLC-like Framing

[17] RFC 1570 January 1994
PPP LCP Extensions

[18] RFC 1989 August 1996
PPP Link Quality Monitoring

[19] RFC 1332 May 1992
The PPP Internet Protocol Control Protocol (IPCP)

[20] RFC 1877 December 1995
PPP IPCP Extensions for Name Server Addresses

[21] RFC 2153 May 1997
PPP Vendor Extensions

[22] RFC 1334 October 1992
PPP Authentication Protocols (for Password Authentication Protocol only)

[23] RFC 1994 August 1996
PPP Challenge Handshake Authentication Protocol (CHAP)

[24] TIA/EIA-136-370
Packet-Data Services – Enhanced General Packet Radio for TIA/EIA-136 (EGPRS-136) - Overview, Telecommunications Industry Association

[25] TIA/EIA-136-376
Packet-Data Services – EGPRS-136 Mobility Management, Telecommunications Industry Association

[26] TIA/EIA-136-972
Packet-Data Services – Stage 2 Description, Telecommunications Industry Association

1.2 Abbreviations

ACI	Application Control Interface
AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
C/R	Command/Response
C1	Path Loss Criterion
C2	Reselection Criterion
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CCI	Compression and Ciphering Interface
CHAP	Challenge Handshake Authentication Protocol
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DCOMP	Identifier of the user data compression algorithm used for the N-DPU
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
E	Extension bit
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
F	Final Bit
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GACI	GPRS Application Control Interface
GMM	GPRS Mobility Management
GP	Guard Period
GRR	GPRS RR
GSM	Global System for Mobile Communication
HDLC	High-level Data Link Control
HISR	High level Interrupt Service Routine
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ITU	International Telecommunication Union
IWF	Interworking Function
Kc	Ciphering Key
L	Length Indicator
LAI	Location Area Information
LCP	Link Control Protocol
LISR	Low level Interrupt Service Routine

LLC	Logical Link Control
LPD	Link Protocol Discriminator
LQM	Link Quality Monitoring
M	More bit used to indicate the last segment of N-DPU
MAC	Medium Access Control
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MT	Mobile Termination
N(R)	Receive Number
N(S)	Send Number
NC	Network Control
NCC	National Colour Code
NCP	Network Control Protocol
NECI	New Establishment Causes included
N-PDU	Network Protocol Data Unit
NSAPI	Network Layer Service Access Point Identifier
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PACCH	Packet Associated Control Channel
PAP	Password Authentication Protocol
PBCCH	Packet BCCH
PCCCH	Packet CCCH
PCOMP	Identifier of the protocol control information compression algorithm used for the N-DPU
PDCH	Packet Data Channel
PDP	Packet Data Protocol e.g. IP or X.25
PDTCH	Packet Data Traffic Channel
PRACH	Packet RACH
PSI	Packet System Information
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Data Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
PPC	Packet Physical Convergence
PPP	Point-to-Point Protocol
PTP	Point to Point
QoS	Quality of Service
RACH	Random Access Channel
REJ	Reject Frame
RLC	Radio Link Control
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
RTOS	Real Time Operating System
SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Stand alone Dedicated Control Channel
SDU	Service Data Unit
SGSN	Serving GPRS Support Node

SIM	Subscriber Identity Module
SM	Session Management
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SNDCP	Subnetwork Dependant Convergence Protocol
SNSM	SNDCP-SM
SS	Supplementary Services
TAP	Test Application Program
TBF	Temporary Block Flow
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TFI	Temporary Flow Identifier
TLLI	Temporary Logical Link Identifier
TMSI	Temporary Mobile Subscriber Identity
TOM	Tunnelling of Messages
TQI	Temporary Queuing Identifier
UA	Unnumbered Acknowledgement Frame
UART	Universal Asynchronous Receiver Transmitter
UI	Unnumbered Information Frame
USF	Uplink State Flag
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visited Public Land Mobile Network

1.3 Terms

Entity: Program which executes the functions of a layer

Message: A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.

Primitive: A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.

Service Access Point: A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

2 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The protocol stack for fax and data transmission consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity. The entity, which is described in this document, is coloured grey in the following figure :

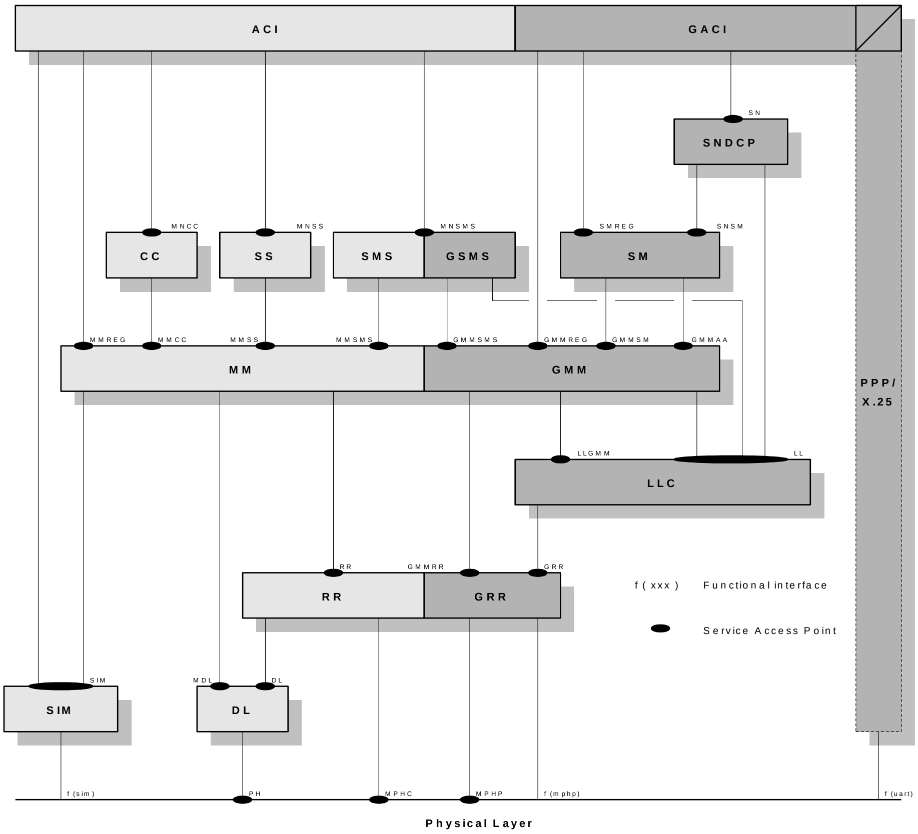

Figure 2-1: Architecture of the GSM/GPRS protocol stack

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the GPRS protocol stack are:

2.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control

This layer contains two functions: The Radio Link Control function provides a radio-solution-dependent reliable link. The Medium Access Control function controls the access signalling (request and grant) procedures for the radio channel, and the mapping of LLC frames onto the GSM physical channel.

2.2 LLC – Logical Link Control

The LLC entity provides multiple highly reliable logical links for asynchronous data transfer between the MS and the network. It supports variable-length information frames, acknowledged and unacknowledged data transfer, flow and sequence control, error detection and recovery, notification of unrecoverable errors, user identity confidentiality, and ciphering of user and signaling data.

2.3 GMM – GPRS Mobility Management

The GMM entity provides procedures for the mobility of the MS, such as informing the network of its present location, and user identity confidentiality. It manages the GMM context (attach, detach, routing area updating), supports security functions such as authentication of user and MS, controls ciphering of data, and initiates the response to paging messages.

2.4 SM – Session Management

The main function of the session management (SM) is to support PDP context handling of the user terminal. Session Management activates, modifies and deletes the contexts for packet data protocols (PDP). Session Management services are provided at the SMREG-SAP and the SNSM-SAP for anonymous and non-anonymous access. The non-anonymous and anonymous access procedures for PDP context activation and PDP context deactivation are available at the SMREG-SAP. In addition there exists a PDP context modification for non-anonymous PDP contexts.

2.5 SNDCP - Subnetwork Dependant Convergence Protocol

SNDCP carries out all functions related to transfer of Network layer Protocol Data Units (N-PDUs) over GPRS in a transparent way. SNDCP helps to improve channel efficiency by means of compression techniques. The set of protocol entities above SNDCP consists of commonly used network protocols. They all use the same SNDCP entity, which then performs multiplexing of data coming from different sources to be sent using the service provided by the LLC layer.

2.6 GACI – GPRS Application Control Interface

The GACI is the GPRS extension of the ACI. It is specified in GSM 07.07 and 07.60. It is responsible for processing of the GPRS related AT Commands to setup, activate and deactivate the PDP context parameter. It also provides functionality for the interworking between GMM/SM/SNDCP and a packet oriented protocol like PPP.

2.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the GPRS. The driver has to be able to perform flow control.

2.8 TOM – Tunnelling of Messages

The TOM entity is present if and only if HS136 is supported (the feature flag FF_HS136 is enabled).

The main function of TOM is to tunnel non-GSM signalling messages between the MS and the SGSN. The only non-GSM signalling which is currently supported by TOM is for the EGPRS-136 system (according to TIA/EIA-136-376). Data transfer in both uplink and downlink direction is possible. Two different priorities (high, low) of signalling data transfer are supported. TOM uses the unacknowledged mode of LLC and the acknowledged mode of GRR (RLC/MAC).

3 Introduction

[GSM 04.65, 7]

3.1 SN-PDU Formats

Omitted, See [GSM 04.65, 7.2]

3.2 SNDCP XID parameters

Omitted, see [GSM 04.65, 8]

3.3 Flow Control

A flow control mechanism not included in [GSM 04.65] has been established between SNDCP and LLC and between SNDCP and its user. Each place in this document where this mechanism applies is marked by the fact that the term “Flow control: “ stands before the sentence in question.

The main idea behind the flow control between SNDCP and LLC and between SNDCP and its user is that the capability to receive ONE next data unit (of a certain kind, for or from a certain SAPI) is indicated to the other layer by sending one of the flow control primitives defined in [LL SAP Document] and [SN SAP Document].

Important: Not in all MSCs the flow control primitives will be shown.

4 Protocol

Also to be considered:

<R.SNDCP.COMBINAT.A.001>, <R.SNDCP.COMBINAT.A.002>, <R.SNDCP.COMBINAT.A.003>, <R.SNDCP.COMBINAT.A.004>, <R.SNDCP.COMBINAT.A.005>, <R.SNDCP.COMBINAT.A.006>, <R.SNDCP.COMBINAT.A.007>, <R.SNDCP.COMBINAT.A.008>, <R.SNDCP.COMBINAT.A.009>, <R.SNDCP.COMBINAT.A.010>, <R.SNDCP.COMBINAT.A.011>, <R.SNDCP.COMBINAT.A.012>, <R.SNDCP.COMBINAT.A.0013> (possible combinations of SNDCP Protocol functions and their connection to service access points.

4.1 Establishment of acknowledged peer-to-peer LLC operation

<R.SNDCP.SERVICES.A.004>

4.1.1 Establishment upon reception of SNSM_ACTIVATE_IND

```
GACI                       SNDCP                       SM/LLC
     |                           |                           |
     |                           | SNSM_ACTIVATE_IND         |
     |                           *<==========================*
     |                         (SNDCP 1)                     |
     |                           | LL_ESTABLISH_REQ          |
     |                           *==========================>*
     |                           |                          (LLC 1)
     |                           |                           |
```

(SNDCP 1)
SNDCP receives an SNSM_ACTIVATE_IND from SM after an NSAPI has been activated for data transfer. The TLLI, the NSAPI, a negotiated QoS profile, the SAPI assigned for this NSAPI and the radio priority level to be used at RLC/MAC are also included with the primitive <R.SNDCP.CRITEST.M.005>. The QoS profile indicates an acknowledged peer-to-peer LLC operation to be requested.

(LLC 1)
SNDCP enters recovery state <R.SNDCP.CRITEST.M.006>. Both N-PDU numbers are reset to 0 <R.SNDCP.DT_ACK_G.A.005>. SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.007>. SNDCP

stores the mapping of the network layer entity in question onto the appropriate LLC connection (<R.SNDCP.SERVICES.A.003>, <R.SNDCP.NPDUMULT.A.001>). SNDCP sends an LL_ESTABLISH_REQ to LLC to establish acknowledged peer-to-peer operation (<R.SNDCP.ESTRELGN.A.001>, <R.SNDCP.CRITEST.M.002>, <R.SNDCP.CRITEST.A.001>, <R.SNDCP.ESTPROC.A.001>). The TLLI from the SNSM_ACTIVATE_IND is included as parameter. An XID Requested parameter is included to deliver the requested SNDCP XID parameters to LLC (<R.SNDCP.SERVICES.A.010>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.002>, <R.SNDCP.XIDNEGGN.M.001>). The parameters have to be set according to the QoS profile given in the SNSM_ACTIVATE_IND. SNDCP suspends the transfer of SN-DATA and SN-UNITDATA primitives to the LLC SAP to which the LL_ESTABLISH_REQ is sent <R.SNDCP.ESTPROC.A.016>.

4.1.2 Mapping of activated PDP context after SNSM_ACTIVATE_IND

```
SNDCP                        LLC                          SM
  |                           |                           |
  |   SNSM_ACTIVATE_IND       |                           |
  *<======================================================*
(SNDCP 1)                     |                           |
  | LL_GETUNITDATA_REQ        |                           |
  *==========================>*                           |
  |                         (LLC 1)                       |
  |     LL_XID_REQ            |                           |
  *==========================>*                           |
  |                      (LLC 2)                          |
  |                           |                           |
  |                           |                           |
```

(SNDCP 1)
SNDCP receives an SNSM_ACTIVATE_IND from SM after an NSAPI has been activated for data transfer. The TLLI, the NSAPI, a negotiated QoS profile, the SAPI assigned for this NSAPI and the radio priority level to be used at RLC/MAC are also included with the primitive <R.SNDCP.CRITEST.M.005>. The QoS profile indicates an unacknowledged peer-to-peer LLC operation to be requested. SNDCP stores the mapping of the network layer entity in question onto the appropriate LLC connection (<R.SNDCP.SERVICES.A.003>, <R.SNDCP.NPDUMULT.A.001>).

(LLC 1)
SNDCP signals its ability to receive one segment.

(LLC 2)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.007>. An LL_XID_REQ primitive is sent to initiate XID negotiation (<R.SNDCP.XIDNEGGN.M.001>, <R.SNDCP.SN_X_NEG.A.002>).

4.1.3 Re-establishment of acknowledged peer-to-peer LLC operation initiated by LLC (no N-PDUs buffered for this link)

```
SNDCP user                   SNDCP                       LLC
  |                           |                           |
  |                           |   LL_ESTABLISH_IND        |
  |                           *<==========================*
  |                        (SNDCP 1)                      |
  |                           |     LL_ESTABLISH_RES      |
  |                           *==========================>*
  |                           |                        (LLC 1)
  |                           |                           |
```

(SNDCP 1)
LLC initiates re-establishment of the acknowledged peer-to-peer LLC operation for a SAPI under situations described in [GSM 04.64] (<R.SNDCP.ESTRELGN.A.002>). There is no SNDCP XID block in the LL_ESTABLISH_IND and so according to GSM 4.65 SNDCP should not respond with an LL_ESTABLISH_RES primitive. But in our solution we send an LL_ESTABLISH_RES because LLC needs this for purposes of flow control synchronisation. No N-PDUs are by now buffered for the re-established link. SNDCP enters the recovery state. SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>.

4.1.4 Re-establishment of acknowledged peer-to-peer LLC operation initiated by LLC (LLC not ready)

```
SNDCP user                   SNDCP                       LLC
  |                           |                          |
  |                           |   LL_ESTABLISH_IND       |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |     LL_ESTABLISH_RES     |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |                          |
```

(SNDCP 1)
LLC initiates re-establishment of the acknowledged peer-to-peer LLC operation for a SAPI under situations described in [GSM 04.64] (<R.SNDCP.ESTRELGN.A.002>).

(LLC 1)
There is no SNDCP XID block in the LL_ESTABLISH_IND and so according to GSM 4.65 SNDCP should not respond with an LL_ESTABLISH_RES primitive. But in our solution we send an LL_ESTABLISH_RES because LLC needs this for purposes of flow control synchronisation. LLC is not ready to receive another SN-Data PDU. SNDCP enters the recovery state to (re-)send the first buffered N-PDU at the next opportunity <R.SNDCP.DT_ACK.A.016>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>.

4.1.5 Re-establishment of acknowledged peer-to-peer LLC operation initiated by LLC (N-PDUs buffered for this link and LLC ready)

```
SNDCP user                   SNDCP                       LLC
  |                           |                          |
  |                           |   LL_ESTABLISH_IND       |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |     LL_ESTABLISH_RES     |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |    LL_DATA_REQ           |
  |                           *=========================>*
  |                           |                       (LLC 2)
  |                           |                          |
```

(SNDCP 1)
LLC initiates re-establishment of the acknowledged peer-to-peer LLC operation for a SAPI under situations described in [GSM 04.64] (<R.SNDCP.ESTRELGN.A.002>).

(LLC 1)
There is no SNDCP XID block in the LL_ESTABLISH_IND and so according to GSM 4.65 SNDCP should not respond with an LL_ESTABLISH_RES primitive. But in our solution we send an LL_ESTABLISH_RES because LLC needs this for purposes of flow control synchronisation. N-PDUs are buffered for the re-established link and LLC is ready to receive another SN-Data PDU (<R.SNDCP.DT_ACK.A.014>).

(LLC 2) (Note: flow control issues not considered here!)
SNDCP enters recovery state <R.SNDCP.DT_ACK.A.016>. SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP sends an LL_DATA_REQ primitive containing the first segment of the next N-PDU in line (the “oldest” one for this link) to LLC (<R.SNDCP.DT_ACK.A.015>).

4.1.6 Reception of LL_ESTABLISH_IND (no N-PDUs buffered for this link)

```
SNDCP user                  SNDCP                       LLC
 |                            |                            |
 |                            |   LL_ESTABLISH_IND       |
 |                            *<=========================*
 |                         (SNDCP 1)                     |
 |                            |     LL_ESTABLISH_RES     |
 |                            *=========================>*
 |                            |                        (LLC 1)
 |                            |                          |
```

(SNDCP 1)
SNDCP receives an LL_ESTABLISH_IND primitive indicating establishment or re-establishment of acknowledged peer-to-peer operation for a SAPI in the LLC layer (<R.SNDCP.REESTACK.A.002>, <R.SNDCP.SN_X_NEG.A.004>). The primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>). No N-PDUs are by now buffered for this link.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP enters the recovery state. SNDCP responds with an LL_ESTABLISH_RES (<R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>).

4.1.7 Reception of LL_ESTABLISH_IND (LLC not ready)

```
SNDCP user                  SNDCP                       LLC
 |                            |                            |
 |                            |   LL_ESTABLISH_IND       |
 |                            *<=========================*
 |                         (SNDCP 1)                     |
 |                            |     LL_ESTABLISH_RES     |
 |                            *=========================>*
 |                            |                        (LLC 1)
 |                            |                          |
```

(SNDCP 1)
SNDCP receives an LL_ESTABLISH_IND primitive indicating establishment or re-establishment of acknowledged peer-to-peer operation for a SAPI in the LLC layer (<R.SNDCP.REESTACK.A.002>, <R.SNDCP.SN_X_NEG.A.004>). The primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>). LLC is not ready to receive another SN-Data PDU.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP enters the recovery state to (re-)send the first buffered N-PDU at the next opportunity <R.SNDCP.DT_ACK.A.016>.SNDCP responds with an LL_ESTABLISH_RES (<R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>).

4.1.8 Reception of LL_ESTABLISH_IND (N-PDUs buffered for this link and LLC ready)

```
 SNDCP user                 SNDCP                       LLC
  |                           |                          |
  |                           |   LL_ESTABLISH_IND       |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |     LL_ESTABLISH_RES     |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |    LL_READY_IND          |
  |                           *<=========================*
  |                          (SNDCP 2)                   |
  |                           |    LL_DATA_REQ           |
  |                           *=========================>*
  |                           |                       (LLC 2)
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an LL_ESTABLISH_IND primitive indicating establishment or re-establishment of acknowledged peer-to-peer operation for a SAPI in the LLC layer (<R.SNDCP.REESTACK.A.002>, <R.SNDCP.SN_X_NEG.A.004>). The primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>). N-PDUs are buffered for this link and LLC is ready to receive another SN-Data PDU (<R.SNDCP.DT_ACK.A.014>).

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP enters recovery state <R.SNDCP.DT_ACK.A.016>. SNDCP responds with an LL_ESTABLISH_RES (<R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>).

(SNDCP 2)
LLC sends an LL_READY_IND.

(LLC 2)
SNDCP sends an LL_DATA_REQ primitive containing the first segment of the next N-PDU in line (the “oldest” one for this link) to LLC (<R.SNDCP.DT_ACK.A.015>).

4.1.9 Successful Establishment confirmed to originator

```
GACI                        SNDCP                        SM/LLC
  |                           |                          |
  |                           |     LL_ESTABLISH_CNF     |
  |                           *<=========================*
  |                       (SNDCP 1)                      |
  |  SN_READY_IND             |                          |
  *<==========================*                          |
(GACI 1)                      |                          |
  |                           |  SNSM_ACTIVATE_RES       |
  |                           *=========================>*
  |                           |                        (SM 1)
  |                           |                          |
```

(SNDCP 1)
LLC on the originating side of an establishment procedure confirms success using an LL_ESTABLISH_CNF primitive (<R.SNDCP.SN_X_NEG.A.009>). SNDCP from now on applies the new parameter values (<R.SNDCP.SN_X_NEG.A.010>). SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>.

(GACI 1)
SNDCP is now ready to receive the first N-PDU from the user and indicates this to GACI.

(SM 1)
SNDCP confirms the success to SM (No Requirement, is only mentioned in the primitive description of SNSM_ACTIVATE_RES).

4.1.10 Successful Re-establishment confirmed to originator

```
SNDCP                        LLC                        SM
  |                            |                          |
  |      LL_ESTABLISH_CNF      |                          |
  *<==========================*                           |
 (SNDCP 1)                     |                          |
  |   SN_READY_IND             |                          |
  *<==========================*                           |
(GACI 1)                       |                          |
  | LL_DATA_REQ                |                          |
  *==========================>*                           |
  |                           (LLC 1)                     |
  |                            |                          |
```

(SNDCP 1)
LLC on the originating side of a re-establishment procedure confirms success using an LL_ESTABLISH_CNF primitive (<R.SNDCP.SN_X_NEG.A.009>). SNDCP from now on applies the new parameter values (<R.SNDCP.SN_X_NEG.A.010>). All NSAPIs mapped to the affected SAPI shall enter the recovery state <R.SNDCP.SN_X_NEG.A.013>.

(GACI 1)
SNDCP is now ready to receive the first N-PDU from the user and indicates this to GACI.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. All buffered N-PDUs (i.e. the ones whose complete reception has not been acknowledged and the ones that have not been transmitted yet) shall be transmitted starting with the oldest N-PDU <R.SNDCP.SN_X_NEG.A.014>. Also all compression entities using acknowledged peer-to-peer LLC operation on this SAPI are reset <R.SNDCP.SN_X_NEG.A.015>. The sent LL_DATA_REQ primitive contains a QoS parameter with "peak throughput" as the only field set <R.SNDCP.DT_ACK.M.017> and a radio priority parameter indicating the level of radio priority to be used by RLC/MAC <R.SNDCP.DT_ACK.M.018>.

4.1.11 Failed Establishment with cause "DM received"

```
SNDCP                        LLC                        SM
  |                            |                          |
  |      LL_RELEASE_IND        |                          |
  |(LL_RELCAUSE_DM_RECEIVED)  |                          |
  *<==========================*                           |
 (SNDCP 1)                     |                          |
  | SNSM_STATUS_REQ                                       |
  |  (SNSM_RELCAUSE_DM_RECEIVED)                          |
  *=====================================================>*
  |                            |                       (SM 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP receives an LL_RELEASE_IND from SM with cause "DM received" (<R.SNDCP.ESTRELGN.A.002>).

(SM 1)
SNDCP informs the SM sub layer using the SNSM_STATUS_REQ primitive with Cause "DM received". SM shall then deactivate all PDP contexts for that SAPI requiring acknowledged peer-to-peer LLC operation (<R.SNDCP.ESTABEX.A.001>).

4.1.12 Failed Establishment with cause “invalid XID received”, version I: LL_RELEASE_IND

```
SNDCP                        LLC                        SM
  |                           |                          |
  |     LL_RELEASE_IND        |                          |
  |( LL_RELCAUSE_INVALID_XID) |                          |
  *<==========================*                          |
 (SNDCP 1)                    |                          |
  | SNSM_STATUS_REQ                                      |
  | (LL_RELCAUSE_INVALID_XID)                            |
  *=====================================================>*
  |                           |                        (SM 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an LL_RELEASE_IND from SM with cause “invalid XID received” (<R.SNDCP.ESTRELGN.A.002>).

(SM 1)
SNDCP informs the SM sub layer using the SNSM_STATUS_REQ primitive with Cause “ invalid XID received ”. SM shall then deactivate all PDP contexts for that SAPI (<R.SNDCP.ESTABEX.A.002>).

4.1.13 Failed Establishment with cause “invalid XID received”, version II: LL_STATUS_IND

```
SNDCP                        LLC                        SM
  |                           |                          |
  |     LL_STATUS_IND         |                          |
  |( LL_RELCAUSE_INVALID_XID) |                          |
  *<==========================*                          |
 (SNDCP 1)                    |                          |
  | SNSM_STATUS_REQ                                      |
  | (LL_RELCAUSE_INVALID_XID)                            |
  *=====================================================>*
  |                           |                        (SM 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an LL_STATUS_IND from SM with cause “invalid XID received” (<R.SNDCP.ESTRELGN.A.002>).

(SM 1)
SNDCP informs the SM sub layer using the SNSM_STATUS_REQ primitive with Cause “ invalid XID received ”. SM shall then deactivate all PDP contexts for that SAPI (<R.SNDCP.ESTABEX.A.002>).

4.1.14 Failed Establishment with cause “no peer response”, version I: LL_RELEASE_IND

```
SNDCP                        LLC                        SM
  |                           |                          |
  |     LL_RELEASE_IND        |                          |
  |(LL_RELCAUSE_NO_PEER_RES)  |                          |
  *<==========================*                          |
 (SNDCP 1)                    |                          |
  | SNSM_STATUS_REQ                                      |
  | (LL_RELCAUSE_NO_PEER_RES)                            |
  *=====================================================>*
  |                           |                       (SM 1)
  *<Timeout T>                |                          |
  |  LL_ESTABLISH_REQ         |                          |
  *==========================>|                          |
  |                         (LLC 1)                      |
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an LL_RELEASE_IND from SM with cause “no peer response” (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTABEX.A.003>).

(SM 1)
SNDCP buffers, if possible, arriving N-PDUs (<R.SNDCP.ESTABEX.A.004>). SNDCP informs the SM sub layer using the SNSM_STATUS_REQ primitive with Cause “no peer response” (<R.SNDCP.ESTABEX.A.005>) and starts timer T (<R.SNDCP.ESTABEX.A.006>).

(LLC 1)
Timer T expires. SNDCP re-invokes the establishment procedure (<R.SNDCP.SERVICES.A.005>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTABEX.A.007>).

4.1.15 Failed Establishment with cause “no peer response”, version II: LL_STATUS_IND

```
SNDCP                        LLC                        SM
  |                           |                          |
  |     LL_STATUS_IND         |                          |
  |(LL_RELCAUSE_NO_PEER_RES)  |                          |
  *<==========================*                          |
 (SNDCP 1)                    |                          |
  | SNSM_STATUS_REQ                                      |
  | (LL_RELCAUSE_NO_PEER_RES)                            |
  *=====================================================>*
  |                           |                       (SM 1)
  *<Timeout T>                |                          |
  |  LL_ESTABLISH_REQ         |                          |
  *==========================>|                          |
  |                         (LLC 1)                      |
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an LL_STATUS_IND from SM with cause “no peer response” (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTABEX.A.003>).

(SM 1)
SNDCP buffers, if possible, arriving N-PDUs (<R.SNDCP.ESTABEX.A.004>). SNDCP informs the SM sub layer using the SNSM_STATUS_REQ primitive with Cause “no peer response” (<R.SNDCP.ESTABEX.A.005>) and starts timer T (<R.SNDCP.ESTABEX.A.006>).

(LLC 1)
Timer T expires. SNDCP re-invokes the establishment procedure (<R.SNDCP.SERVICES.A.005>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTABEX.A.007>).

4.1.16 Failed Establishment with cause “normal release”

```
SNDCP                        LLC                         SM
  |                            |                          |
  |     LL_RELEASE_IND         |                          |
  |    (LL_RELCAUSE_NORMAL)    |                          |
  *<===========================*                          |
 (SNDCP 1)                     |                          |
  | SNSM_STATUS_REQ                                       |
  | (LL_RELCAUSE_NORMAL)                                  |
  *======================================================>*
  |                            |                       (SM 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP receives an LL_RELEASE_IND from SM with cause “normal release” (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTABEX.A.009>).

(SM 1)
SNDCP buffers, if possible, all down-link N-PDUs for PDP contexts using the affected SAPI that require acknowledged peer-to-peer LLC operation (<R.SNDCP.SERVICES.A.005>, <R.SNDCP.ESTABEX.A.010>). Transfer of N-PDUs for PDP contexts that do not require acknowledged peer-to-peer LLC operation shall not be affected (<R.SNDCP.ESTABEX.A.011>). SNDCP informs SM using a SNSM_STATUS_REQ primitive with cause “normal release” (<R.SNDCP.ESTABEX.A.012>).

4.1.17 Failed Establishment because of Collision between LL_ESTABLISH_REQ and LL_ESTABLISH_IND (XID)

```
SNDCP user                   SNDCP                       LLC
  |                            |                          |
  |                            |   LL_ESTABLISH_IND       |
  |                            *<=========================*
  |                         (SNDCP 1)                     |
  |                            |     LL_ESTABLISH_RES     |
  |                            *=========================>*
  |                            |                       (LLC 1)
  |                            |                          |
```

(SNDCP 1)
After having sent an LL_ESTABLISH_REQ (<R.SNDCP.ESTPROC.A.001>), SNDCP receives an LL_ESTABLISH_IND instead of the expected LL_ESTABLISH_CNF. The LL_ESTABLISH primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.002>, <R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.002>).

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020> .SNDCP responds with an LL_ESTABLISH_RES containing the (possibly empty) negotiated XID parameters (<R.SNDCP.ESTRELGN.A.003>, <R.SNDCP.ESTPROC.A.003>, <R.SNDCP.ESTABEX.A.013>).

4.1.18 Failed Establishment because of Collision between LL_ESTABLISH_REQ and LL_ESTABLISH_IND (no XID)

```
SNDCP user                   SNDCP                       LLC
  |                           |                          |
  |                           |   LL_ESTABLISH_IND       |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |     LL_ESTABLISH_RES     |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
After having sent an LL_ESTABLISH_REQ (<R.SNDCP.ESTPROC.A.001>) with possibly empty XID block (<R.SNDCP.ESTPROC.A.002>), SNDCP receives an LL_ESTABLISH_IND instead of the expected LL_ESTABLISH_CNF. The LL_ESTABLISH primitive contains NO SNDCP XID parameter. There is no SNDCP XID block in the LL_ESTABLISH_IND and so according to GSM 4.65 SNDCP should not respond with an LL_ESTABLISH_RES primitive. But in our solution we send an LL_ESTABLISH_RES because LLC needs this for purposes of flow control synchronisation. If SNDCP had suspended the transfer of SN-DATA and SN-UNITDATA primitives then this transfer will be resumed <R.SNDCP.ESTABEX.A.020>. SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.009>.

4.2 Release of acknowledged peer-to-peer LLC operation

4.2.1 Release upon receipt of SNSM_DEACTIVATE_IND

```
SNDCP                        LLC                        SM
  |                           |                          |
  | SNSM_DEACTIVATE_IND                                  |
  *<=====================================================*
 (SNDCP 1)                    |                          |
  | LL_RELEASE_REQ            |                          |
  *==========================>|                          |
  |                          (LLC 1)                     |
  | LL_RELEASE_CNF            |                          |
  *<==========================|                          |
 (SNDCP 2)                    |                          |
  | SNSM_DEACTIVATE_RES                                  |
  *=====================================================>*
  |                           |                         (SM 1)
  |                           |                          |
```

(SNDCP 1)
SM sends an SNSM_DEACTIVATE_IND to inform SNDCP that an NSAPI has been deallocated (<R.SNDCP.RELCRIT.A.001>, <R.SNDCP.RELCRIT.A.002>, <R.SNDCP.ESTABEX.A.009>)and cannot be used by the SNDCP entity anymore. All buffered N-PDUs corresponding to this NSAPI are deleted (<R.SNDCP.NPDUBUF.A.002>).

(LLC 1)
The SNDCP layer releases the acknowledged peer-to-peer LLC operation by sending an LL_RELEASE_REQ primitive to LLC with Local parameter set (<R.SNDCP.ESTRELGN.A.001>).

(SNDCP 1)
LLC confirms the successfuk activation of the acknowledged LLC operation mode.

(SM 1)
SNDCP responds successful deactivation of context to SM.

4.2.2 Release initiated by the LLC layer

LLC may initiate the release of the acknowledged peer-to-peer LLC operation for a SAPI under situations described in [GSM 04.64]. The LLC layer shall inform the SNDCP layers of this release using the LL_RELEASE_INDICATION primitive. SNDCP shall process this indication as described in sub clauses 3.1.4 – 3.1.8 (<R.SNDCP.ESTRELGN.A.002>).

4.3 LLC Status Indication

4.3.1 Reception of LL_STATUS_IND

```
SNDCP                        LLC                        SM
  |                            |                          |
  | LL_STATUS_IND              |                          |
  *<==========================|                          |
  (SNDCP 1)                    |                          |
  | SNSM_STATUS_REQ                                       |
  *=====================================================>*
  |                            |                        (SM 1)
  |                            |                          |
```

(SNDCP 1)
Indication used by the LLC layer to inform SNDCP when an LLC error that cannot be corrected by the LLC layer has occurred. The Cause parameter indicates the cause of the failure.

(SM 1)
On receipt of LL-STATUS.indication, SNDCP shall inform the SM sub-layer by means of the SNSM-STATUS.request primitive <R.SNDCP.LLSTAT.A.001>.

4.4 PDP context modification

[13, 5.1.2.23][GSM 04.07, C.11] Also to be considered: <R.SNDCP.DT_ACK_G.A.012>, <R.SNDCP.DT_ACK_G.A.013>, <R.SNDCP.DT_ACK_G.A.006>

4.4.1 Context modification initiated by the network, no release or establish of acknowledged LLC operation

```
SNDCP user                   SNDCP                      SM
  |                            |                          |
  |                            | SNSM_MODIFY_IND          |
  |                            *<=========================*
  |                        (SNDCP 1)                      |
  |                            | SNSM_MODIFY_RES          |
  |                            *=========================>*
  |                            |                       (SM 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP receives an SNSM_MODIFY_IND to indicate a change of the QoS profile, SAPI or radio prio for an NSAPI <R.SNDCP.CONT_MOD.A.001>. From the given QoS profile SNDCP decides that no indication for a release or an establishment of acknowledged LLC operation is given.

(SM 1)
SNDCP sends back an SNSM_MODIFY_RES immediately. (See below for possible reasons for release or establishment.)

4.4.2 Context modification initiated by the network, release of acknowledged peer-tp-peer LLC operation

```
SNDCP                         LLC                        SM
  |                            |                          |
  | SNSM_MODIFY_IND                                       |
  *<======================================================*
 (SNDCP 1)                     |                          |
  | LL_RELEASE_IND             |                          |
  *<===========================|                          |
 (SNDCP 2)                     |                          |
  | SNSM_MODIFY_RES                                       |
  *======================================================>*
  |                            |                         (SM 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP receives an SNSM_MODIFY_IND to indicate a change of the QoS profile, SAPI or radio prio for an NSAPI <R.SNDCP.CONT_MOD.A.001>. From the given QoS profile SNDCP decides that the acknowledged peer-to-peer LLC operation for the affected NSAPI has to be released, but no new peer-to-peer LLC operation will have to be established. Possible reasons: The reliability class in the QoS profile changed from a class with acknowledged to a class with unacknowledged peer-to-peer LLC operation. The link to be released must not be used by any other PDP context using ack'ed mode.

(SNDCP 2)
SNDCP receives an LL_RELEASE_IND for the affected NSAPI <R.SNDCP.CONT_MOD.A.006>.

(SM 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.008>. All buffered N-PDUs shall be deleted and the Send N-PDU number shall be set to 0 <R.SNDCP.CONT_MOD.A.007>. SNDCP sends an SNSM_MODIFY_RES to inform the SM entity that the indicated NSAPI and QoS profile are now in use
and the acknowledged peer-to-peer LLC operation for the appropriate SAPI is released, if
necessary <R.SNDCP.CONT_MOD.A.007>, <R.SNDCP.CONT_MOD.A.001>. In addition, if the newly assigned SAPI is different from the original SAPI <R.SNDCP.CONT_MOD.A.010> then LL_DATA_IND, LL_DATASENT_IND, LL_DATA_CNF, LL_UNITDATA_IND received on the old SAPI shall be ignored <R.SNDCP.CONT_MOD.A.011>, LL_DATA_REQ and LL_UNITDATA_REQ shall be sent on the new SAPI <R.SNDCP.CONT_MOD.A.012>

4.4.3 Context modification initiated by the network, establishment of acknowledged peer-tp-peer LLC operation

```
SNDCP                         LLC                        SM
  |                            |                          |
  | SNSM_MODIFY_IND                                       |
  *<======================================================*
 (SNDCP 1)                     |                          |
  | LL_ESTABLISH_IND           |                          |
  *<===========================|                          |
 (SNDCP 2)                     |                          |
  | LL_ESTABLISH_RES           |                          |
  *===========================>*                          |
  |                           (LLC 1)                     |
  | SNSM_MODIFY_RES                                       |
  *======================================================>*
  |                            |                         (SM 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP receives an SNSM_MODIFY_IND to indicate a change of the QoS profile for an NSAPI. The indicated SAPI is not changed. From the given QoS profile SNDCP decides that the formerly unacknowledged peer-to-peer LLC operation for the affected NSAPI has to be changed to acknowledged mode. Possible reasons: The reliability class in the QoS profile changed from a class with unacknowledged to a class with acknowledged peer-to-peer LLC operation. (According to [13, 6.2.1, 6.2.2]

starting releasing/establishing of ack LLC operation mode as a reaction to an SNSM_MODIFY_IND is only done in the SGSN!).

(SNDCP 2)
SNDCP receives an LL_ESTABLISH_IND for the affected NSAPI <R.SNDCP.CONT_MOD.A.008>.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.008>. SNDCP sends back an LL_ESTABLISH_RES to LLC to respond to the indicated establishment and to deliver the negotiated SNDCP XID parameters. Since peer-to-peer LLC operation mode is changed from unacknowledged to acknowledged then both N-PDU numbers are reset to 0 <R.SNDCP.DT_ACK_G.A.006>.

(SM 1)
Send N-PDU number and Receive N-PDU number are set to 0 <R.SNDCP.CONT_MOD.A.009>. SNDCP sends an SNSM_MODIFY_RES to inform the SM entity that the indicated NSAPI and QoS profile are now in use
and the acknowledged peer-to-peer LLC operations for the appropriate SAPIs are established and/or released, if necessary <R.SNDCP.CONT_MOD.A.007>.

4.4.4 Context modification initiated by the network, release and establishment of acknowledged peer-to-peer LLC operation

```
SNDCP                        LLC                        SM
  |                           |                          |
  | SNSM_MODIFY_IND                                      |
  *<=====================================================*
 (SNDCP 1)                    |                          |
  | LL_RELEASE_IND            |                          |
  *<==========================|                          |
 (SNDCP 2)                    |                          |
  | LL_ESTABLISH_IND          |                          |
  *<==========================|                          |
 (SNDCP 3)                    |                          |
  | LL_ESTABLISH_RES          |                          |
  *==========================>*                          |
  |                          (LLC 1)                     |
  | SNSM_MODIFY_RES                                      |
  *=====================================================>*
  |                           |                         (SM 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP receives an SNSM_MODIFY_IND to indicate a change of the QoS profile, SAPI or radio prio for an NSAPI <R.SNDCP.CONT_MOD.A.001>. SNDCP decides to release an acknowledged peer-to-peer LLC operation and to establish another one. The only possible reason is that the affected SAPI is changed.

(SNDCP 2)
SNDCP receives an LL_RELEASE_IND for the affected NSAPI <R.SNDCP.CONT_MOD.A.006> indicating the need to release the link to the SAPI currently in use.

(SNDCP 3)
SNDCP receives an LL_ESTABLISH_IND for the affected NSAPI <R.SNDCP.CONT_MOD.A.008> indicating the need to establish a new acknowledged peer-to-peer LLC link to the SAPI given in the SNSM_MODIFY_IND primitive.

(LLC 1)
SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.008>. SNDCP sends back an LL_ESTABLISH_RES to LLC to respond to the indicated establishment and to deliver the negotiated SNDCP XID parameters.

(SM 1)
The Send N-PDU number for the released NSAPI shall be set to 0 <R.SNDCP.CONT_MOD.A.007>. The Send N-PDU number and Receive N-PDU number for the just established NSAPI shall be set to 0 <R.SNDCP.CONT_MOD.A.009>. SNDCP sends an SNSM_MODIFY_RES to inform the SM entity that the indicated NSAPI and QoS profile are now in use
and the acknowledged peer-to-peer LLC operations for the appropriate SAPIs are established and/or released <R.SNDCP.CONT_MOD.A.007>. In addition, if the newly assigned SAPI is different from the original SAPI

<R.SNDCP.CONT_MOD.A.010> then LL_DATA_IND, LL_DATASENT_IND, LL_DATA_CNF, LL_UNITDATA_IND received on the old SAPI shall be ignored <R.SNDCP.CONT_MOD.A.011>, LL_DATA_REQ and LL_UNITDATA_REQ shall be sent on the new SAPI <R.SNDCP.CONT_MOD.A.012> and (since acknowledged peer-to-peer LLC operation is used both before and after the receipt of the SNSM_MODIFY_IND <R.SNDCP.CONT_MOD.A.013>) all buffered N-PDUs (i.e. the ones whose complete reception has not been acknowledged and the ones that have not been transmitted yet) shall be transmitted starting from the oldest one <R.SNDCP.CONT_MOD.A.014>. Recovery state shall be entered <R.SNDCP.CONT_MOD.A.015>.

4.5 XID parameter negotiation

<R.SNDCP.SERVICES.A.010>, <R.SNDCP.ESTRELGN.A.004>

Also to be considered:

<R.SNDCP.XIDNEGG.A.008>(SNDCP version not negotiated),

<R.SNDCP.SN_X_NEG.A.003> (LLC issues XID command),

<R.SNCDP.C_ENTNEG.A.001>, <R.SNCDP.C_ENTNEG.A.002>, <R.SNCDP.C_ENTNEG.A.003>, <R.SNCDP.C_ENTNEG.A.004>, <R.SNCDP.C_ENTNEG.A.005>, <R.SNCDP.C_ENTNEG.A.006>, <R.SNCDP.C_ENTNEG.A.007>, <R.SNCDP.C_ENTNEG.A.008>, <R.SNCDP.C_ENTNEG.A.009> (cardinalities and mapping of compression entities),

<R.SNCDP.XID_PARM.A.001>, <R.SNCDP.XID_PARM.A.002>, <R.SNCDP.XID_PARM.A.003>, <R.SNCDP.XID_PARM.A.004>, <R.SNCDP.XID_PARM.A.005> (rules for parameter values)

<R.SNDCP.XID_PAR_X.A.001>, <R.SNDCP.XID_PAR_X.A.002>, <R.SNDCP.XID_PAR_X.A.003>, <R.SNDCP.XID_PAR_X.A.004>, <R.SNDCP.XID_PAR_X.A.011> (Exception handling for XID negotiation, error on originating side)

4.5.1 XID parameter negotiation initiated by the SNDCP user (only change of applicable NSAPIs)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.2 XID parameter negotiation initiated by the SNDCP user

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.3 LLC indicating the current values of N201-I and N201-U

```
SNDCP user                   SNDCP                       LLC
   |                           |                           |
   |                           |   LL_XID_IND              |
   |                           *<==========================*
   |                         (SNDCP 1)                     |
   |                           |                           |
```

(SNDCP 1)
SNDCP receives an LL_XID_IND primitive without an SNDCP-XID block. This primitive is used by LLC to comunicate the actual values for N201-I and N201-U. SNCDP stores these values (<R.SNCDP.LL_X_NEG.A.001>, <R.SNCDP.LL_X_NEG.A.002>).

4.5.4 Responding to proposed SNDCP XID parameters after LL_XID_REQ (SNDCP decides)

```
SNDCP user                   SNDCP                         LLC
  |                             |                           |
  |                             |   LL_XID_IND              |
  |                             *<==========================*
  |                          (SNDCP 1)                      |
  |                             |     LL_XID_RES            |
  |                             *==========================>*
  |                             |                        (LLC 1)
  |                             |                           |
```

(SNDCP 1)
On the receiving side of an SNDCP XID parameter negotiation initiated with an LL_XID_REQ on the originating side, LLC receives an XID command and indicates the SNDCP XID parameters to the SNDCP entity using LL_XID_IND including the requested XID parameters (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.SN_X_NEG.A.004>).

(LLC 1)
SNDCP responds with an LL_XID_RES including the negotiated XID parameters, IF the received primitive contains an SNDCP XID block (<R.SNDCP.XIDNEGG.A.007>, <R.SNDCP.SN_X_NEG.A.005>, <R.SNDCP.SN_X_NEG.A.008>). If the LL_XID_IND does not contain an SNDCP XID block but is only used to deliver new values of N201-I and N201_U then SNDCP shall not respond with the LL_XID_RES primitive (<R.SNDCP.ESTRELGN.A.004>). SNDCP sends the proposed SNDCP XID parameters to the LLC SAP with the LL_ESTABLISH_REQ. If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>. SNDCP from now on applies the parameter values it has included in its response (<R.SNDCP.SN_X_NEG.A.007>).

4.5.5 Responding to proposed SNDCP XID parameters after LL_XID_REQ (SNDCP asks SNDCP user for decision)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.6 Responding to proposed SNDCP XID parameters after LL_XID_REQ (SNDCP asks SNDCP user for decision)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.7 Delivering negotiated SNDCP XID parameters to the SNDCP user (originator changed acknowledged parameters)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.8 Delivering negotiated SNDCP XID parameters to the SNDCP user (responder changed acknowledged parameters)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.9 Re-negotiation with LL_XID_REQ after reception of invalid negotiated XID block

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |     LL_XID_CNF           |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |    LL_XID_REQ            |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |                          |
```

(SNDCP 1)
The SNDCP on the originating side of an XID parameter negotiation (with LL_XID_REQ) receives an invalid XID block in the terms of [13, 6.8.4](<R.SNDCP.SN_X_NEG.A.009>, <R.SNDCP.XID_PAR_X.A.005>). The number of re-negotiations has not exceeded an implementation-specific count.

(LLC 1)
SNDCP ignores the received SNDCP XID block (<R.SNDCP.XID_PAR_X.A.006>) and reinitiates the negotiation by sending another LL_XID_REQ (<R.SNDCP.SN_X_NEG.A.002>, <R.SNDCP.XID_PAR_X.A.007>). If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>.

4.5.10 Re-negotiation with LL_ESTABLISH_REQ after reception of invalid negotiated XID block

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |     LL_XID_CNF           |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |    LL_ESTABLISH_REQ      |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |                          |
```

(SNDCP 1)
The SNDCP on the originating side of an XID parameter negotiation (with LL_ESTABLISH_REQ) receives an invalid XID block in the terms of [13, 6.8.4](<R.SNDCP.SN_X_NEG.A.009>, <R.SNDCP.XID_PAR_X.A.005>). The number of re-negotiations has not exceeded an implementation-specific count.

(LLC 1)

SNDCP ignores the received SNDCP XID block (<R.SNDCP.XID_PAR_X.A.006>) and reinitiates the negotiation by sending another LL_ESTABLISH_REQ (<R.SNDCP.SN_X_NEG.A.002>, <R.SNDCP.XID_PAR_X.A.007>). If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>. SNDCP suspends the transfer ofSN-DATA and SN-UNITDATA primitives to the LLC SAP to which the LL_ESATBLISH_REQ is sent <R.SNDCP.SN_X_NEG.A.005>.

4.5.11 Context deactivation after repeated failed Re-negotiation

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.12 Collision between LL_XID_REQ and LL_XID_IND (XID, SNDCP decides), renegotiation not necessary

```
SNDCP user                   SNDCP                       LLC
  |                           |                           |
  |                           |   LL_XID_IND              |
  |                           *<==========================*
  |                        (SNDCP 1)                      |
  |                           |     LL_XID_RES            |
  |                           *==========================>*
  |                           |                        (LLC 1)
  |                           |                           |
```

(SNDCP 1)
After having sent an LL_XID_REQ, SNDCP receives an LL_XID_IND instead of the expected LL_XID_CNF. The LL_XID_IND primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.ESTABEX.A.014>)). Re-negotiation is not necessary because the just sent LL_XID_REQ contains no XID parameters, no compression fields in an XID parameter, no parameters in a compression field, that are not negotiated as part of the collision resolution <R.SNDCP.ESTABEX.A.021>.

(LLC 1)

SNDCP responds with an LL_XID_RES (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.SN_X_NEG.A.005>, <R.SNDCP.SN_X_NEG.A.008>). If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>. SNDCP from now on applies the parameter values it has included in its response (<R.SNDCP.SN_X_NEG.A.007>).

4.5.13 Collision between LL_XID_REQ and LL_XID_IND (XID, SNDCP decides), renegotiation necessary

```
SNDCP user                     SNDCP                       LLC
  |                              |                           |
  |                              |    LL_XID_IND             |
  |                              *<==========================*
  |                           (SNDCP 1)                      |
  |                              |      LL_XID_RES           |
  |                              *==========================>*
  |                              |                        (LLC 1)
  |                              |      LL_XID_REQ           |
  |                              *==========================>*
  |                              |                        (LLC 2)
  |                              |                           |
```

(SNDCP 1)
After having sent an LL_XID_REQ, SNDCP receives an LL_XID_IND instead of the expected LL_XID_CNF. The LL_XID_IND primitive contains a (possibly empty) SNDCP XID parameter (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.ESTABEX.A.014>)). Re-negotiation is necessary because the just sent LL_XID_REQ contains one or more XID parameters, or one or more compression fields in an XID parameter, or one or more parameters in a compression field, that are not negotiated as part of the collision resolution <R.SNDCP.ESTABEX.A.021>.

(LLC 1)
SNDCP responds with an LL_XID_RES (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.SN_X_NEG.A.005>, <R.SNDCP.SN_X_NEG.A.008>). If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>. SNDCP from now on applies the parameter values it has included in its response (<R.SNDCP.SN_X_NEG.A.007>).

(LLC 2)
By re-sending the just sent LL_XID_REQ negotiation of these XID parameters is performed at the earliest opportunity after conclusion of the collision resolution <R.SNDCP.ESTABEX.A.021>.

4.5.14 Collision between LL_XID_REQ and LL_XID_IND (XID, SNDCP asks SNDCP user for decision)

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.15 Responding after Collision between LL_XID_REQ and LL_XID_IND (SNDCP asked SNDCP user for decision), re-negotiation not necessary

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.16 Responding after Collision between LL_XID_REQ and LL_XID_IND (SNDCP asked SNDCP user for decision), re-negotiation necessary

Note: SN_XID_REQ, SN_XID_CNF, SN_XID_IND, SN_XID_RES are not used in our solution. Information about compressors will be passed with SNSM_ACTIVATE_IND. This MSC is deprecated.

4.5.17 Collision between LL_XID_REQ and LL_ESTABLISH_IND, re-negotiation not necessary

```
SNDCP user                 SNDCP                        LLC
  |                          |                           |
  |                          |   LL_ESTABLISH_IND        |
  |                          *<==========================*
  |                       (SNDCP 1)                      |
  |                          |     LL_ESTABLISH_RES      |
  |                          *==========================>*
  |                          |                        (LLC 1)
  |                          |                           |
```

(SNDCP 1)
After having sent an LL_XID_REQ SNDCP expects an LL_XID_CNF but receives an LL_ESTABLISH_IND primitive indicating establishment or re-establishment of acknowledged peer-to-peer operation for a SAPI in the LLC layer. Re-negotiation is not necessary because the just sent LL_XID_REQ contains no XID parameters, no compression fields in an XID parameter, no parameters in a compression field, that are not negotiated as part of the collision resolution <R.SNDCP.ESTABEX.A.021>.

(LLC 1)
SNDCP treats its LL_XID_REQ as not transmitted and responds with an LL_ESTABLISH_RES (<R.SNDCP.ESTPROC.A.003>, <R.SNDCP.ESTABEX.A.015>, <R.SNDCP.SN_X_NEG.A.005>, <R.SNDCP.SN_X_NEG.A.008>). SNDCP sends the proposed SNDCP XID parameters to the LLC SAP with the LL_ESTABLISH_REQ. If XID parameters are negotiated during a connection, the PCOMP value of an algorithm used both before an after the negotiation is not changed (<R.SNDCP.PCINEG.A.004>, <R.SNDCP.DCOMPNEG.A.004>). A PCOMP value used before a re-negotiation for one algorithm cannot be used for another algorithm after the re-negotiation (<R.SNDCP.PCINEG.A.005>, <R.SNDCP.DCOMPNEG.A.005>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>.

4.5.18 Receipt of LLC Reset

[13, 5.1.2.1]

```
SNDCP user                 SNDCP                        LLC
  |                          |                           |
  |                          |   LL_RESET_IND            |
  |                          *<==========================*
  |                       (SNDCP 1)                      |
  |                          |                           |
```

(SNDCP 1)
Upon receipt of the LL_RESET_IND <R.SNDCP.LLCRESETA.001> SNDCP shall reset all SNDCP XID parameters to their default values <R.SNDCP.LLCRESET.A.002>; reset the assignment of N-PDU number to start from 0 for every PDP context <R.SNDCP.LLCRESET.A.003> and treat all outstanding SNDCP LLC primitives as not sent <R.SNDCP.LLCRESET.A.004> Both N-PDU numbers are reset to 0. If peer-to-peer LLC operation mode is changed from unacknoqledged to acknowledged then both N-PDU numbers are reset to 0 <R.SNDCP.DT_ACK_G.A.007>.

4.5.19 Collision between LL_XID_REQ and LL_ESTABLISH_IND, re-negotiation necessary

[13, 6.2.2.4]

```
 SNDCP user                 SNDCP                       LLC
  |                           |                          |
  |                           |   LL_ESTABLISH_IND       |
  |                           *<=========================*
  |                        (SNDCP 1)                     |
  |                           |     LL_ESTABLISH_RES     |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |     LL_XID_REQ           |
  |                           *=========================>*
  |                           |                       (LLC 2)
  |                           |                          |
```

(SNDCP 1)
After having sent an LL_XID_REQ SNDCP expects an LL_XID_CNF but receives an LL_ESTABLISH_IND primitive indicating establishment or re-establishment of acknowledged peer-to-peer operation for a SAPI in the LLC layer. Re-negotiation is necessary because the just sent LL_XID_REQ contains one or more XID parameters, or one or more compression fields in an XID parameter, or one or more parameters in a compression field, that are not negotiated as part of the collision resolution <R.SNDCP.ESTABEX.A.021>.

(LLC 1)
SNDCP treats its LL_XID_REQ as not transmitted and responds with an LL_ESTABLISH_RES (<R.SNDCP.ESTPROC.A.003>, <R.SNDCP.ESTABEX.A.017>, <R.SNDCP.SN_X_NEG.A.005>, <R.SNDCP.SN_X_NEG.A.008>). Also to be considered: <R.SNDCP.DATACOMP.A.001>, <R.SNDCP.DATACOMP.A.002>, <R.SNDCP.DATACOMP.A.003>, <R.SNDCP.DATACOMP.A.004>, <R.SNDCP.DATACOMP.A.005>, <R.SNDCP.DATACOMP.A.006>, <R.SNDCP.VBISMAN. A.001>, <R.SNDCP.VBISMAN. A.002>.

(LLC 2)
SNDCP resends the LL_XID_REQ to complete the SNDCP XID negotiation (<R.SNDCP.ESTRELGN.A.004>, <R.SNDCP.ESTABEX.A.018>, <R.SNDCP.SN_X_NEG.A.002>).

4.6 Data transfer

Also to be considered: <R.SNDCP.REASSEMB.A.001>, <R.SNDCP.REASSEMB.A.006>, <R.SNDCP.SEGUNAC.A.005>,

<R.SNDCP.DT_ACK_G.A.001>, <R.SNDCP.DT_ACK_G.A.002>, <R.SNDCP.DT_ACK_G.A.003>, <R.SNDCP.DT_ACK_G.A.004>, <R.SNDCP.DT_ACK_G.A.008>, <R.SNDCP.DT_ACK_G.A.014>, <R.SNDCP.DT_ACK_G.A.015>, <R.SNDCP.DT_ACK_G.A.016> for acknowledged mode

4.6.1 Initiating acknowledged data transmission (incoming queue still receptive)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |    SN_DATA_REQ            |                          |
  *==========================>*                          |
  |                        (SNDCP 1)                     |
  |                           |    LL_DATA_REQ           |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |  SN_READY_IND             |                          |
  *<==========================*                          |
 (SNDCP user 1)               |                          |
  |                           |                          |
```

(SNDCP 1) <R.SNDCP.SERVICES.A.001>
Flow control: LLC has indicated its capability to receive the next SN-Data PDU at the requested SAPI with an LL_READY_IND. The SNDCP user sends an SN_DATA_REQ primitive to initiate acknowledged data transmission

(<R.SNDCP.DT_ACK.A.001>) containing a QoS parameter with "peak throughput" as the only field set <R.SNDCP.DT_ACK.M.017> and a radio priority parameter indicating the level of radio priority to be used by RLC/MAC <R.SNDCP.DT_ACK.M.018>, <R.SNDCP.DT_ACK_G.A.009>. Flow control: SNDCP has indicated its capability to receive the next N-PDU at the NSAPI identified in the SN_DATA_REQ with an SN_READY_IND. The PDP context for the NSAPI identified in the SN_DATA_REQ has been activated and acknowledged LLC operation has been established. All pending N-PDUs for this NSAPI have been sent (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DELIVSEQ.A.001>, <R.SNDCP.DELIVSEQ.A.001>). For the other NSAPIs there may be N-PDUs, that have not yet been sent, but for the SAPI in question there are no N-PDUs belonging to other NSAPIs (<R.SNDCP.DELIVSEQ.A.002>).

(LLC 1)
Flow control: The variable in SNDCP indicating LLC's capability to receive the next SN-Data PDU is set to FALSE. SNDCP assigns the current value of the Send N-PDU number as the N-PDU number of the N-PDU received <R.SNDCP.DT_ACK_G.A.010>. The Send N-PDU number is incremented by 1 <R.SNDCP.DT_ACK_G.A.011>. After having performed the necessary compression (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.DT_ACK.A.002>) and segmentation functions (<R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_ACK.A.003>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>) and stored the incoming N-PDU (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>, <R.SNDCP.DT_ACK.A.005>), SNDCP forwards the first SN-PDU in an LL_DATA_REQ primitive to the LLC layer (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DT_ACK.A.004>). The F bit in the SNDCP header is set to 1 for this first segment <R.SNDCP.SEGMENT.A.003>. DCOMP, PCOMP and N-PDU number shall be included <R.SNDCP.SEGMENT.A.005>. The segment number is set to 0 <R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>. If this first SN_PDU is also the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>, <R.SNDCP.SEGMENT.A.008>. Compression parameters are only included in this first SN-PDU. (<R.SNDCP.SERVICES.A.005>)].

(SNDCP user 1)
SNDCP sends an SN_READY_IND to the SNDCP user to indicate the possibility of sending another N-PDU to this SAPI.

4.6.2 Initiating acknowledged data transmission (incoming queue full)

```
SNDCP user                  SNDCP                       LLC
   |                          |                          |
   |    SN_DATA_REQ           |                          |
   *=========================>*                          |
   |                      (SNDCP 1)                      |
   |                          |    LL_DATA_REQ           |
   |                          *=========================>*
   |                          |                       (LLC 1)
   |                          |                          |
```

(SNDCP 1) <R.SNDCP.SERVICES.A.001>
Flow control: LLC has indicated its capability to receive the next SN-Data PDU at the requested SAPI with an LL_READY_IND. The SNDCP user sends an SN_DATA_REQ primitive to initiate acknowledged data transmission <R.SNDCP.DT_ACK.A.001>, <R.SNDCP.DT_ACK_G.A.009>. Flow control: SNDCP has indicated its capability to receive the next N-PDU at the NSAPI identified in the SN_DATA_REQ with an SN_READY_IND. The PDP context for the NSAPI identified in the SN_DATA_REQ has been activated and acknowledged LLC operation has been established. All pending N-PDUs for this NSAPI have been sent (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DELIVSEQ.A.001>, <R.SNDCP.DELIVSEQ.A.001>). For the other NSAPIs there may be N-PDUs, that have not yet been sent, but for the SAPI in question there are no N-PDUs belonging to other NSAPIs (<R.SNDCP.DELIVSEQ.A.002>).

(LLC 1)

Flow control: The variable in SNDCP indicating LLC's capability to receive the next SN-Data PDU is set to FALSE. SNDCP assigns the current value of the Send N-PDU number as the N-PDU number of the N-PDU received <R.SNDCP.DT_ACK_G.A.010>. The Send N-PDU number is incremented by 1 <R.SNDCP.DT_ACK_G.A.011>. After having performed the necessary compression (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.DT_ACK.A.002>) and segmentation functions (<R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_ACK.A.003>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>) and stored the incoming N-PDU (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>, <R.SNDCP.DT_ACK.A.005>), SNDCP forwards the first SN-PDU in an LL_DATA_REQ primitive to the LLC layer (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DT_ACK.A.004>). The F bit in the SNDCP header is set to 1 for this first segment <R.SNDCP.SEGMENT.A.003>. DCOMP, PCOMP and N-PDU number shall be included <R.SNDCP.SEGMENT.A.005>. The segment nuber is set to 0

<R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>. If this first SN_PDU is also the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>, <R.SNDCP.SEGMENT.A.008>. Compression parameters are only included in this first SN-PDU. (<R.SNDCP.SERVICES.A.005>)]

4.6.3 Not initiating acknowledged data transmission because of LLC not being ready (incoming queue still perceptive)

```
SNDCP user                   SNDCP                      LLC
  |                            |                          |
  |    SN_DATA_REQ             |                          |
  *===========================>*                          |
  |                        (SNDCP 1)                      |
  |                            |                          |
```

(SNDCP 1)
Flow control: SNDCP has indicated its capability to receive the next N-PDU at the NSAPI identified in the SN_DATA_REQ with an SN_READY_IND. The SNDCP user sends an SN_DATA_REQ primitive to initiate acknowledged data transmission. Flow control: LLC has not yet indicated its capability to receive the next SN-Data PDU at the requested SAPI with an LL_READY_IND or a once sent LL_READY_IND has been “used up” by sending an LL_DATA_REQ <R.SNDCP.DT_ACK_G.A.009>. The PDP context for the NSAPI identified in the SN_DATA_REQ has been activated and acknowledged LLC operation has been established. Flow control: SNDCP stores the received N-PDU (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>), it does neither compress, segment nor forward the N-PDU. If other N-PDUs are pending for the N-SAPI in question then the new N-PDU is stored AFTER the ones already buffered to manage the delivery sequence for the N-SAPI in question (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DELIVSEQ.A.001>). (<R.SNDCP.SERVICES.A.005>)] If there are other N-PDUs buffered for the SAPI in question then the order of incoming N-PDUs for this SAPI has to be kept.
SNDCP assigns the current value of the Send N-PDU number as the N-PDU number of the N-PDU received <R.SNDCP.DT_ACK_G.A.010>. The Send N-PDU number is incremented by 1 <R.SNDCP.DT_ACK_G.A.011>. The N-PDU is buffered. No SN_READY_IND is sent to the user.

4.6.4 Not initiating acknowledged data transmission because of LLC not being ready (incoming queue full)

```
SNDCP user                   SNDCP                      LLC
  |                            |                          |
  |    SN_DATA_REQ             |                          |
  *===========================>*                          |
  |                        (SNDCP 1)                      |
  |                            |                          |
```

(SNDCP 1)
Flow control: SNDCP has indicated its capability to receive the next N-PDU at the NSAPI identified in the SN_DATA_REQ with an SN_READY_IND. The SNDCP user sends an SN_DATA_REQ primitive to initiate acknowledged data transmission <R.SNDCP.DT_ACK_G.A.009>. Flow control: LLC has not yet indicated its capability to receive the next SN-Data PDU at the requested SAPI with an LL_READY_IND or a once sent LL_READY_IND has been “used up” by sending an LL_DATA_REQ. The PDP context for the NSAPI identified in the SN_DATA_REQ has been activated and acknowledged LLC operation has been established. Flow control: SNDCP stores the received N-PDU (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>), it does neither compress, segment nor forward the N-PDU. If other N-PDUs are pending for the N-SAPI in question then the new N-PDU is stored AFTER the ones already buffered to manage the delivery sequence for the N-SAPI in question (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.DELIVSEQ.A.001>). (<R.SNDCP.SERVICES.A.005>)] If there are other N-PDUs buffered for the SAPI in question then the order of incoming N-PDUs for this SAPI has to kept. No SN_READY_IND is sent to the SNDCP user. SNDCP assigns the current value of the Send N-PDU number as the N-PDU number of the N-PDU received <R.SNDCP.DT_ACK_G.A.010>. The Send N-PDU number is incremented by 1 <R.SNDCP.DT_ACK_G.A.011>.

4.6.5 Sending data in acknowledged mode after LL_READY_IND

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_READY_IND          |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           |    LL_DATA_REQ           |
  |                           *========================>*
  |                           |                        (LLC 1)
  |  SN_READY_IND             |                          |
  *<==========================*                          |
(GACI 1)                      |                          |
  |                           |                          |
```

(SNDCP 1) <R.SNDCP.SERVICES.A.001>
On the originating side SNDCP has buffered at least one N-PDU that has to be transmitted in acknowledged mode and not completely sent all SN-PDUS derived from segmenting this N-PDU. Flow control: LLC sends an LL_READY_IND to allow SNDCP another data request. (<R.SNDCP.SERVICES.A.005>)]

(LLC 1)
SNDCP sends an LL_DATA_REQ containing the next SN-PDU (“next” means that the segments of the N-PDUs are sent ordered after their place in the segmented N-PDU (<R.SNDCP.SERVICES.A.006>, <R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_ACK.A.004>). If this SN_PDU is the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>. If the sent SN-PDU is the first segment of the segmented N-PDU then compression parameters are included and the F bit in the SNDCP header is set to 1 <R.SNDCP.SEGMENT.A.003> and DCOMP, PCOMP and N-PDU number shall be included <R.SNDCP.SEGMENT.A.005>. If the sent SN-PDU is not the first segment of the segmented N-PDU then the F bit is set to 0 <R.SNDCP.SEGMENT.A.003> and DCOMP, PCOMP and N-PDU number shall not be included <R.SNDCP.SEGMENT.A.005>. The segment number is set to 0 in case of first segment, otherwise it is incremented, Modulo 16 is applied <R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>.

(GACI 1)
SNDCP is now ready to receive the first N-PDU from the user and indicates this to GACI.

4.6.6 Not sending data in acknowledged mode after LL_READY_IND

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_READY_IND          |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |  SN_READY_IND             |                          |
  *<==========================*                          |
(GACI 1)                      |                          |
  |                           |                          |
```

(SNDCP 1)
On the originating side SNDCP has already sent all SN-Data PDUs for the SAPI in question to LLC. Flow control: LLC sends an LL_READY_IND to allow SNDCP another data request. SNDCP stores the information that LLC is able to receive another LL_DATA_REQ. No primitive is sent. Flow control: No SN_READY_IND is sent.

4.6.7 Reacting to SNSM_SEQUENCE_IND

```
SNDCP user                  SNDCP                       SM
  |                            |                          |
  |                            |    LL_READY_IND          |
  |                            *<=========================*
  |                           (SNDCP 1)                   |
  |                            |   SNSM_SEQUENCE_RES      |
  |                            *=========================>*
  |                            |                        (SM 1)
  |                            |                          |
```

(SNDCP 1)
During an inter-SGSN routeing area update (an NSAPIs is using acknowledged peer-to-peer LLC operation) SM uses the SNSM_SEQUENCE_IND containing the Receive N-PDU number in the SGSN.

(SM 1)
SNDCP answers with an SNSM_SEQUENCE_RES to deliver the MS's Receive N-PDU number to the SGSN.

4.6.8 Receiving data in acknowledged mode, Receive First Segment state, F=0, M=0 or M=1, user ready or not, normal operation or recovery state

```
SNDCP user                  SNDCP                       LLC
  |                            |                          |
  |                            |    LL_DATA_IND           |
  |                            *<=========================*
  |                           (SNDCP 1)                   |
  |                            | LL_ESTABLISH_REQ         |
  |                            *=========================>*
  |                            |                        (LLC 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP is or is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 0 <R.SNDCP.SEGEX.A.004>, the M bit is set to 0 or to 1. The SNDCP user has or has not indicated his ability to receive another N-PDU for this NSAPI.

(LLC 1)
The delivered SN-PDU is discarded <R.SNDCP.SEGEX.A.005>. SNDCP sends an LL_ESTABLISH_REQ to LLC to re-establish acknowledged LLC operation <R.SNDCP.SEGEX.A.006> The LL_GETDATA_REQ is not sent to LLC because of re-establishment.

(LLC 2)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.9 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user ready, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  | SN_DATA_IND               |                          |
  *<==========================*                          |
 (SNDCP user 1)               |                          |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI.

(SNDCP user 1)
The received SN-PDU is the only segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU) with size <= N201-I. The Receive N-PDU number is incremented by 1 <R.SNDCP.DT_ACK.A.019>. The N-PDU is decompressed and sent to the SNDCP user with an SN_DATA_IND. DCOMP, PCOMP and N-PDU number are retrievd from the first segment <R.SNDCP.REASSEMB.A.001>. Now SNDCP is ready to receive the segments of another N-PDU and stays in Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.10 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user ready, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  | SN_DATA_IND               |                          |
  *<==========================*                          |
 (SNDCP user 1)               |                          |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is in recovery state <R.SNDCP.DT_ACK.A.020>. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the now complete N-PDU will be equal to the Receive N-PDU number.

(SNDCP user 1)
The received SN-PDU is the only segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU) with size <= N201-I. The N-PDU is decompressed and its N-PDU number is equal to the Receive N-PDU number <R.SNDCP.DT_ACK.A.021>. The Receive N-PDU number is incremented by 1 <R.SNDCP.DT_ACK.A.022>. The recovery state is exited. <R.SNDCP.DT_ACK.A.023>. Now normal operation is applied <R.SNDCP.DT_ACK.A.025>: The N-PDU is sent to the SNDCP user with an SN_DATA_IND. DCOMP, PCOMP and N-PDU number are retrievd from the first segment <R.SNDCP.REASSEMB.A.001>. Now SNDCP is ready to receive the segments of another N-PDU and stays in Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.11 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user ready, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                    SNDCP                      LLC
  |                             |                          |
  |                             |    LL_DATA_IND           |
  |                             *<=========================*
  |                            (SNDCP 1)                   |
  |                             | LL_GETDATA_REQ           |
  |                             *=========================>*
  |                             |                        (LLC 1)
  |                             |                          |
```

(SNDCP 1)
SNDCP is in recovery state <R.SNDCP.DT_ACK.A.020>. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the now complete N-PDU will be equal to the Receive N-PDU number.

(LLC 1)
The received SN-PDU is the only segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU) with size <= N201-I. The N-PDU is decompressed and its N-PDU number is not equal to the Receive N-PDU number <R.SNDCP.DT_ACK.A.021>. So the received N-PDU is discarded <R.SNDCP.DT_ACK.A.025>. Now SNDCP is ready to receive the segments of another N-PDU and stays in Receive First Segment state <R.SNDCP.REASSEMB.A.010>.SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.12 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user not ready, normal operation (not in recovery state)

```
SNDCP user                    SNDCP                      LLC
  |                             |                          |
  |                             |    LL_DATA_IND           |
  |                             *<=========================*
  |                            (SNDCP 1)                   |
  |                             |                          |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has not indicated his ability to receive another N-PDU for this NSAPI. Since the user is not ready, no further segment is requested from LLC.

4.6.13 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user not ready, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                      LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           |                          |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has not indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the now complete N-PDU will be equal to the Receive N-PDU number. Since the user is not ready, no further segment is requested from LLC.

4.6.14 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=0, user not ready, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                  SNDCP                      LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                         (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has not indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the now complete N-PDU will not be equal to the Receive N-PDU number.

(LLC 1)
The received SN-PDU is the only segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU) with size <= N201-I. The N-PDU is decompressed. The N-PDU number of the now complete N-PDU is not equal to the Receive N-PDU number. So the whole N-PDU is discarded <R.SNDCP.DT_ACK.025>. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.15 Receiving data in acknowledged mode, Receive First Segment state, F=1, M=1, user ready or not, normal operation (not in recovery state) or in recovery state

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is or is not in recovery state (this does not matter here because no N-PDU will be assembled and recovery state is only significant in the situation where a whole N-PDU is examined and then stored, sent or discarded). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 1. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI.

(LLC 1)
The received SN-PDU is not the only segment of an N-PDU (M bit set to 1 indicates that more segments are about to follow). SNDCP stores the received SN-PDU. Now SNDCP is ready to receive more segments of the affected N-PDU and enters Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>, <R.SNDCP.REASSEMB.A.012>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.16 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user ready, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  | SN_DATA_IND               |                          |
  *<==========================*                          |
 (SNDCP user 1)               |                          |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI.

(SNDCP user 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDUnumber are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed and sent to the SNDCP user with an SN_DATA_IND. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.17 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user ready, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  | SN_DATA_IND               |                           |
  *<==========================*                           |
 (SNDCP user 1)               |                           |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the N-PDU to be re-assembled and de-compressed will be equal to the Receive N-PDU number.

(SNDCP user 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDU number are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed. The Receive N-PDU number is incremented by 1 <R.SNDCP.DT_ACK.A.022>. Recovery state is exited <R.SNDCP.DT_ACK.A.023>. From now on normal operation will be applied <R.SNDCP.DT_ACK.A.024>: The N-PDU is sent to the SNDCP user with an SN_DATA_IND. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.18 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user ready, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.003>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the N-PDU to be re-assembled and de-compressed will not be equal to the Receive N-PDU number.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDU number are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed. Since the N-PDU number of the received N-PDU and the Receive N-PDU number are equal the N-PDU is discarded <R.SNDCP.DT_ACK.A.025>. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.19 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user not ready, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                        LLC
  |                            |                          |
  |                            |    LL_DATA_IND           |
  |                            *<=========================*
  |                           (SNDCP 1)                   |
  |                            | LL_GETDATA_REQ           |
  |                            *=========================>*
  |                            |                        (LLC 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed and stored, but not sent to the SNDCP user with an SN_DATA_IND because the user is not ready. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.20 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user not ready, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                        LLC
  |                            |                          |
  |                            |    LL_DATA_IND           |
  |                            *<=========================*
  |                           (SNDCP 1)                   |
  |                            | LL_GETDATA_REQ           |
  |                            *=========================>*
  |                            |                        (LLC 1)
  |                            |                          |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the received N-PDU will is equal to the Receive N-PDU number.

(LLC 1)

The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed. Since the N-PDU number of the decompressed N-PDU is equal to the Receive N-PDU number the Receive N-PDU number is incremented by 1

<R.SNDCP.DT_ACK.A.022>. The recovery state is exited <R.SNDCP.DT_ACK.A.023>. From now on normal operation is applied <R.SNDCP.DT_ACK.A.024>: The N-PDU is stored, but not sent to the SNDCP user with an SN_DATA_IND because the user is not ready. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.21 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=0, user not ready, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                        (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI. The N-PDU number of the received N-PDU is not equal to the Receive N-PDU number.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0 indicates the end of the N-PDU). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed. Since the N-PDU number of the decompressed N-PDU is not equal to the Receive N-PDU number the N-PDU number is discarded <R.SNDCP.DT_ACK.A.025>. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.22 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=0, M=1, user ready or not, normal operation (not in recovery state) or in recovery state

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                        (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is or is not in recovery state (this does not matter here because no N-PDU will be assembled and recovery state is only significant in the situation where a whole N-PDU is examined and then stored, sent or discarded). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 0, the M bit is set to 1. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not.

(LLC 1)
The received SN-PDU is not the last segment of an N-PDU (M bit set to 1). SNDCP stores the received SN-PDU and remains in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>, <R.SNDCP.REASSEMB.A.012>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.23 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user ready or not, DCOMP, PCOMP or N-PDU number different, normal operation (not in recovery state) or recovery state

```
SNDCP user                  SNDCP                       LLC
   |                          |                          |
   |                          |     LL_DATA_IND          |
   |                          *<=========================*
   |                         (SNDCP 1)                   |
   |                          | LL_ESTABLISH_REQ         |
   |                          *=========================>*
   |                          |                        (LLC 1)
   |                          |                          |
```

(SNDCP 1)
SNDCP is or is not in recovery state (This does not matter because the received SN-PDU will be discarded without affecting recovery / normal operation). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DCOMP, PCOMP and the N-PDU number are different from those in the first segment <R.SNDCP.SEGEX.A.011>.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The received SN-PDU and all other stored segments of the N-PDU in question are discarded <R.SNDCP.SEGEX.A.012>. SNDCP re-establishes acknowledged LLC operation by sending an LL_ESTABLISH_IND to LLC <R.SNDCP.SEGEX.A.013>.

(LLC 2)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.24 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user ready, DCOMP, PCOMP or N-PDU number not different, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                       LLC
   |                          |                          |
   |                          |     LL_DATA_IND          |
   |                          *<=========================*
   |                         (SNDCP 1)                   |
   | SN_DATA_IND              |                          |
   *<=========================*                          |
 (SNDCP user 1)               |                          |
   |                          | LL_GETDATA_REQ           |
   |                          *=========================>*
   |                          |                        (LLC 1)
   |                          |                          |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment <R.SNDCP.SEGMENT.A.006>.

(SNDCP user 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDU number are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed and sent to the SNDCP user with an SN_DATA_IND. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.25 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user ready, DCOMP, PCOMP or N-PDU number not different, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                            |                           |
  |                            |     LL_DATA_IND           |
  |                            *<==========================*
  |                           (SNDCP 1)                    |
  | SN_DATA_IND                |                           |
  *<===========================*                           |
 (SNDCP user 1)                |                           |
  |                            | LL_GETDATA_REQ            |
  |                            *==========================>*
  |                            |                         (LLC 1)
  |                            |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment <R.SNDCP.SEGMENT.A.006>. The N-PDU number of the rceived N-PDU is equal to the Receive N-PDU number.

(SNDCP user 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDU number are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed. The N-PDU number is equal to the Receive N-PDU number. The Receive N-PDU number is incremented by 1 <R.SNDCP.DT_ACK.A.002>. The recoverey state is exited <R.SNDCP.DT_ACK.A.023>. From now on normal operation will be applied <R.SNDCP.DT_ACK.A.024>: the N-PDU is sent to the SNDCP user with an SN_DATA_IND. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state.

(LLC 1)
SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.26 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user ready, DCOMP, PCOMP or N-PDU number not different, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                  SNDCP                        LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment <R.SNDCP.SEGMENT.A.006>. The N-PDU number of the received N-PDU is not equal to the Receive N-PDU number.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. DCOMP, PCOMP and N-PDU number are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. The N-PDU is decompressed. The N-PDU number is not equal to the Receive N-PDU number. The SN-PDU and all segments belonging to the N-PDU in question are discarded <R.SNDCP.DT_ACK.A.025>. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.27 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user not ready, DCOMP, PCOMP or N-PDU number not different, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                        LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed and stored, but not sent to the user because the user is not ready. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.28 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user not ready, DCOMP, PCOMP or N-PDU number not different, in recovery state, N-PDU number = Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment. The N-PDU number of the N-PDU after re-assembly and decompression will be equal to the receive N-PDU number.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed. The N-PDU number is equal to the Receive N-PDU number. The Receive N-PDU number is incremented by 1 <R.SNDCP.DT_ACK.A.002>. The recoverey state is exited <R.SNDCP.DT_ACK.A.023>. From now on normal operation will be applied <R.SNDCP.DT_ACK.A.024>: the N-PDU is stored, but not sent to the user because the user is not ready. Now SNDCP is ready to receive the segments of another N-PDU and enters Receive First Segment state. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.29 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=0, user not ready, DCOMP, PCOMP or N-PDU number not different, in recovery state, N-PDU number != Receive N-PDU number

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment. The N-PDU number of the N-PDU after re-assembly and decompression will not be equal to the receive N-PDU number.

(LLC 1)
The received SN-PDU is the last segment of an N-PDU (M bit set to 0). The stored segments for the now complete N-PDU are re-assembled. The N-PDU is decompressed. The N-PDU number is not equal to the Receive N-PDU number. The SN-PDU and all segments belonging to the N-PDU in question are discarded <R.SNDCP.DT_ACK.A.025>. Now SNDCP is

ready to receive the segments of another N-PDU and enters Receive First Segment state. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.30 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=1, user ready or not, DCOMP, PCOMP or N-PDU number different, normal operation (not in recovery state) or in recovery state

```
SNDCP user                  SNDCP                      LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_ESTABLISH_REQ         |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is or is not in recovery state (this does not matter here because no N-PDU will be assembled and recovery state is only significant in the situation where a whole N-PDU is examined and then stored, sent or discarded). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 0. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are different from those in the first segment <R.SNDCP.SEGEX.A.011>.

(LLC 1)
The received SN-PDU is not the last segment of an N-PDU (M bit set to 1). The received SN-PDU and all other stored segments of the N-PDU in question are discarded <R.SNDCP.SEGEX.A.012>. SNDCP re-establishes acknowledged LLC operation by sending an LL_ESTABLISH_IND to LLC <R.SNDCP.SEGEX.A.013>.

4.6.31 Receiving data in acknowledged mode, Receive Subsequent Segment state, F=1, M=1, user ready or not, DCOMP, PCOMP or N-PDU number not different, normal operation (not in recovery state) or in recovery state

```
SNDCP user                  SNDCP                      LLC
  |                           |                          |
  |                           |    LL_DATA_IND           |
  |                           *<=========================*
  |                          (SNDCP 1)                   |
  |                           | LL_GETDATA_REQ           |
  |                           *=========================>*
  |                           |                        (LLC 1)
  |                           |                          |
```

(SNDCP 1)
SNDCP is or is not in recovery state (this does not matter here because no N-PDU will be assembled and recovery state is only significant in the situation where a whole N-PDU is examined and then stored, sent or discarded). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. In the delivered SN-PDU the F bit is set to 1, the M bit is set to 1. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not. DOMP, PCOMP and the N-PDU number are not different from those in the first segment <R.SNDCP.SEGMENT.A.006>.

(LLC 1)
Because neither DCOMP, PCOMP nor N-PDU number is changed the SN-PDU is processed as normal. The received SN-PDU is not the last segment of an N-PDU (M bit set to 1) and is stored. Now SNDCP is ready to receive more segments of

the affected N-PDU and enters Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>, <R.SNDCP.REASSEMB.A.012>. SNDCP sends an LL_GETDATA_REQ to signal its capability to receive another SN-PDU.

4.6.32 Receiving data in acknowledged mode, Discard state, F=1 or F=0, M=1, user ready or not, normal operation (not in recovery state) or in recovery state

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is not in recovery state (this does not matter here because no N-PDU will be assembled and recovery state is only significant in the situation where a whole N-PDU is examined and then stored, sent or discarded). Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Discard state <R.SNDCP.REASSEMB.A.005>. In the delivered SN-PDU the F bit is set to 1 or to 0, the M bit is set to 1 <R.SNDCP.SEGEX.A.014>. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not.

(LLC 1)
The received SN-PDU is discarded <R.SNDCP.SEGEX.A.014> and SNDCP remains in the Discard state <R.SNDCP.SEGEX.A.016>.

4.6.33 Receiving data in acknowledged mode, Discard state, F=1 or F=0, M=0, user ready or not, normal operation (not in recovery state)

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is not in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Discard state <R.SNDCP.REASSEMB.A.005>. In the delivered SN-PDU the F bit is set to 1 or to 0, the M bit is set to 0 <R.SNDCP.SEGEX.A.017>. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not.

(LLC 1)
The received SN-PDU is discarded <R.SNDCP.SEGEX.A.018> and SNDCP enters Receive First Segment state <R.SNDCP.SEGEX.A.019>.

4.6.34 Receiving data in acknowledged mode, Discard state, F=1 or F=0, M=0, user ready or not, in recovery state

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_DATA_IND            |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           | LL_GETDATA_REQ            |
  |                           *==========================>*
  |                           |                         (LLC 1)
  |                           |                           |
```

(SNDCP 1)
SNDCP is in recovery state. Flow control: On the receiving side of an acknowledged data transmission SNDCP has indicated its capability to receive the next SN-Data PDU from the requested SAPI using an LL_GETDATA_REQ. LLC sends an LL_DATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_ACK.A.007>). SNDCP is in Discard state <R.SNDCP.REASSEMB.A.005>. In the delivered SN-PDU the F bit is set to 1 or to 0, the M bit is set to 0 <R.SNDCP.SEGEX.A.017>. The SNDCP user has indicated his ability to receive another N-PDU for this NSAPI or not.

(LLC 1)
The received SN-PDU is discarded <R.SNDCP.SEGEX.A.018> and SNDCP enters Receive First Segment state <R.SNDCP.SEGEX.A.019>.

4.6.35 Receiving data in unacknowledged mode, Receive First Segment state, F=0, M=0

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP <R.SNDCP.DT_UNACK.A.006>. The F bit in the SN-PDU is set to 0, although an SN-PDU with F bit set to 1 is expected <R.SNDCP.SEGEX.A.001>. The SN-PDU is the last segment of the N-PDU since the M bit is set to 0. SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. Reaction: the SN-PDU is discarded <R.SNDCP.SEGEX.A.002>. SNDCP remains in Receive First Segment state <R.SNDCP.SEGEX.A.003>.

4.6.36 Receiving data in unacknowledged mode, Receive First Segment state, F=0, M=1

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 0, although an SN-PDU with F bit set to 1 is expected <R.SNDCP.SEGEX.A.001>. The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1. SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>. Reaction: the SN-PDU is discarded <R.SNDCP.SEGEX.A.002>. SNDCP enters Discard state <R.SNDCP.SEGEX.A.003>.

4.6.37 Receiving data in unacknowledged mode, Receive First Segment state, F=1, M=0

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_UNITDATA_IND       |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |     SN_UNITDATA_IND       |                          |
  *<==========================*                          |
(SNDCP user 1)                |                          |
  |                           |                          |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The data is the first segment of an N-PDU (F bit set to 1). The SN-PDU is the last segment of the N-PDU since the M bit is set to 0. SNDCP is in Receive First Segment state <R.SNDCP.REASSEMB.A.003>.

(SNDCP user 1)
DCOMP and PCOMP are retrieved from the first segment <R.SNDCP.REASSEMB.A.001>. SNDCP reassembles the N-PDU from the list of segments (with one element here)(<R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_UNACK.A.008>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>), decompresses the information (<R.SNDCP.DT_UNACK.A.009>) and forwards it to the SNDCP user with the SN_UNITDATA_IND primitive (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.DT_UNACK.A.010>). The correct user is identified by the NSAPI field in the SN-PDU. (<R.SNDCP.DT_UNACK.A.007>). SNDCP remains in Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

4.6.38 Receiving data in unacknowledged mode, Receive First Segment state, F=1, M=1

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_UNITDATA_IND       |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |                           |                          |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The data is the first segment of an N-PDU (F bit set to 1). The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1. SNDCP is in Receive First Segments state <R.SNDCP.REASSEMB.A.003>. Reaction: the SN-PDU is stored. SNDCP enters Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>.

4.6.39 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=0, M=0

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_UNITDATA_IND       |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |     SN_UNITDATA_IND       |                          |
  *<==========================*                          |
(SNDCP user 1)                |                          |
  |                           |                          |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The data is a segment of an N-PDU for which some segments have

already been received (F bit set to 0). The SN-PDU is the last segment of the N-PDU since the M bit is set to 0. SNDCP is in Receive Subsequent Segments state <R.SNDCP.REASSEMB.A.004>.

(SNDCP user 1)
SNDCP reassembles the N-PDU from the list of segments (<R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_UNACK.A.008>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>), decompresses the information (<R.SNDCP.DT_UNACK.A.009>) and forwards it to the SNDCP user with the SN_UNITDATA_IND primitive (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.DT_UNACK.A.010>). DCOMP and PCOMP are retrievd from the first segment <R.SNDCP.REASSEMB.A.001>. The correct user is identified by the NSAPI field in the SN-PDU. (<R.SNDCP.DT_UNACK.A.007>). SNDCP shall detect lost SN-PDUs (<R.SNDCP.DT_UNACK.A.012>), discard duplicate SN-PDUs (<R.SNDCP.DT_UNACK.A.013>) and re-order out-of-sequence SN-PDUs, if possible (<R.SNDCP.DT_UNACK.A.014>). SNDCP enters Receive First Segment state <R.SNDCP.REASSEMB.A.010>.

4.6.40 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=0, M=1

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_UNITDATA_IND       |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |                           |                          |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 0. The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1. SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. Reaction: the SN-PDU is stored. SNDCP remains in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>, <R.SNDCP.REASSEMB.A.012>.

4.6.41 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=1, M=0, DCOMP and PCOMP unchanged

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_UNITDATA_IND       |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |     SN_UNITDATA_IND       |                          |
  *<==========================*                          |
(SNDCP user 1)                |                          |
  |                           |                          |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit is set to 1. The SN-PDU is the last segment of the N-PDU since the M bit is set to 0. SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. DCOMP and PCOMP have the same values as in the first segment <R.SNDCP.SEGMENT.A.006>.

(SNDCP user 1)
The F bit set to 1 is not expected since SNDCP is in Receive Subsequent Segment state. But as neither DCOMP nor PCOMP is different fom the values in the first segment the SN-PDU is processed as normal. SNDCP reassembles the N-PDU from the list of segments (<R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_UNACK.A.008>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>), decompresses the information (<R.SNDCP.DT_UNACK.A.009>) and forwards it to the SNDCP user with the SN_UNITDATA_IND primitive (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.DT_UNACK.A.010>). DCOMP and PCOMP are retrievd from the first segment <R.SNDCP.REASSEMB.A.001>. (<R.SNDCP.DT_UNACK.A.007>). SNDCP shall detect lost SN-PDUs (<R.SNDCP.DT_UNACK.A.012>), discard duplicate SN-PDUs (<R.SNDCP.DT_UNACK.A.013>) and re-order out-of-sequence SN-PDUs, if possible (<R.SNDCP.DT_UNACK.A.014) SNDCP enters Receive First Segment state.

4.6.42 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=1, M=0, PCOMP or DCOMP changed

```
SNDCP user                  SNDCP                      LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 1. The SN-PDU is the last segment of the N-PDU since the M bit is set to 0. SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. DCOMP or PCOMP are different from the values in the first segment <R.SNDCP.SEGEX.A.008>. Reaction: The received SN-PDU and all previous segments of the affected N-PDU are discarded <R.SNDCP.SEGEX.A.009>. SNDCP enters Receive First Segment state <R.SNDCP.SEGEX.A.010>.

4.6.43 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=1, M=1, PCOMP and DCOMP unchanged

```
SNDCP user                  SNDCP                      LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 1. The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1. SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.004>. DCOMP or PCOMP are not different from the values in the first segment <R.SNDCP.SEGMENT.A.006>. Reaction: The F bit set to 1 is not expected in this state, but since DCOMP and PCOMP are not different from the values in the first segmen t the SN-PDU is processed as normal. The received SN-PDU is stored. SNDCP remains in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.011>, <R.SNDCP.REASSEMB.A.012>.

4.6.44 Receiving data in unacknowledged mode, Receive Subsequent Segment state, F=1, M=1, PCOMP and DCOMP changed

```
SNDCP user                  SNDCP                      LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 1. The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1. SNDCP is in Receive Subsequent Segment state <R.SNDCP.REASSEMB.A.003>. DCOMP or PCOMP are different from the values in the first segment <R.SNDCP.SEGEX.A.008>. Reaction: The received SN-PDU and all previous segments of the affected N-PDU are discarded <R.SNDCP.SEGEX.A.009>. SNDCP enters Discard state <R.SNDCP.SEGEX.A.010>.

4.6.45 Receiving data in unacknowledged mode, Discard state, F=1 or F=0, M=0

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 1 or to 0. The SN-PDU is the last segment of the N-PDU since the M bit is set to 0 <R.SNDCP.SEGEX.A.017>. SNDCP is in Discard state <R.SNDCP.REASSEMB.A.005>. Reaction: The received SN-PDU is discarded <R.SNDCP.SEGEX.A.018>. SNDCP enters Receive First Segment state <R.SNDCP.SEGEX.A.019>.

4.6.46 Receiving data in unacknowledged mode, Discard state, F=1 or F=0, M=1

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITDATA_IND        |
  |                           *<==========================*
  |                         (SNDCP 1)                     |
  |                           |                           |
```

(SNDCP 1)
On the receiving side of an unacknowledged data transmission LLC sends an LL_UNITDATA_IND primitive to deliver the data to SNDCP (<R.SNDCP.DT_UNACK.A.006>). The F bit in the SN-PDU is set to 1 or to 0. The SN-PDU is not the last segment of the N-PDU since the M bit is set to 1 <R.SNDCP.SEGEX.A.014>. SNDCP is in Discard state <R.SNDCP.REASSEMB.A.005>. Reaction: The received SN-PDU is discarded <R.SNDCP.SEGEX.A.015>. SNDCP remains in Discard state <R.SNDCP.SEGEX.A.016>.

4.6.47 Receiving data in unacknowledged mode, timeout

deprecated

4.6.48 Receiving data in acknowledged mode (N-PDU complete, user ready)

deprecated

4.6.49 Receiving data in acknowledged mode (N-PDU complete, user not ready)

deprecated

4.6.50 Receiving data in acknowledged mode (N-PDU not complete)

deprecated

4.6.51 Receiving unacknowledged data in acknowledged mode

deprecated

4.6.52 Receiving a request from user for more data in acknowledged mode (no data available)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |   SN_GETDATA_REQ          |                          |
  *==========================>*                          |
  |                         (SNDCP 1)                    |
  |                           |                          |
```

(SNDCP 1)
The SNDCP user indicates the possibility to deliver another N-PDU to the specified NSAPI. There is no pending N-PDU for this NSAPI. SNDCP stores this information for the next N-PDU to be sent.

4.6.53 Receiving a request from user for more data in acknowledged mode (data available)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |   SN_GETDATA_REQ          |                          |
  *==========================>*                          |
  |                         (SNDCP 1)                    |
  |     SN_DATA_IND           |                          |
  *<==========================*                          |
(SNDCP user 1)                |                          |
  |                           |                          |
```

(SNDCP 1)
The SNDCP user indicates the possibility to deliver another N-PDU to the specified NSAPI. There is at least one pending N-PDU for this NSAPI.

(SNDCP user 1)
SNDCP forwards the N-PDU to the SNDCP user.

4.6.54 Receiving confirmation for data transmission in acknowledged mode (N-PDU not complete)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |                           |    LL_DATA_CNF           |
  |                           *<=========================*
  |                         (SNDCP 1)                    |
  |                           |                          |
```

(SNDCP 1)
Data has been transmitted in acknowledged mode and the LLC at the originating side sends an LL_DATA_CNF primitive to confirm the transmission.

4.6.55 Initiating unacknowledged data transmission (incoming queue still receptive)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |  SN_UNITDATA_REQ          |                          |
  *==========================>*                          |
  |                        (SNDCP 1)                     |
  |                           |    LL_UNITDATA_REQ       |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |  SN_UNITREADY_IND         |                          |
  *<==========================*                          |
 (SNDCP user 1)               |                          |
  |                           |                          |
```

(SNDCP 1)
Flow control: LLC has indicated its capability to receive the next SN-Unitdata PDU at the requested SAPI with an LL_UNITREADY_IND. The SNDCP user sends an SN_UNITDATA_REQ primitive to initiate unacknowledged data transmission (<R.SNDCP.DT_UNACK.A.001>). The PDP context for the NSAPI identified in the SN_UNITDATA_REQ. The incoming queue for this SAPI can take more than one more N-PDU.

(LLC 1) <R.SNDCP.SERVICES.A.002>
After having performed the necessary compression and segmentation functions (<R.SNDCP.SERVICES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_UNACK.A.002>, <R.SNDCP.DT_UNACK.A.003>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>), SNDCP forwards the first SN_PDU in an LL_UNITDATA_REQ primitive to the LLC layer (<R.SNDCP.DT_UNACK.A.004>). The F bit in the SNDCP header is set to 1 in this first segment <R.SNDCP.SEGMENT.A.003>. The segment number is set to 0 <R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>. DCOMP and PCOMP shall be included <R.SNDCP.SEGMENT.A.004>. Parameters included (in the MS) are: QoS parameters in the MS includes peak throughput and reliability class. Reliability class indicates whether the LLC frame carrying the SN-PDU shall be transmitted in protected mode, and whether RLC/MAC acknowledged or unacknowledged mode shall be used. Radio Priority is included only in the MS, and indicates the radio priority level to be used by RLC/MAC <R.SNDCP.DT_UNACK.M.017>. If this first SN_PDU is also the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>. (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>, <R.SNDCP.DT_UNACK.A.005>).

(SNDCP user 1)
SNDCP sends an SN_UNITREADY_IND to the user to indicate the possibility to sen d one more N.PDU.

4.6.56 Initiating unacknowledged data transmission (incoming queue full)

```
SNDCP user                  SNDCP                       LLC
  |                           |                          |
  |  SN_UNITDATA_REQ          |                          |
  *==========================>*                          |
  |                        (SNDCP 1)                     |
  |                           |    LL_UNITDATA_REQ       |
  |                           *=========================>*
  |                           |                       (LLC 1)
  |                           |                          |
```

(SNDCP 1)
Flow control: LLC has indicated its capability to receive the next SN-Unitdata PDU at the requested SAPI with an LL_UNITREADY_IND. The SNDCP user sends an SN_UNITDATA_REQ primitive to initiate unacknowledged data transmission (<R.SNDCP.DT_UNACK.A.001>). The PDP context for the NSAPI identified in the SN_UNITDATA_REQ. In the incoming queue for this NSAPI there is only place for one more N-PDU.

(LLC 1) <R.SNDCP.SERVICES.A.002>
After having performed the necessary compression and segmentation functions (<R.SNDCP.SERVIC ES.A.007>, <R.SNDCP.SERVICES.A.008>, <R.SNDCP.SERVICES.A.009>, <R.SNDCP.DT_UNACK.A.002>, <R.SNDCP.DT_UNACK.A.003>, <R.SNDCP.SEGREAS.A.001>, <R.SNDCP.SEGMENT.A.001>, <R.SNDCP.SEGMENT.A.002>), SNDCP forwards the first SN_PDU in an LL_UNITDATA_REQ primitive to the LLC layer (<R.SNDCP.DT_UNACK.A.004>). The F bit in the SNDCP header is set to 1 in this first segment <R.SNDCP.SEGMENT.A.003>. The segment number is set to 0 <R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>. DCOMP and PCOMP shall be included <R.SNDCP.SEGMENT.A.004>. Parameters included are: QoS parameters in the MS includes peak throughput and reliability class. Reliability class indicates whether the LLC frame carrying the SN-PDU shall be transmitted in protected mode, and whether RLC/MAC acknowledged or unacknowledged mode shall be used. Radio Priority is included only in the MS, and indicates the radio priority level to be used by RLC/MAC. If this first SN_PDU is also the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>. (<R.SNDCP.NPDUBUF.A.001>, <R.SNDCP.NPDUBUF.A.003>, <R.SNDCP.DT_UNACK.A.005>).

4.6.57 Not initiating unacknowledged data transmission because of LLC not being ready (incoming queue still receptive)

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |    SN_UNITDATA_REQ        |                           |
  *==========================>*                           |
  |                        (SNDCP 1)                      |
  |  SN_UNITREADY_IND         |                           |
  *<==========================*                           |
 (SNDCP user 1)               |                           |
  |                           |                           |
```

(SNDCP 1)
Flow control: LLC has not yet indicated its capability to receive the next SN-Unitdata PDU at the requested SAPI with an LL_UNITREADY_IND or a once sent LL_UNITREADY_IND has been “used up” by sending an LL_UNITDATA_REQ. The SNDCP user sends an SN_UNITDATA_REQ primitive to initiate acknowledged data transmission. SNDCP stores the received N-PDU (<R.SNDCP.NPDUBUF.A.001>,), it does neither compress, segment nor forward the N-PDU. In the incoming queue for this NSAPI there is place for more than this one N-PDU.

(SNDCP user 1)
SNDCP sends an SN_UNITREADY_IND to the SNDCP user to indicate the possibility to send one more N-PDU.

4.6.58 Not initiating unacknowledged data transmission because of LLC not being ready (incoming queue full)

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |    SN_UNITDATA_REQ        |                           |
  *==========================>*                           |
  |                        (SNDCP 1)                      |
  |                           |                           |
```

(SNDCP 1)
Flow control: LLC has not yet indicated its capability to receive the next SN-Unitdata PDU at the requested SAPI with an LL_UNITREADY_IND or a once sent LL_UNITREADY_IND has been “used up” by sending an LL_UNITDATA_REQ. The SNDCP user sends an SN_UNITDATA_REQ primitive to initiate acknowledged data transmission. SNDCP stores the received N-PDU (<R.SNDCP.NPDUBUF.A.001>,), it does neither compress, segment nor forward the N-PDU. In the incoming queue for this NSAPI there is only place for this one N-PDU.

4.6.59 Sending data in unacknowledged mode after LL_UNITREADY_IND

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITREADY_IND       |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           |    LL_UNITDATA_REQ        |
  |                           *==========================>*
  |                           |                          (LLC 1)
  |                           |                           |
```

(SNDCP 1)
On the originating side SNDCP has buffered an N-PDU that has to be transmitted in unacknowledged mode and not completely sent all SN-Unitdata PDUs derived from segmenting this N-PDU. (Alternatively there are whole N-PDUs in a waiting queue for the affected SAPI). Flow control: LLC sends an LL_UNITREADY_IND to allow SNDCP another unitdata request for the SAPI used for the transmission of the above N-PDU.

(LLC 1) <R.SNDCP.SERVICES.A.002>
SNDCP sends an LL_UNITDATA_REQ containing the next SN_PDU (<R.SNDCP.DT_UNACK.A.004>). If this SN_PDU is the last segment derived from the incoming N-PDU then the M bit is set to 0, otherwise it is set to 1 <R.SNDCP.SEGMENT.A.007>. If the segment sent is the first segment of the N-PDU then the F bit is set to 1 and DCOMP and PCOMP shall be included, otherwise the F bit shall be set to 0 and no DCOMP or PCOMP shall be included. The segment number is set to 0 in case of first segment, otherwise it is incremented with Modulo 16 operation applied <R.SNDCP.SEGUNAC.A.001>, <R.SNDCP.SEGUNAC.A.002>.

4.6.60 Not sending data in unacknowledged mode after LL_UNITREADY_IND

```
SNDCP user                  SNDCP                       LLC
  |                           |                           |
  |                           |    LL_UNITREADY_IND       |
  |                           *<==========================*
  |                          (SNDCP 1)                    |
  |                           |                           |
```

(SNDCP 1)
On the originating side SNDCP has already sent all SN-Unitdata PDUs for the SAPI indicated in the following LL_UNITREADY_IND to LLC. Flow control: LLC sends an LL_UNITREADY_IND to allow SNDCP another unitdata request for the SAPI in question. SNDCP stores the information that LLC is able to receive another LL_UNITDATA_REQ. No primitive is sent.

4.6.61 Receiving data in unacknowledged mode, first segment of N-PDU, not last segment, no timeout

deprecated

4.6.62 Receiving data in unacknowledged mode, not first segment of N-PDU, list not complete, no timeout

deprecated

4.6.63 Receiving data in unacknowledged mode, not first segment of N-PDU, last segment, no timeout, no segment missing

deprecated (Provide a short description of this document.)

Appendices

A. Acronyms

DS-WCDMA	Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)

Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>