3GPP TSG-CN Meeting #25 **NP-040374**
8th – 10th September 2004. Palm Springs, USA.

Source: **TSG CN WG1**

Title: **CR to Rel-4 WI TEI4 (with mirror CRs) for TS 44.065**

Agenda item: **7.11**

Document for: **APPROVAL**

This document contains **3 CRs on Rel-4 Work Item "TEI4"**, that have been agreed by TSG CN WG1 CN#35 meeting and forwarded to TSG CN Plenary meeting #25 for approval.

TDoc #	Tdoc Title	Spec	CR #	Rev	CAT	C_Version	WI	Rel
N1-041634	Negotiation of compression entities with unknown algorithm type	44.065	15	2	F	4.2.0	TEI4	Rel-4
N1-041635	Negotiation of compression entities with unknown algorithm type	44.065	16	2	A	5.1.0	TEI4	Rel-5
N1-041636	Negotiation of compression entities with unknown algorithm type	44.065	17	2	A	6.2.0	TEI4	Rel-6

3GPP TSG-CN1 Meeting #35
Sophia Antipolis, France, 16-20 August 2004

Tdoc N1-041634
(rev of Tdoc N1-041616)

CR-Form-v7.1

CHANGE REQUEST

⌘ **44.065** **CR** **015** ⌘ **rev** **2** ⌘ Current version: **4.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Negotiation of compression entities with unknown algorithm type
Source:	⌘	Siemens AG
Work item code:	⌘	TEI4
Date:	⌘	06.08.2004
Category:	⌘	**F**
Release:	⌘	Rel-4

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change: ⌘ Ambiguity in the specification of the negotiation of compression entities:

According to subclauses 6.8.1 and 6.8.2:

> "If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity."

and

> "If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator."

I.e. an SNDCP entity can accept a compression entity proposed by its peer SNDCP entity by not including it in the responding SNDCP XID block.

On the other hand, according to subclause 6.8.3, exception handling:

> "In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.
> If the originating SNDCP XID block includes a parameter with **unrecognised Type field**, the parameter shall be ignored by the responder."

Some manufacturers have interpreted this requirement in such a way that it applies also to the parameter "algorithm type" in the SNDCP XID parameters "protocol control information compression" and "data compression".

	With such an interpretation, however, in some scenarios the SNDCP entity originating the XID negotiation cannot decide whether the peer SNDCP entity did not respond to a proposed compression entity in order to accept it or because it did not know the algorithm and therefore ignored the proposed compression entity. Example: Some R97/98 MS implementations ignore a proposal for a protocol control information compression with algorithm type RFC2507, because this algorithm was only introduced in R99. Since the MS does not explicitly reject the proposed compression entity for RFC2507, a R99 SGSN following subclauses 6.8.1 and 6.8.2 will assume that the MS has accepted the proposal and will start using this compression entity. The MS will then discard the downlink packets compressed by the SGSN with RFC2507. For the algorithm RFC2507 which was introduced in R99, the SNDCP entity originating the XID negotiation can resolve this ambiguity only by evaluating the revision level of its peer entity (revision level in the MS network capability IE or SGSNR bit in the BCCH or PBCCH, respecitvely). In order to remove this ambiguity for algorithms introduced in Rel-4 or later, the following is proposed:
Summary of change: ⌘	1) It is clarified that in the "protocol control information compression" field and "data compression" field the two octets following the PCOMP or DCOMP values always contain the "Applicable NSAPIs" parameter. 2) The parameter "algorithm type" is renamed to "algorithm identifier" so that 6.8.3 is no longer applicable to unknown compression algorithms. 3) It is clarified that if an SNDCP entity in the **MS** receives a proposal for a compression entity containing an unknown algorithm, it **should** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0. From Rel-5 onwards, with SNDCP version 1 this behaviour is mandatory. If an SNDCP entity in the **SGSN** receives a proposal for a compression entity containing an unknown algorithm, it **shall** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0, from Rel-4 onwards . 4) From Rel-5 onwards, the SDNCP version number is raised to 1 so that the entity originating the XID negotiation can determine whether its peer entity is a R99 implementation or a 'Rel-4 or later' implementation. 5) If an SNDCP entity in Rel-4 wants to accept a proposal for a data compression entity with algorithm V.44, it shall include in the responding SNDCP XID block at least the corresponding compression entity number. The length values in the compression parameter table for V.44 are corrected. 6) The peer entity of the XID negotiation shall accept only one header compression algorithm and one data compression algorithm for an NSAPI.
Consequences if not approved: ⌘	It will not be possible to use the compression algorithms introduced after R99 (V.44 and ROHC), since it will not be possible for the entity originating the XID negotiation to determine whether its peer entity is a legacy R99 implementation or a 'Rel-4 or later' implementation. Consequently the entity originating the XID negotiation will not be able to decide whether the peer SNDCP entity did not respond to a proposed compression entity for V.44 or ROHC in order to accept it or because the peer entity did not know the algorithm and therefore ignored the proposed compression entity.

Clauses affected:	⌘	6.5.1, 6.5.1.1, 6.5.1.1.4, 6.5.2.1, 6.5.3.1, 6.5.4.1, 6.6, 6.6.1, 6.6.1.1, 6.6.1.1.4, 6.6.2.1, 6.6.3.1, 6.8.1, 6.8.3, 6.10			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.5 Protocol control information compression

Protocol control information compression is an optional SNDCP feature.

Negotiation of the supported algorithms and their parameters is carried out between MS and SGSN using the SNDCP XID parameters (see clause 8).

6.5.1 Negotiation of multiple protocol control information compression types

Each SNDCP entity that supports protocol control information compression shall be able to negotiate one or several protocol control information compression entities with the compression field format shown in figure 7. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one protocol control information compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

6.5.1.1 Format of the protocol control information compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier~~type~~				
Octet 3	Length=n-3							
Octet 4	PCOMP1				PCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 7: Protocol control information compression field format for SNDCP XID negotiation

6.5.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.5.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of PCOMP values needed by the corresponding compression algorithm (e.g. PCOMP3 and PCOMP4 shall not be included if the number of PCOMP values needed by a compression algorithm is one or two). If an odd number of PCOMP values are used by a compression algorithm, then the last PCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.5.1.1.3 Entity number

The entity number shall be used to identify a protocol control information compression entity on a SAPI. The entity number shall be assigned using the following rules:

- The entity number shall be an integer from 0 to 31.
- The entity number shall be assigned independently on each of the SAPIs.
- An entity number shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, an unassigned entity number shall become selected. If there is no unassigned entity number left, the compression entity shall not be proposed.
- A selected entity number shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned entity number shall become unassigned when the corresponding compression entity is deleted as a result of an XID negotiation, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4) in which an entity number is currently selected:
  - If the selected entity number is included with the P bit set to 0 in the incoming SNDCP XID block, then it shall be assumed that the peer SNDCP entity agreed to the creation of the proposed entity but the response was lost. Therefore the selected entity number shall become assigned, any selected PCOMP and DCOMP values for the algorithm of the entity shall become assigned, and the compression entity shall be created, before the incoming SNDCP XID block is processed. After the incoming SNDCP XID block is processed, the compression entity shall be negotiated again if necessary, as defined in subclause 6.2.1.4.
  - Otherwise (i.e. if the selected entity number is not included, or is included with the P bit set to 1 in the incoming SNDCP XID block), the selected entity number shall become unassigned, and any selected PCOMP and DCOMP values for the algorithm of the entity shall become unassigned, before the incoming SNDCP XID block, if any, is processed. Following the collision resolution procedure, the originally-proposed compression entity shall be proposed again (i.e. the originally-proposed compression entity shall not be considered created even if the originally-selected entity number is proposed in the incoming SNDCP XID block) by sending the appropriate primitive (LL-ESTABLISH.request or LL-XID.request). The originally-selected entity number, PCOMP and DCOMP values shall be used for the compression entity being re-proposed if they are unassigned, otherwise a new entity number, PCOMP or DCOMP value shall be selected.
- In the case of a collision in which an entity number is currently assigned:
  - If the peer SNDCP entity proposes a new compression entity with the same entity number, then it shall be assumed that the peer SNDCP entity negotiated the deletion of the entity but the response was lost, and the entity number is being reused. Therefore the original compression entity shall be deleted, the entity number shall become unassigned, PCOMP and DCOMP values shall be unassigned if necessary (see subclause 6.5.1.1.5), and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned entity number is not included, or is included with the P bit set to 0 in the incoming SNDCP XID block), the usual rules regarding collision handling shall apply.
- In the case of a collision in which a PCOMP or DCOMP value is currently assigned to a compression algorithm:
  - If the peer SNDCP entity proposes a new compression entity with the same PCOMP or DCOMP assigned to a different algorithm, then it shall be assumed that the peer SNDCP entity negotiated the deletion of all entities using the algorithm to which the PCOMP or DCOMP value was assigned, but the response was lost, and the PCOMP or DCOMP value is being reused. Therefore, all compression entities using that algorithm shall be deleted, all corresponding entity numbers shall become unassigned, and all PCOMP or DCOMP values assigned to the algorithm shall become unassigned, and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned PCOMP or DCOMP is not included, or is included and assigned to the same algorithm), the usual rules regarding collision handling shall apply.

6.5.1.1.4 Algorithm identifiertype

Table 4 show the list of protocol control information compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 4 shall be updated.

Table 4: List of protocol control information compression algorithms supported by SNDCP

Compression algorithm	Algorithm identifiertype (Range 0 to 31)
RFC 1144	0
RFC 2507	1
-	Other values Reserved

6.5.1.1.5 PCOMP

One or more PCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for protocol control information compression. Each of the assigned PCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the PCOMP values follows the following general rules:

- PCOMP shall be an integer from 0 to 15.
- PCOMP value 0 is reserved permanently for no compression.
- PCOMP shall be assigned independently on each of the SAPIs.
- An assigned PCOMP value applies to all NSAPIs mapped to the same SAPI.
- PCOMP values shall be assigned to compression algorithms, not to compression entities (i.e. the same PCOMP value(s) shall be used by different compression entities on the same SAPI using the same compression algorithm).
- A PCOMP value shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, and if PCOMP values have not yet been assigned to the corresponding compression algorithm, then the appropriate number of unassigned PCOMP values shall be selected. If there is not enough unassigned PCOMP values left, the compression entity shall not be proposed.
- A selected PCOMP value shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned PCOMP value shall become unassigned when the corresponding compression algorithm is no longer in use by any compression entity, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4), the handling of PCOMP values shall be in accordance with subclause 6.5.1.1.3.

While transferring data, the compressed frame type for an N-PDU is conveyed in the PCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.5.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of protocol control information compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the protocol control information compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.5.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.5.2 TCP/IP header compression (RFC 1144)

The protocol control information compression method is specific for each network layer protocol type. TCP/IP (IPv4) header compression is specified in RFC 1144 [9].

6.5.2.1 Parameters

Table 5 contains the parameters defined for a compression entity using TCP/IP header compression. They may be negotiated during SNDCP XID negotiation.

Table 5: RFC 1144 TCP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 1144	0	0, 2 or 3 if P bit is 0, 1, 3 or 4 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			S_0 - 1	bbbbbbbb	0 through 255	down	15

6.5.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.2.1.2 S_0

The number of state slots, as defined in [9]. The S_0 range is 1 through 256, with 16 as default value.

6.5.2.2 Assignment of PCOMP values

The underlying service shall be able to distinguish the three types of compressed N-PDUs (i.e. Type IP, Uncompressed TCP, and Compressed TCP), as defined in RFC 1144 [9]. These three N-PDU types are differentiated by using different PCOMP values.

Two PCOMP values shall be assigned to the TCP/IP header compression algorithm. PCOMP1 shall contain the PCOMP value for the frame type "Uncompressed TCP", and PCOMP2 shall contain the PCOMP value for the frame type "Compressed TCP".

The PCOMP value of 0 shall be used for the frame type "Type IP".

6.5.2.3 Error Recovery

When TCP/IP header compression is used with unacknowledged peer-to-peer LLC operation, the decompression entity shall be notified in case an N-PDU is dropped, so that error recovery procedure (see RFC 1144 [9]) can be invoked.

6.5.3 TCP/IP and UDP/IP header compression (RFC 2507)

Detailed operation of the RFC 2507 header compression for IPv4 and IPv6 is described in clause 3 of the IETF specification RFC 2507 [10].

6.5.3.1 Parameters

Table 6 contains the parameters defined for a compression entity using RFC 2507 header compression. They may be negotiated during SNDCP XID negotiation.

Table 6: RFC 2507 TCP/IP and UDP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier ~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 2507	1	0, 2, 4, 5, 6, 7 or 9 if P bit is 0, 3, 5, 7, 8, 9, 10 or 12 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			F_MAX_PERIOD	bbbbbbbb bbbbbbbb	1-65535	down	256
			F_MAX_TIME	bbbbbbbb	1-255	down	5
			MAX_HEADER	bbbbbbbb	60-255	down	168
			TCP_SPACE	bbbbbbbb	3-255	down	15
			NON_TCP_SPACE	bbbbbbbb bbbbbbbb	3-65535	down	15

The explanation of the individual parameters can be found in the clause 14 of the IETF specification RFC 2507 [10].

6.5.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.3.2 Assignment of PCOMP values for RFC 2507

The following PCOMP values shall be assigned to the RFC 2507 header compression. The PCOMP value 0 shall be used for regular IPv4 and IPv6 packets.

Table 7: PCOMP values assigned to RFC 2507 header compression algorithm

PID value	Packet type
PCOMP1	Full header
PCOMP2	Compressed TCP
PCOMP3	Compressed TCP non-delta
PCOMP4	Compressed non-TCP
PCOMP5	Context state

6.5.3.3 Error Recovery

The mechanisms related to error recovery and packet reordering are described in clauses 10 and 11 of the RFC 2507[10].

6.6 Data compression

Data compression is an optional SNDCP feature. Data compression applies to both SN-DATA and SN-UNITDATA primitives.

Data compression, if used, shall be performed on the entire N-PDU, including the possibly compressed protocol control information.

Figure 8 shows an example how the SNDCP functions may be used. Several NSAPIs may use a common data compression entity, i.e. the same compression algorithm and the same dictionary. Separate data compression entities shall be used for acknowledged (SN-DATA) and unacknowledged (SN-UNITDATA) data transfer. Several NSAPIs may be associated with one SAPI, i.e. they may use the same QoS profile.

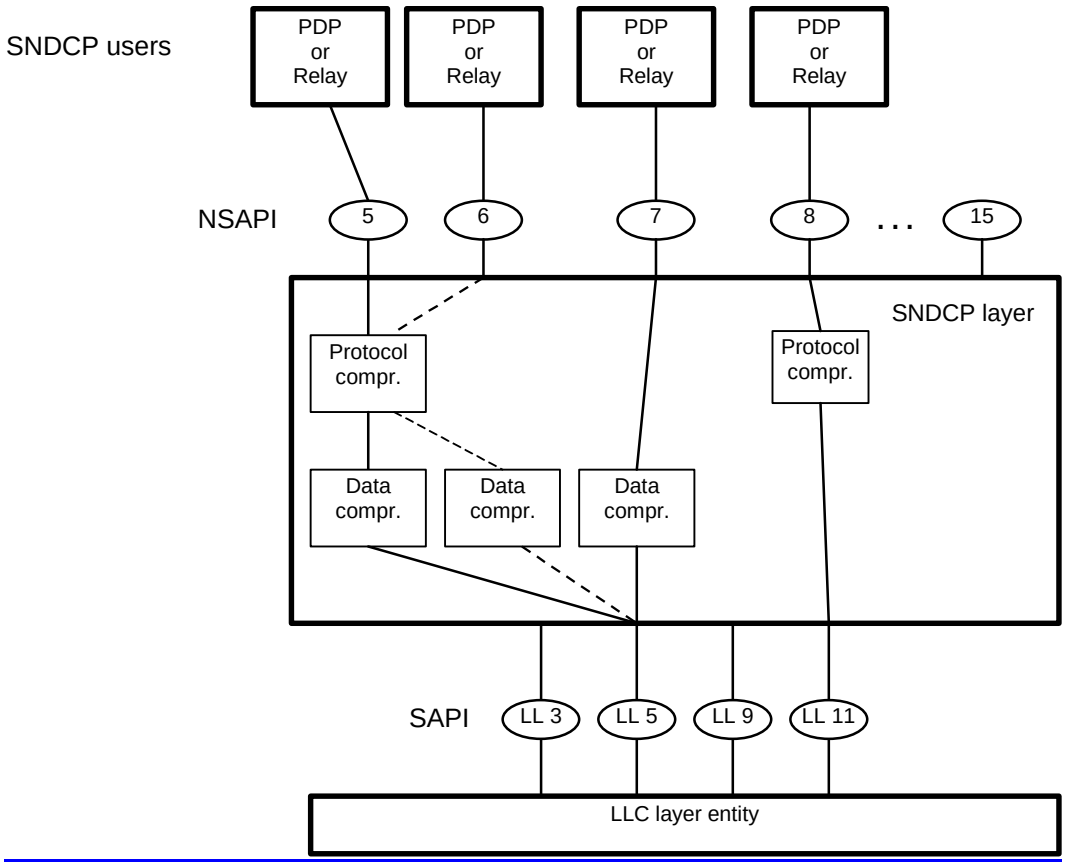

Figure 8: An example for the usage of NSAPIs, SNDCP functions, and SAPIs

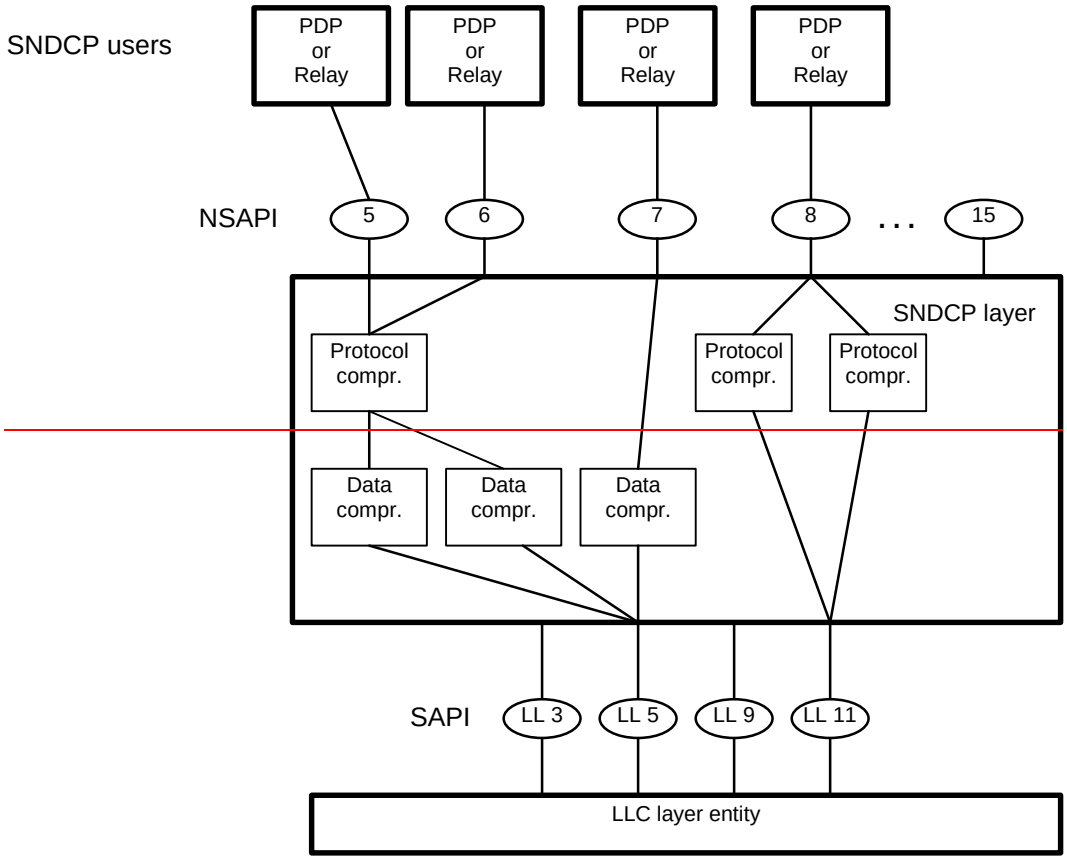

Figure 8: An example for the usage of NSAPIs, SNDCP functions, and SAPIs

6.6.1 Negotiation of multiple data compression types

Each SNDCP entity that supports data compression shall be able to negotiate one or several data compression entities with the compression field format shown in figure 9. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one data compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

For each NSAPI one or more data compression are chosen. This choice is also indicated in the SNDCP XID. Only NSAPIs that are using the same SAPI may use the same data compression entity. If more than one compression entity is chosen for an NSAPI, these entities must use different data compression algorithms. However, only one data compression entity is used for one N-PDU; i.e. the used data compression entity may be changed from N-PDU to N-PDU.

6.6.1.1 Format of the data compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier~~type~~				
Octet 3	Length=n-3							
Octet 4	DCOMP1				DCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 9: Data compression field format for SNDCP XID negotiation

6.6.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.6.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of DCOMP values needed by the corresponding compression algorithm (e.g. DCOMP3 and DCOMP4 shall not be included if the number of DCOMP values needed by a compression algorithm is one or two). If an odd number of DCOMP values are used by a compression algorithm, then the last DCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.6.1.1.3 Entity number

The entity number shall be used to identify a data compression entity on a SAPI. See subclause 6.5.1.1.3 for the rules for assigning entity numbers. The assignment of entity numbers for protocol control information compression entities and data compression entities shall be independent.

6.6.1.1.4 Algorithm identifier~~type~~

Table 6a shows the list of data compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 6a shall be updated.

Table 6a: List of data compression algorithms supported by SNDCP

Data compression algorithm	Algorithm identifier~~type~~ (Range 0 to 31)
V.42 bis	0
V.44	1
-	Other values Reserved

6.6.1.1.5 DCOMP

One or more DCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for data compression. Each of the assigned DCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the DCOMP values shall follow the rules for the assignment of PCOMP values in subclause 6.5.1.1.5.

While transferring data, the compressed frame type for an N-PDU is conveyed in the DCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.6.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of data compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the data compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.6.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.6.2 Management of V.42 bis data compression

ITU-T Recommendation V.42 bis [8] data compression may be used with SN-DATA primitives and SN-UNITDATA primitives.

6.6.2.1 Parameters

Table 7a contains the parameters defined for a compression entity using ITU-T Recommendation V.42 bis data compression. They may be negotiated during SNDCP XID negotiation.

Table 7a: V.42 bis data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.42 bis	0	0, 2, 3, 5, or 6 if P bit is 0, 1, 3, 4, 6, or 7 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_1	bbbbbbbb bbbbbbbb	512 through 65535	down	2048
			P_2	bbbbbbbb	6 through 250	down	20

6.6.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.2.1.2 P_0

Two bits are used to indicate the usage of compression, one bit for each direction.

00 compress neither direction.

01 compress MS-to-SGSN direction only.

10 compress SGSN-to-MS direction only.

11 compress both directions.

6.6.2.1.3 P_1

Maximum number of codewords in the compressor dictionary (see [8]).

6.6.2.1.4 P_2

Maximum number of characters in an uncompressed data string that is accepted to be encoded.

6.6.2.2 Assignment of DCOMP values

One DCOMP value shall be assigned (as DCOMP1) to the V.42 bis data compression algorithm.

6.6.2.3 Operation of V.42 bis data compression

When V.42 bis is used with SN-DATA primitives, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent.

When V.42 bis is used with SN-UNITDATA primitives, the compression entity shall be reset (using the C-INIT primitive defined in [8]) before an N-PDU is compressed or decompressed. After compression, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent. The LLC protocol shall operate in the protected mode of operation.

When V.42 bis is used with SN-DATA primitives and an error is detected by the decoder, the SNDCP entity shall use LL-ESTABLISH.request primitive to reset the acknowledged peer-to-peer LLC operation for the SAPI used.

6.6.3 Management of V.44 data compression

ITU-T Recommendation V.44 data compression, as described in [11], may be used with SN-DATA primitives and SN-UNITDATA primitives. Annex B of ITU-T Recommendation V.44 describes two methods of implementation and operation of V.44 in packet networks: Packet Method and Multi-Packet Method. Multi-Packet Method is a superset of Packet Method and an MS or SGSN that supports Multi-Packet Method must also support Packet Method.

6.6.3.1 Parameters

Table 7c contains the parameters defined for a compression entity using V.44 data compression. They may be negotiated during SNDCP XID negotiation. During V.44 data compression negotiation, unless both the MS and SGSN support Multi-Packet Method, Packet Method is used. Parameter C_0 indicates support of Packet Method (10000000) or both methods (11000000).

NOTE 1: V.44 data compression negotiation is not required. If V.44 is selected and no compression parameters are specified, then Packet Method with defaults as defined in subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2, is used.

Table 7c: V.44 data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier ~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.44	1	0, 2, 3, 4, 6, 8, 10, or 12 if P bit is 0, 1, 3, 4, 5, 7, 9, 11, or 13 if P bit is 1.~~5 or 6~~	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			C_0	bb000000	10000000 or 11000000	11000000 down to 10000000	10000000
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_{1T}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.4
			P_{1R}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.5
			P_{3T}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1T})$	down	$3 \times P_{1T}$
			P_{3R}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1R})$	down	$3 \times P_{1R}$

NOTE 2: V.44 parameters P_{2T} and P_{2R} are set to 255 and not negotiated in packet networks.

6.6.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.3.1.2 C_0

Two bits are used to indicate the V.44 method of operation supported (refer to [11] Annex B).

10 Packet Method supported.

11 Packet Method and Multi-Packet Method supported.

If parameter C_0 is not specified then Packet Method is selected with its default parameter values (refer to subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2).

********** NEXT MODIFIED SECTION **********

6.8 XID parameter negotiation

Negotiation of XID parameters between peer SNDCP entities may be carried out to ensure optimal information transfer. The parameters are called SNDCP exchange identity (XID) parameters.

SNDCP XID parameter negotiation may be initiated by the SNDCP entity at the MS or at the SGSN. If SNDCP XID parameters are to be changed, SNDCP XID negotiation shall be initiated prior to data transfer - the MS shall initiate SNDCP XID negotiation upon receipt of SNSM-ACTIVATE.indication; the SGSN shall initiate SNDCP XID negotiation upon receipt of the SNSM-MODIFY.indication primitive if an NSAPI has been put into use (in the case of an Inter-SGSN Routeing Area Update), or if the change in QoS profile to an existing NSAPI results in a change in compressor(s) used by the NSAPI.

When an NSAPI no longer uses a compression entity due to a PDP context deactivation or a PDP context modification, an SNDCP XID negotiation shall be performed to remove the NSAPI from the Applicable NSAPIs of the compression entity. The negotiation shall be initiated by the MS upon receipt of the SNSM-DEACTIVATE.indication in the case of PDP context deactivation, or by the SGSN upon receipt of the SNSM-MODIFY.indication in the case of PDP context modification.

The XID negotiation is a one-step procedure; i.e. the initiating end proposes parameter values, and the responding end either accepts these or offers different values in their place according to the XID negotiation rules described in the present document; the rules limit the range of parameter values as well as the sense of negotiation. The initiating end accepts (or rejects) the values in the response; this concludes the negotiation.

The block format for the SNDCP XID parameter negotiation is shown in figure 10. Not all parameters have to be included in the XID block, only parameters that are negotiated. Parameters may be included in any order. Also it shall be possible to negotiate parameters for more than one NSAPI in one XID block since more than one NSAPI can use the same SAPI.

Bit	8	7	6	5	4	3	2	1
Octet 1	Parameter type=0							
Octet 2	Length=1							
Octet 3	Version number							
Octet 4	Parameter type=1							
Octet 5	Length=n-5							
Octet 6	P	X	X	Entity number				
Octet 7 (optional)								
Octet 8	Length=k-8							
Octet 9 … (optional)								
Octet j	High-order octet							
…	…							
Octet k	Low-order octet							
Octet k+1	P	X	X	Entity number				
Octet k+2 (optional)								
Octet k+3	Length=m-(k+3)							
Octet k+4… (optional)								
Octet k+y	High-order octet							
…	…							
Octet m	Low-order octet							
…	…							
Octet n	Low-order octet							
Octet n+1	Parameter type=2							
Octet n+2	Length=r-(n+2)							
Octet n+3	P	X	X	Entity number				
Octet n+4 (optional)								
Octet n+5	Length=p-(n+5)							
Octet n+6… (optional)								
Octet n+w	High-order octet							
…	…							
Octet p	Low-order octet							
Octet p+1	P	X	X	Entity number				
Octet p+2 (optional)								
Octet p+3	Length=q-(p+3)							
Octet p+4… (optional)								
Octet p+v	High-order octet							
…	…							
Octet q	Low-order octet							
…	…							
Octet r	Low-order octet							

Figure 10: Example of SNDCP XID block format

The SNDCP user uses SN-XID.request to initiate the negotiation of the XID parameters. The SNDCP entity sends the proposed SNDCP XID parameters to the LLC SAP with the LL-XID.request or LL-ESTABLISH.request. The LLC SAP shall issue an XID command containing the SNDCP XID parameters (see 3GPP TS 44.064). The peer LLC SAP shall, upon receipt of the XID command, indicate the SNDCP XID parameters to SNDCP entity using LL-XID.indication or LL-ESTABLISH.indication. The peer SNDCP entity shall select appropriate values for the proposed parameters or negotiate the appropriate values with the SNDCP user entity with the SN-XID.indication and SN-XID.response primitives. When the appropriate parameter values are known by the peer SNDCP entity, it shall use the LL-XID.response or LL-ESTABLISH.response primitive to continue negotiation. Upon reception of the response, the LLC SAP shall send the received parameters to the SNDCP entity using the LL-XID.confirm or LL-ESTABLISH.confirm primitive. The SNDCP entity delivers the negotiated parameters to the SNDCP user. This is illustrated in figure 11. The originator of the negotiation shall apply the new parameter values after it has received the 'confirm' primitive. The responding end of the negotiation shall apply the new parameter values after it has sent the replying 'response' primitive.

Following the sending of the LL-XID.request primitive, the SNDCP layer shall suspend the transfer of SN-DATA and SN-UNITDATA primitives to the LLC SAP to which the LL-XID.request is sent. Transfer of SN-DATA and SN-UNITDATA primitives shall resume when the SNDCP XID negotiation ends through one of the following means:

- successful (receiving LL-XID.confirm);
- failure (receiving LL-RELEASE.indication, or LL-STATUS.indication); or
- successful following collision resolution (receiving LL-ESTABLISH.indication and sending LL-ESTABLISH.response, or receiving LL-XID.indication and sending LL-XID.response, see subclause 6.2.1.4).

LLC may also initiate LLC XID negotiation, in which case LLC may send an LL-XID.indication to inform SNDCP the values of N201-I and N201-U. This is illustrated in figure 12. If the SNDCP entity receives an LL-XID.indication without an SNDCP XID block, it shall not respond with the LL-XID.response primitive.

Negotiation of SNDCP version number is always between the peer SNDCP entities. The version number is not known by the SNDCP user. However, negotiation of the parameters for compression algorithms may be carried out between the SNDCP user entities.

Negotiation of SNDCP XID parameters for an NSAPI shall be carried out in the SAPI to which the NSAPI is mapped.

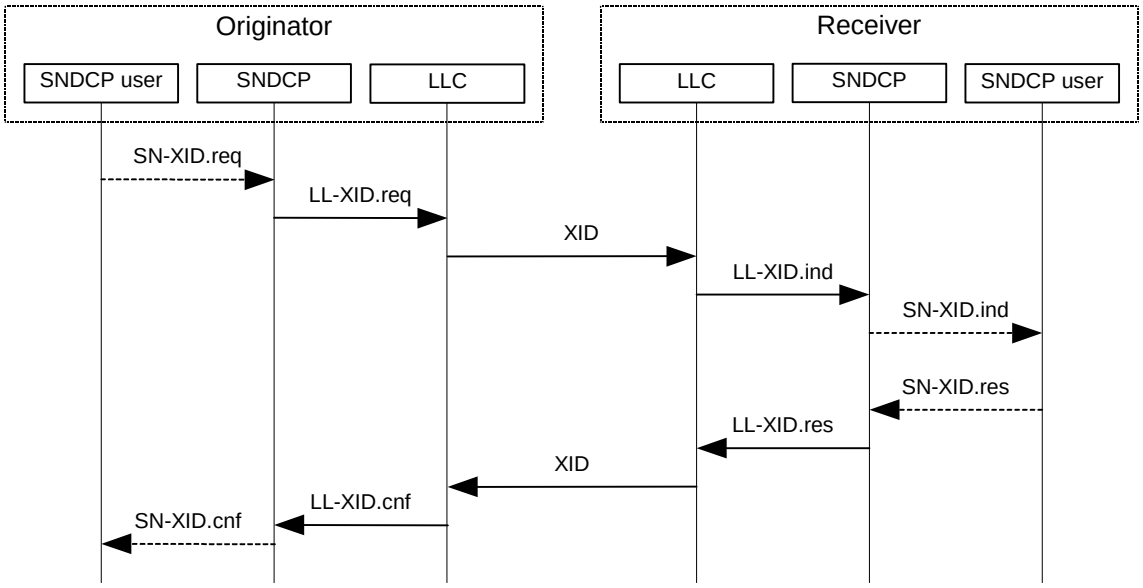

Figure 11: SNDCP XID negotiation procedure

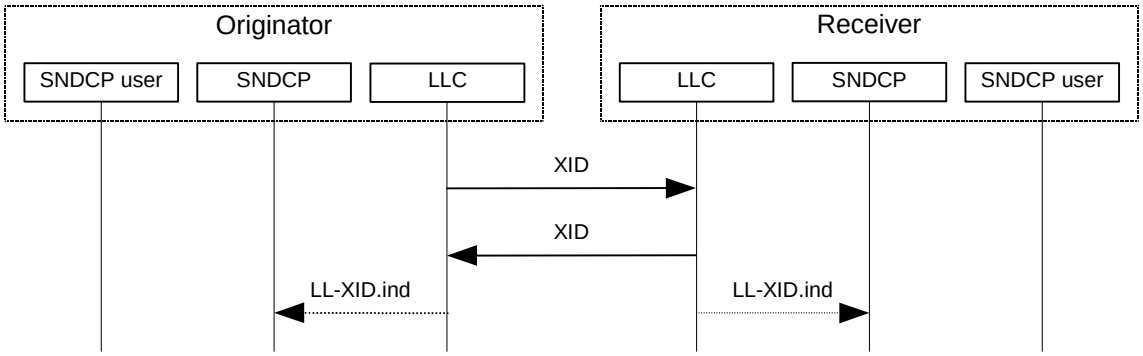

Figure 12: LLC XID negotiation procedure

6.8.1 Negotiation of compression entities

For parameter type 1 and 2, multiple compression fields (as shown in figure 7 and figure 9) may be specified. Each compression field corresponds to a compression entity.

In each compression field, the "Applicable NSAPIs" parameter indicates the NSAPIs that uses the compression entity. The parameter, if included, shall consist of 2 octets. Multiple NSAPIs may share the same compression entity by setting multiple bits in the parameter. NSAPIs requiring acknowledged peer-to-peer LLC operation and unacknowledged peer-to-peer LLC operation shall not share the same compressor (see subclause 6.10).

During SNDCP XID negotiation or re-negotiation, if a parameter type is specified in the SNDCP XID block, compression entities currently in use and compression entities proposed to be added may be included in the SNDCP XID block. Not all entities need to be included in the SNDCP XID block. If a compression entity is not included, the value of its parameters shall be determined by the rules defined in subclause 6.8.2.

If, implicitly or explicitly (see subclause 6.8.2), a compression entity is specified in the responding SNDCP XID block with one or more bits set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be created (if it does not exist yet).

If, implicitly or explicitly, a compression entity is specified in the responding SNDCP XID block with no bit set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be deleted (if it currently exists).

If an SNDCP entity in the MS receives a proposal for a compression entity containing an unknown algorithm, it should reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0.

NOTE 1: Some legacy MS implementations will ignore a proposed compression entity containing an unknown algorithm and will not reply with an explicit rejection.

If an SNDCP entity in the SGSN receives a proposal for a compression entity containing an unknown algorithm, it shall reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0.

If an SNDCP entity wants to accept a proposal for a data compression entity with algorithm V.44, it shall include in the responding SNDCP XID block at least the corresponding compression entity number.

NOTE 2: Without this explicit acknowledgement, the SNDCP entity proposing the data compression entity cannot decide whether the peer SNDCP entity supports V.44 and wants to accept the proposed compression entity implicitly or whether it is a legacy implementation which ignored the compression entity because of the unknown algorithm.

If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity.

If, implicitly or explicitly, one or more bits are set to 0 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall release the compression entity (if they have been using the compression entity).

6.8.2 Values of SNDCP XID parameters

In this subclause, the term "parameter" refers to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If an SNDCP XID parameter has not been negotiated, default values shall apply. The default value for a compression field (entity) is "non-existing".

If the originating SNDCP XID block does not include a parameter (implicit command), it shall be treated as equivalent to requesting for the current value for the parameter. The responder may explicitly include this parameter in its response. If the responder explicitly includes the parameter in the response, then it shall also explicitly include this parameter in every SNDCP XID response until the parameter has been explicitly negotiated, either by responding to an SNDCP XID command that included the parameter, or by explicitly including the parameter the next time an SNDCP XID command is transmitted.

If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator.

If both the originator and the responder do not include a parameter in the negotiation, the value of the parameter is not changed.

6.8.3 Exception handling

In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If the originating SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (as specified in clause 8, table 8), the parameter shall be ignored by the responder.

If the originating SNDCP XID block includes a parameter with unsupported length or an out-of-range value, then the responder shall respond to the parameter with lengths and values set according to the responder's preference.

If the originating SNDCP XID block includes parameter type 1 or 2 which violates the rules in subclause 6.8.1, the responder shall treat the parameter as not transmitted by the originator, and responds according to subclause 6.8.2.

If the originating SNDCP XID block includes a parameter with duplicated instances, the subsequent instances of the duplicated parameter shall be ignored.

If the originating SNDCP XID block is sent on LL-XID primitives and contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation, then the responder shall respond with these parameters set to their previously-negotiated values.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, when an assigned entity number is included with the P bit set to 1, the algorithm and the PCOMP and DCOMP fields shall be ignored if they are the same as the previously-assigned values. If the algorithm and PCOMP or DCOMP fields are not the same as the previously-assigned values, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if an unassigned entity number is included with the P bit set to 0, then the Applicable NSAPIs field in the response shall be set to 0.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, if one or more of the PCOMP or DCOMP specified is already assigned to a different compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if one or more new PCOMP or DCOMP values are specified for an existing compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the responding SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (see clause 8, table 8), unsupported length, an out-of-range value or a value violating the sense of negotiation, a parameter type 1 or 2 which violates the rules in subclause 6.8.1, a parameter with duplicated instances, contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation when the SNDCP XID block is sent on LL-XID primitives, or a compression field with the P bit set to 1, then the originator shall ignore the block and reinitiate the negotiation. If the renegotiation fails for an implementation-specific number of times, the originating SNDCP layer shall send an SNSM-STATUS.request primitive with Cause "invalid XID response" to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the LLC layer indicates that the XID parameter negotiation failed, by sending an LL-RELEASE.indication with Cause "no peer response" or an LL-STATUS.indication with Cause "no peer response", then, as an implementation option, the SNDCP layer may wait for an implementation-specific amount of time and re-invoke the XID negotiation procedure.

********** NEXT MODIFIED SECTION **********

6.10 Possible combinations of SNDCP protocol functions and their connection to service access points

The following combinations of SNDCP protocol functions are allowed:

- One or several NSAPIs may use one SAPI.
- Only one SAPI shall be used by one NSAPI.
- One or several NSAPIs may use the same protocol control information compression entity.
- One NSAPI may use zero or~~,~~ one~~, or several~~ protocol control information compression entities.
- One or several NSAPIs may use the same data compression entity.
- One NSAPI may use zero or~~,~~ one~~, or several~~ data compression entities.
- Separate data compression entities shall be used for SN-DATA and SN-UNITDATA PDUs.
- Separate protocol control information compression entities shall be used for SN-DATA and SN-UNITDATA PDUs.
- One data compression entity shall be connected to one SAPI.
- One protocol control information compression entity shall be connected to one SAPI.
- One or several protocol control information compression entities may be connected to the same data compression entity.
- One protocol control information compression entity shall be connected to zero, one, or several data compression entities.

3GPP TSG-CN1 Meeting #35
Sophia Antipolis, France, 16-20 August 2004

Tdoc N1-041635
(rev of Tdoc N1-041617)

CR-Form-v7.1

CHANGE REQUEST

⌘ **44.065** **CR** **016** ⌘ **rev** **2** ⌘ Current version: **5.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Negotiation of compression entities with unknown algorithm type
Source:	⌘	Siemens AG
Work item code:	⌘	TEI4
Date:	⌘	06.08.2004
Category:	⌘	**A**
Release:	⌘	Rel-5

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change: ⌘ Ambiguity in the specification of the negotiation of compression entities:

According to subclauses 6.8.1 and 6.8.2:
> "If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity."

and
> "If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator."

I.e. an SNDCP entity can accept a compression entity proposed by its peer SNDCP entity by not including it in the responding SNDCP XID block.

On the other hand, according to subclause 6.8.3, exception handling:
> "In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.
> If the originating SNDCP XID block includes a parameter with **unrecognised Type field**, the parameter shall be ignored by the responder."

Some manufacturers have interpreted this requirement in such a way that it applies also to the parameter "algorithm type" in the SNDCP XID parameters "protocol control information compression" and "data compression".

	With such an interpretation, however, in some scenarios the SNDCP entity originating the XID negotiation cannot decide whether the peer SNDCP entity did not respond to a proposed compression entity in order to accept it or because it did not know the algorithm and therefore ignored the proposed compression entity. Example: Some R97/98 MS implementations ignore a proposal for a protocol control information compression with algorithm type RFC2507, because this algorithm was only introduced in R99. Since the MS does not explicitly reject the proposed compression entity for RFC2507, a R99 SGSN following subclauses 6.8.1 and 6.8.2 will assume that the MS has accepted the proposal and will start using this compression entity. The MS will then discard the downlink packets compressed by the SGSN with RFC2507. For the algorithm RFC2507 which was introduced in R99, the SNDCP entity originating the XID negotiation can resolve this ambiguity only by evaluating the revision level of its peer entity (revision level in the MS network capability IE or SGSNR bit in the BCCH or PBCCH, respecitvely). In order to remove this ambiguity for algorithms introduced in Rel-4 or later, the following is proposed:
Summary of change: ⌘	1) It is clarified that in the "protocol control information compression" field and "data compression" field the two octets following the PCOMP or DCOMP values always contain the "Applicable NSAPIs" parameter. 2) The parameter "algorithm type" is renamed to "algorithm identifier" so that 6.8.3 is no longer applicable to unknown compression algorithms. 3) It is clarified that if an SNDCP entity in the **MS** receives a proposal for a compression entity containing an unknown algorithm, it **should** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0. From Rel-5 onwards, with SNDCP version 1 this behaviour is mandatory. If an SNDCP entity in the **SGSN** receives a proposal for a compression entity containing an unknown algorithm, it **shall** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0, from Rel-4 onwards . 4) From Rel-5 onwards, the SDNCP version number is raised to 1 so that the entity originating the XID negotiation can determine whether its peer entity is a R99 implementation or a 'Rel-4 or later' implementation. 5) If an SNDCP entity in Rel-4 wants to accept a proposal for a data compression entity with algorithm V.44, it shall include in the responding SNDCP XID block at least the corresponding compression entity number. The length values in the compression parameter table for V.44 are corrected. 6) The peer entity of the XID negotiation shall accept only one header compression algorithm and one data compression algorithm for an NSAPI.
Consequences if not approved: ⌘	It will not be possible to use the compression algorithms introduced after R99 (V.44 and ROHC), since it will not be possible for the entity originating the XID negotiation to determine whether its peer entity is a legacy R99 implementation or a 'Rel-4 or later' implementation. Consequently the entity originating the XID negotiation will not be able to decide whether the peer SNDCP entity did not respond to a proposed compression entity for V.44 or ROHC in order to accept it or because the peer entity did not know the algorithm and therefore ignored the proposed compression entity.

Clauses affected:	⌘	6.5.1, 6.5.1.1, 6.5.1.1.4, 6.5.2.1, 6.5.3.1, 6.5.4.1, 6.6, 6.6.1, 6.6.1.1, 6.6.1.1.4, 6.6.2.1, 6.6.3.1, 6.8.1, 6.8.1a (new), 6.8.3, 6.10, 8			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.5 Protocol control information compression

Protocol control information compression is an optional SNDCP feature.

Negotiation of the supported algorithms and their parameters is carried out between MS and SGSN using the SNDCP XID parameters (see clause 8).

6.5.1 Negotiation of multiple protocol control information compression types

Each SNDCP entity that supports protocol control information compression shall be able to negotiate one or several protocol control information compression entities with the compression field format shown in figure 7. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one protocol control information compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

6.5.1.1 Format of the protocol control information compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier~~type~~				
Octet 3	Length=n-3							
Octet 4	PCOMP1				PCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 7: Protocol control information compression field format for SNDCP XID negotiation

6.5.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.5.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of PCOMP values needed by the corresponding compression algorithm (e.g. PCOMP3 and PCOMP4 shall not be included if the number of PCOMP values needed by a compression algorithm is one or two). If an odd number of PCOMP values are used by a compression algorithm, then the last PCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.5.1.1.3 Entity number

The entity number shall be used to identify a protocol control information compression entity on a SAPI. The entity number shall be assigned using the following rules:

- The entity number shall be an integer from 0 to 31.
- The entity number shall be assigned independently on each of the SAPIs.
- An entity number shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, an unassigned entity number shall become selected. If there is no unassigned entity number left, the compression entity shall not be proposed.
- A selected entity number shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned entity number shall become unassigned when the corresponding compression entity is deleted as a result of an XID negotiation, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4) in which an entity number is currently selected:
  - If the selected entity number is included with the P bit set to 0 in the incoming SNDCP XID block, then it shall be assumed that the peer SNDCP entity agreed to the creation of the proposed entity but the response was lost. Therefore the selected entity number shall become assigned, any selected PCOMP and DCOMP values for the algorithm of the entity shall become assigned, and the compression entity shall be created, before the incoming SNDCP XID block is processed. After the incoming SNDCP XID block is processed, the compression entity shall be negotiated again if necessary, as defined in subclause 6.2.1.4.
  - Otherwise (i.e. if the selected entity number is not included, or is included with the P bit set to 1 in the incoming SNDCP XID block), the selected entity number shall become unassigned, and any selected PCOMP and DCOMP values for the algorithm of the entity shall become unassigned, before the incoming SNDCP XID block, if any, is processed. Following the collision resolution procedure, the originally-proposed compression entity shall be proposed again (i.e. the originally-proposed compression entity shall not be considered created even if the originally-selected entity number is proposed in the incoming SNDCP XID block) by sending the appropriate primitive (LL-ESTABLISH.request or LL-XID.request). The originally-selected entity number, PCOMP and DCOMP values shall be used for the compression entity being re-proposed if they are unassigned, otherwise a new entity number, PCOMP or DCOMP value shall be selected.
- In the case of a collision in which an entity number is currently assigned:
  - If the peer SNDCP entity proposes a new compression entity with the same entity number, then it shall be assumed that the peer SNDCP entity negotiated the deletion of the entity but the response was lost, and the entity number is being reused. Therefore the original compression entity shall be deleted, the entity number shall become unassigned, PCOMP and DCOMP values shall be unassigned if necessary (see subclause 6.5.1.1.5), and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned entity number is not included, or is included with the P bit set to 0 in the incoming SNDCP XID block), the usual rules regarding collision handling shall apply.
- In the case of a collision in which a PCOMP or DCOMP value is currently assigned to a compression algorithm:
  - If the peer SNDCP entity proposes a new compression entity with the same PCOMP or DCOMP assigned to a different algorithm, then it shall be assumed that the peer SNDCP entity negotiated the deletion of all entities using the algorithm to which the PCOMP or DCOMP value was assigned, but the response was lost, and the PCOMP or DCOMP value is being reused. Therefore, all compression entities using that algorithm shall be deleted, all corresponding entity numbers shall become unassigned, and all PCOMP or DCOMP values assigned to the algorithm shall become unassigned, and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned PCOMP or DCOMP is not included, or is included and assigned to the same algorithm), the usual rules regarding collision handling shall apply.

6.5.1.1.4 Algorithm identifier~~type~~

Table 4 show the list of protocol control information compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 4 shall be updated.

Table 4: List of protocol control information compression algorithms supported by SNDCP

Compression algorithm	Algorithm identifier ~~type~~ (Range 0 to 31)
RFC 1144	0
RFC 2507	1
-	Other values Reserved

6.5.1.1.5 PCOMP

One or more PCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for protocol control information compression. Each of the assigned PCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the PCOMP values follows the following general rules:

- PCOMP shall be an integer from 0 to 15.
- PCOMP value 0 is reserved permanently for no compression.
- PCOMP shall be assigned independently on each of the SAPIs.
- An assigned PCOMP value applies to all NSAPIs mapped to the same SAPI.
- PCOMP values shall be assigned to compression algorithms, not to compression entities (i.e. the same PCOMP value(s) shall be used by different compression entities on the same SAPI using the same compression algorithm).
- A PCOMP value shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, and if PCOMP values have not yet been assigned to the corresponding compression algorithm, then the appropriate number of unassigned PCOMP values shall be selected. If there is not enough unassigned PCOMP values left, the compression entity shall not be proposed.
- A selected PCOMP value shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned PCOMP value shall become unassigned when the corresponding compression algorithm is no longer in use by any compression entity, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4), the handling of PCOMP values shall be in accordance with subclause 6.5.1.1.3.

While transferring data, the compressed frame type for an N-PDU is conveyed in the PCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.5.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of protocol control information compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the protocol control information compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.5.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.5.2 TCP/IP header compression (RFC1144)

The protocol control information compression method is specific for each network layer protocol type. TCP/IP (IPv4) header compression is specified in RFC 1144 [9].

6.5.2.1 Parameters

Table 5 contains the parameters defined for a compression entity using TCP/IP header compression. They may be negotiated during SNDCP XID negotiation.

Table 5: RFC 1144 TCP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 1144	0	0, 2 or 3 if P bit is 0, 1, 3 or 4 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			S_0 - 1	bbbbbbbb	0 through 255	down	15

6.5.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.2.1.2 S_0

The number of state slots, as defined in [9]. The S_0 range is 1 through 256, with 16 as default value.

6.5.2.2 Assignment of PCOMP values

The underlying service shall be able to distinguish the three types of compressed N-PDUs (i.e. Type IP, Uncompressed TCP, and Compressed TCP), as defined in RFC 1144 [9]. These three N-PDU types are differentiated by using different PCOMP values.

Two PCOMP values shall be assigned to the TCP/IP header compression algorithm. PCOMP1 shall contain the PCOMP value for the frame type "Uncompressed TCP", and PCOMP2 shall contain the PCOMP value for the frame type "Compressed TCP".

The PCOMP value of 0 shall be used for the frame type "Type IP".

6.5.2.3 Error Recovery

When TCP/IP header compression is used with unacknowledged peer-to-peer LLC operation, the decompression entity shall be notified in case an N-PDU is dropped, so that error recovery procedure (see RFC 1144 [9]) can be invoked.

6.5.3 TCP/IP and UDP/IP header compression (RFC 2507)

Detailed operation of the RFC 2507 header compression for IPv4 and IPv6 is described in clause 3 of the IETF specification RFC 2507 [10].

6.5.3.1 Parameters

Table 6 contains the parameters defined for a compression entity using RFC2507 header compression. They may be negotiated during SNDCP XID negotiation.

Table 6: RFC 2507 TCP/IP and UDP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier ~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 2507	1	0, 2, 4, 5, 6, 7 or 9 if P bit is 0, 3, 5, 7, 8, 9, 10 or 12 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			F_MAX_PERIOD	bbbbbbbb bbbbbbbb	1-65535	down	256
			F_MAX_TIME	bbbbbbbb	1-255	down	5
			MAX_HEADER	bbbbbbbb	60-255	down	168
			TCP_SPACE	bbbbbbbb	3-255	down	15
			NON_TCP_SPACE	bbbbbbbb bbbbbbbb	3-65535	down	15

The explanation of the individual parameters can be found in the clause 14 of the IETF specification RFC 2507 [10].

6.5.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.3.2 Assignment of PCOMP values for RFC2507

The following PCOMP values shall be assigned to the RFC 2507 header compression. The PCOMP value 0 shall be used for regular IPv4 and IPv6 packets.

Table 7: PCOMP values assigned to RFC 2507 header compression algorithm

PID value	Packet type
PCOMP1	Full header
PCOMP2	Compressed TCP
PCOMP3	Compressed TCP non-delta
PCOMP4	Compressed non-TCP
PCOMP5	Context state

6.5.3.3 Error Recovery

The mechanisms related to error recovery and packet reordering are described in clauses 10 and 11 of the RFC 2507[10].

6.6 Data compression

Data compression is an optional SNDCP feature. Data compression applies to both SN-DATA and SN-UNITDATA primitives.

Data compression, if used, shall be performed on the entire N-PDU, including the possibly compressed protocol control information.

Figure 8 shows an example how the SNDCP functions may be used. Several NSAPIs may use a common data compression entity, i.e. the same compression algorithm and the same dictionary. Separate data compression entities shall be used for acknowledged (SN-DATA) and unacknowledged (SN-UNITDATA) data transfer. Several NSAPIs may be associated with one SAPI, i.e. they may use the same QoS profile.

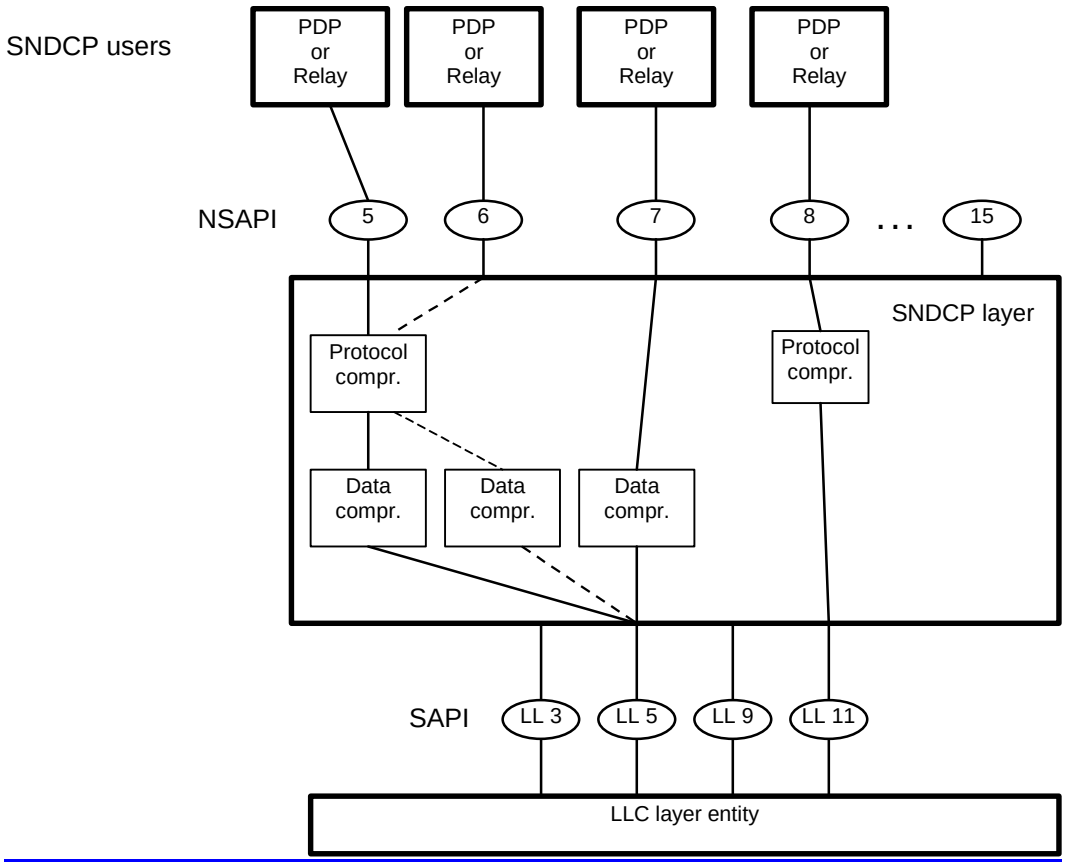

Figure 8: An example for the usage of NSAPIs, SNDCP functions, and SAPIs

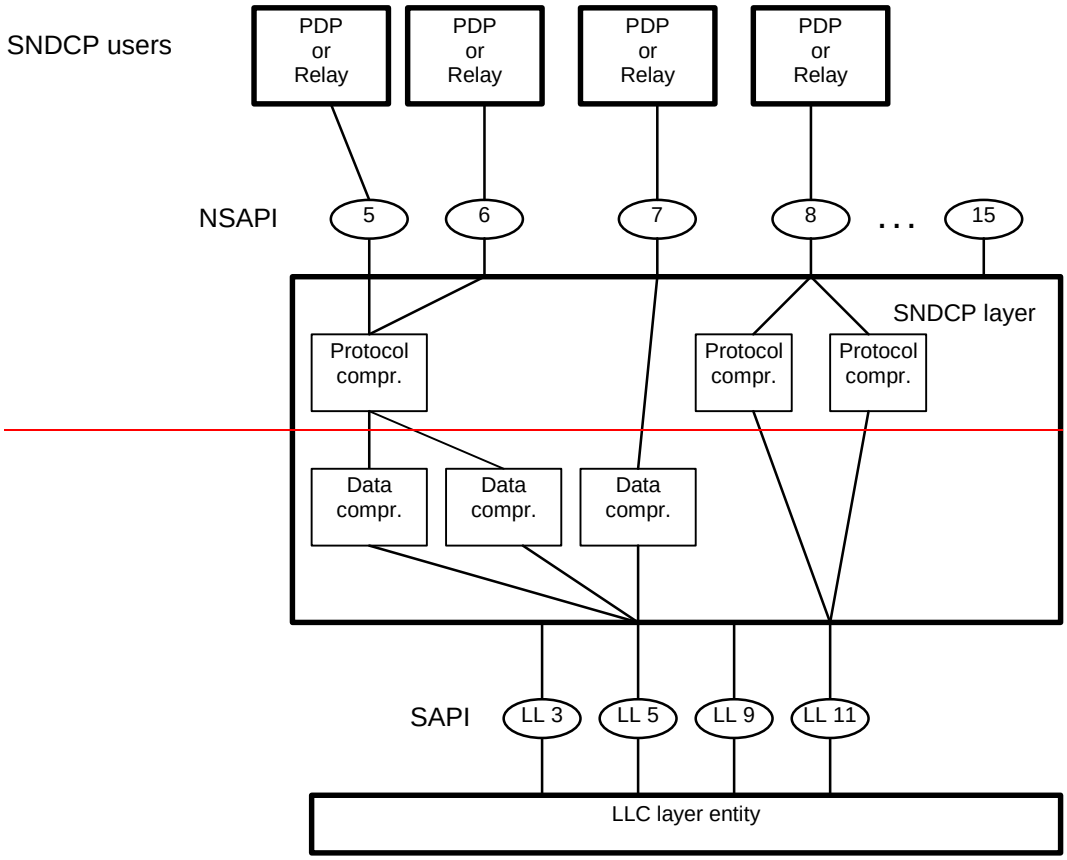

Figure 8: An example for the usage of NSAPIs, SNDCP functions, and SAPIs

6.6.1 Negotiation of multiple data compression types

Each SNDCP entity that supports data compression shall be able to negotiate one or several data compression entities with the compression field format shown in figure 9. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one data compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

For each NSAPI one or more data compression are chosen. This choice is also indicated in the SNDCP XID. Only NSAPIs that are using the same SAPI may use the same data compression entity. If more than one compression entity is chosen for an NSAPI, these entities must use different data compression algorithms. However, only one data compression entity is used for one N-PDU; i.e. the used data compression entity may be changed from N-PDU to N-PDU.

6.6.1.1 Format of the data compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier~~type~~				
Octet 3	Length=n-3							
Octet 4	DCOMP1				DCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 9: Data compression field format for SNDCP XID negotiation

6.6.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.6.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of DCOMP values needed by the corresponding compression algorithm (e.g. DCOMP3 and DCOMP4 shall not be included if the number of DCOMP values needed by a compression algorithm is one or two). If an odd number of DCOMP values are used by a compression algorithm, then the last DCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.6.1.1.3 Entity number

The entity number shall be used to identify a data compression entity on a SAPI. See subclause 6.5.1.1.3 for the rules for assigning entity numbers. The assignment of entity numbers for protocol control information compression entities and data compression entities shall be independent.

6.6.1.1.4 Algorithm identifier~~type~~

Table 6a shows the list of data compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 6a shall be updated.

Table 6a: List of data compression algorithms supported by SNDCP

Data compression algorithm	Algorithm identifier~~type~~ (Range 0-31)
V.42 bis	0
V.44	1
-	Other values Reserved

6.6.1.1.5 DCOMP

One or more DCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for data compression. Each of the assigned DCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the DCOMP values shall follow the rules for the assignment of PCOMP values in subclause 6.5.1.1.5.

While transferring data, the compressed frame type for an N-PDU is conveyed in the DCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.6.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of data compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the data compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.6.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.6.2 Management of V.42 bis data compression

ITU-T Recommendation V.42 bis [8] data compression may be used with SN-DATA primitives and SN-UNITDATA primitives.

6.6.2.1 Parameters

Table 7a contains the parameters defined for a compression entity using ITU-T Recommendation V.42 bis data compression. They may be negotiated during SNDCP XID negotiation.

Table 7a: V.42 bis data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.42 bis	0	0, 2, 3, 5, or 6 if P bit is 0, 1, 3, 4, 6, or 7 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_1	bbbbbbbb bbbbbbbb	512 through 65535	down	2048
			P_2	bbbbbbbb	6 through 250	down	20

6.6.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.2.1.2 P_0

Two bits are used to indicate the usage of compression, one bit for each direction.

00 compress neither direction.

01 compress MS-to-SGSN direction only.

10 compress SGSN-to-MS direction only.

11 compress both directions.

6.6.2.1.3 P_1

Maximum number of codewords in the compressor dictionary (see [8]).

6.6.2.1.4 P_2

Maximum number of characters in an uncompressed data string that is accepted to be encoded.

6.6.2.2 Assignment of DCOMP values

One DCOMP value shall be assigned (as DCOMP1) to the V.42 bis data compression algorithm.

6.6.2.3 Operation of V.42 bis data compression

When V.42 bis is used with SN-DATA primitives, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent.

When V.42 bis is used with SN-UNITDATA primitives, the compression entity shall be reset (using the C-INIT primitive defined in [8]) before an N-PDU is compressed or decompressed. After compression, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent. The LLC protocol shall operate in the protected mode of operation.

When V.42 bis is used with SN-DATA primitives and an error is detected by the decoder, the SNDCP entity shall use LL-ESTABLISH.request primitive to reset the acknowledged peer-to-peer LLC operation for the SAPI used.

6.6.3 Management of V.44 data compression

ITU-T Recommendation V.44 data compression, as described in [11], may be used with SN-DATA primitives and SN-UNITDATA primitives. Annex B of ITU-T Recommendation V.44 describes two methods of implementation and operation of V.44 in packet networks: Packet Method and Multi-Packet Method. Multi-Packet Method is a superset of Packet Method and an MS or SGSN that supports Multi-Packet Method must also support Packet Method.

6.6.3.1 Parameters

Table 7c contains the parameters defined for a compression entity using V.44 data compression. They may be negotiated during SNDCP XID negotiation. During V.44 data compression negotiation, unless both the MS and SGSN support Multi-Packet Method, Packet Method is used. Parameter C_0 indicates support of Packet Method (10000000) or both methods (11000000).

NOTE 1: V.44 data compression negotiation is not required. If V.44 is selected and no compression parameters are specified, then Packet Method with defaults as defined in subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2, is used.

Table 7c: V.44 data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier ~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.44	1	0, 2, 3, 4, 6, 8, 10, or 12 if P bit is 0, 1, 3, 4, 5, 7, 9, 11, or 13 if P bit is 1.~~5 or 6~~	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			C_0	bb000000	10000000 or 11000000	11000000 down to 10000000	10000000
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_{1T}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.4
			P_{1R}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.5
			P_{3T}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1T})$	down	$3 \times P_{1T}$
			P_{3R}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1R})$	down	$3 \times P_{1R}$

NOTE 2: V.44 parameters P_{2T} and P_{2R} are set to 255 and not negotiated in packet networks.

6.6.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.3.1.2 C_0

Two bits are used to indicate the V.44 method of operation supported (refer to [11] Annex B).

10 Packet Method supported.

11 Packet Method and Multi-Packet Method supported.

If parameter C_0 is not specified then Packet Method is selected with its default parameter values (refer to subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2).

********** NEXT MODIFIED SECTION **********

6.8 XID parameter negotiation

Negotiation of XID parameters between peer SNDCP entities may be carried out to ensure optimal information transfer. The parameters are called SNDCP exchange identity (XID) parameters.

SNDCP XID parameter negotiation may be initiated by the SNDCP entity at the MS or at the SGSN. If SNDCP XID parameters are to be changed, SNDCP XID negotiation shall be initiated prior to data transfer - the MS shall initiate SNDCP XID negotiation upon receipt of SNSM-ACTIVATE.indication; the SGSN shall initiate SNDCP XID negotiation upon receipt of the SNSM-MODIFY.indication primitive if an NSAPI has been put into use (in the case of an Inter-SGSN Routeing Area Update), or if the change in QoS profile to an existing NSAPI results in a change in compressor(s) used by the NSAPI.

When an NSAPI no longer uses a compression entity due to a PDP context deactivation or a PDP context modification, an SNDCP XID negotiation shall be performed to remove the NSAPI from the Applicable NSAPIs of the compression entity. The negotiation shall be initiated by the MS upon receipt of the SNSM-DEACTIVATE.indication in the case of PDP context deactivation, or by the SGSN upon receipt of the SNSM-MODIFY.indication in the case of PDP context modification.

The XID negotiation is a one-step procedure; i.e. the initiating end proposes parameter values, and the responding end either accepts these or offers different values in their place according to the XID negotiation rules described in the present document; the rules limit the range of parameter values as well as the sense of negotiation. The initiating end accepts (or rejects) the values in the response; this concludes the negotiation.

The block format for the SNDCP XID parameter negotiation is shown in figure 10. Not all parameters have to be included in the XID block, only parameters that are negotiated. Parameters may be included in any order. Also it shall be possible to negotiate parameters for more than one NSAPI in one XID block since more than one NSAPI can use the same SAPI.

Bit	8	7	6	5	4	3	2	1
Octet 1	Parameter type=0							
Octet 2	Length=1							
Octet 3	Version number							
Octet 4	Parameter type=1							
Octet 5	Length=n-5							
Octet 6	P	X	X	Entity number				
Octet 7 (optional)								
Octet 8	Length=k-8							
Octet 9 … (optional)								
Octet j	High-order octet							
…	…							
Octet k	Low-order octet							
Octet k+1	P	X	X	Entity number				
Octet k+2 (optional)								
Octet k+3	Length=m-(k+3)							
Octet k+4… (optional)								
Octet k+y	High-order octet							
…	…							
Octet m	Low-order octet							
…	…							
Octet n	Low-order octet							
Octet n+1	Parameter type=2							
Octet n+2	Length=r-(n+2)							
Octet n+3	P	X	X	Entity number				
Octet n+4 (optional)								
Octet n+5	Length=p-(n+5)							
Octet n+6… (optional)								
Octet n+w	High-order octet							
…	…							
Octet p	Low-order octet							
Octet p+1	P	X	X	Entity number				
Octet p+2 (optional)								
Octet p+3	Length=q-(p+3)							
Octet p+4… (optional)								
Octet p+v	High-order octet							
…	…							
Octet q	Low-order octet							
…	…							
Octet r	Low-order octet							

Figure 10: Example of SNDCP XID block format

The SNDCP user uses SN-XID.request to initiate the negotiation of the XID parameters. The SNDCP entity sends the proposed SNDCP XID parameters to the LLC SAP with the LL-XID.request or LL-ESTABLISH.request. The LLC SAP shall issue an XID command containing the SNDCP XID parameters (see 3GPP TS 44.064 [6]). The peer LLC SAP shall, upon receipt of the XID command, indicate the SNDCP XID parameters to SNDCP entity using LL-XID.indication or LL-ESTABLISH.indication. The peer SNDCP entity shall select appropriate values for the proposed parameters or negotiate the appropriate values with the SNDCP user entity with the SN-XID.indication and SN-XID.response primitives. When the appropriate parameter values are known by the peer SNDCP entity, it shall use the LL-XID.response or LL-ESTABLISH.response primitive to continue negotiation. Upon reception of the response, the LLC SAP shall send the received parameters to the SNDCP entity using the LL-XID.confirm or LL-ESTABLISH.confirm primitive. The SNDCP entity delivers the negotiated parameters to the SNDCP user. This is illustrated in figure 11. The originator of the negotiation shall apply the new parameter values after it has received the 'confirm' primitive. The responding end of the negotiation shall apply the new parameter values after it has sent the replying 'response' primitive.

Following the sending of the LL-XID.request primitive, the SNDCP layer shall suspend the transfer of SN-DATA and SN-UNITDATA primitives to the LLC SAP to which the LL-XID.request is sent. Transfer of SN-DATA and SN-UNITDATA primitives shall resume when the SNDCP XID negotiation ends through one of the following means:

- successful (receiving LL-XID.confirm);
- failure (receiving LL-RELEASE.indication, or LL-STATUS.indication); or
- successful following collision resolution (receiving LL-ESTABLISH.indication and sending LL-ESTABLISH.response, or receiving LL-XID.indication and sending LL-XID.response, see subclause 6.2.1.4).

LLC may also initiate LLC XID negotiation, in which case LLC may send an LL-XID.indication to inform SNDCP the values of N201-I and N201-U. This is illustrated in figure 12. If the SNDCP entity receives an LL-XID.indication without an SNDCP XID block, it shall not respond with the LL-XID.response primitive.

Negotiation of SNDCP version number is always between the peer SNDCP entities. The version number is not known by the SNDCP user. However, negotiation of the parameters for compression algorithms may be carried out between the SNDCP user entities.

Negotiation of SNDCP XID parameters for an NSAPI shall be carried out in the SAPI to which the NSAPI is mapped.

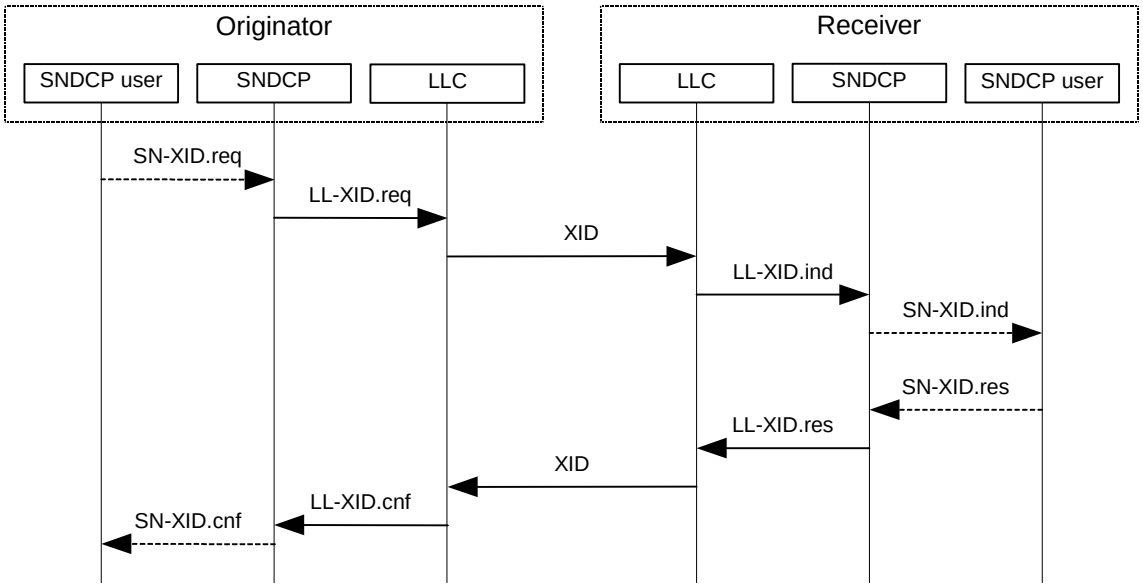

Figure 11: SNDCP XID negotiation procedure

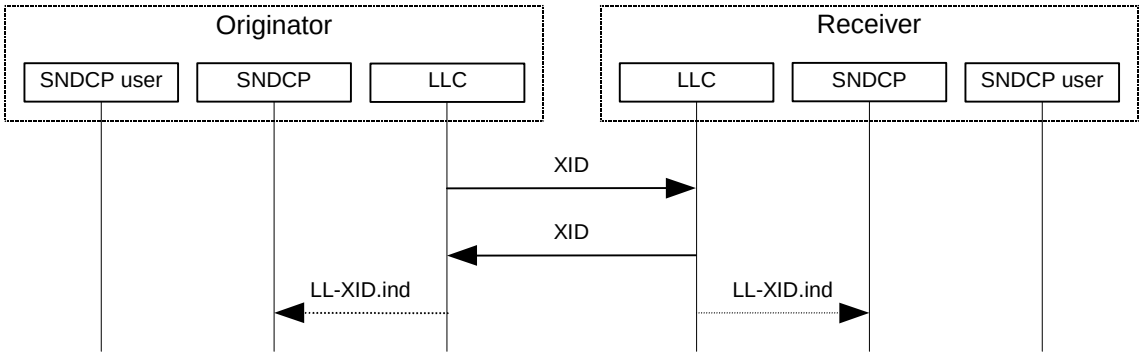

Figure 12: LLC XID negotiation procedure

6.8.1 Negotiation of compression entities

For parameter type 1 and 2, multiple compression fields (as shown in figure 7 and figure 9) may be specified. Each compression field corresponds to a compression entity.

In each compression field, the "Applicable NSAPIs" parameter indicates the NSAPIs that uses the compression entity. The parameter, if included, shall consist of 2 octets. Multiple NSAPIs may share the same compression entity by setting multiple bits in the parameter. NSAPIs requiring acknowledged peer-to-peer LLC operation and unacknowledged peer-to-peer LLC operation shall not share the same compressor (see subclause 6.10).

During SNDCP XID negotiation or re-negotiation, if a parameter type is specified in the SNDCP XID block, compression entities currently in use and compression entities proposed to be added may be included in the SNDCP XID block. Not all entities need to be included in the SNDCP XID block. If a compression entity is not included, the value of its parameters shall be determined by the rules defined in subclause 6.8.2.

If, implicitly or explicitly (see subclause 6.8.2), a compression entity is specified in the responding SNDCP XID block with one or more bits set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be created (if it does not exist yet).

If, implicitly or explicitly, a compression entity is specified in the responding SNDCP XID block with no bit set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be deleted (if it currently exists).

If an SNDCP entity receives a proposal for a compression entity containing an unknown algorithm, it shall reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0.

NOTE: Some legacy MS implementations of SDNCP version 0 will ignore a proposed compression entity containing an unknown algorithm and will not reply with an explicit rejection.

If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity.

If, implicitly or explicitly, one or more bits are set to 0 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall release the compression entity (if they have been using the compression entity).

6.8.1a Negotiation of SNDCP version

An SNDCP entity supporting this version of SNDCP shall set the version number in the originating SNDCP XID block to 1. For the sense of negotiation of the version number see clause 8, table 8.

6.8.2 Values of SNDCP XID parameters

In this subclause, the term "parameter" refers to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If an SNDCP XID parameter has not been negotiated, default values shall apply. The default value for a compression field (entity) is "non-existing".

If the originating SNDCP XID block does not include a parameter (implicit command), it shall be treated as equivalent to requesting for the current value for the parameter. The responder may explicitly include this parameter in its response. If the responder explicitly includes the parameter in the response, then it shall also explicitly include this parameter in every SNDCP XID response until the parameter has been explicitly negotiated, either by responding to an SNDCP XID command that included the parameter, or by explicitly including the parameter the next time an SNDCP XID command is transmitted.

If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator.

If both the originator and the responder do not include a parameter in the negotiation, the value of the parameter is not changed.

6.8.3 Exception handling

In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If the originating SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (as specified in clause 8, table 8), the parameter shall be ignored by the responder.

If the originating SNDCP XID block includes a parameter with unsupported length or an out-of-range value, then the responder shall respond to the parameter with lengths and values set according to the responder's preference.

If the originating SNDCP XID block includes parameter type 1 or 2 which violates the rules in subclause 6.8.1, the responder shall treat the parameter as not transmitted by the originator, and responds according to subclause 6.8.2.

If the originating SNDCP XID block includes a parameter with duplicated instances, the subsequent instances of the duplicated parameter shall be ignored.

If the originating SNDCP XID block is sent on LL-XID primitives and contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation, then the responder shall respond with these parameters set to their previously-negotiated values.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, when an assigned entity number is included with the P bit set to 1, the algorithm and the PCOMP and DCOMP fields shall be ignored if they are the same as the previously-assigned values. If the algorithm and PCOMP or DCOMP fields are not the same as the previously-assigned values, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if an unassigned entity number is included with the P bit set to 0, then the Applicable NSAPIs field in the response shall be set to 0.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, if one or more of the PCOMP or DCOMP specified is already assigned to a different compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if one or more new PCOMP or DCOMP values are specified for an existing compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the responding SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (see clause 8, table 8), unsupported length, an out-of-range value or a value violating the sense of negotiation, a parameter type 1 or 2 which violates the rules in subclause 6.8.1, a parameter with duplicated instances, contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation when the SNDCP XID block is sent on LL-XID primitives, or a compression field with the P bit set to 1, then the originator shall ignore the block and reinitiate the negotiation. If the renegotiation fails for an

implementation-specific number of times, the originating SNDCP layer shall send an SNSM-STATUS.request primitive with Cause "invalid XID response" to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the LLC layer indicates that the XID parameter negotiation failed, by sending an LL-RELEASE.indication with Cause "no peer response" or an LL-STATUS.indication with Cause "no peer response", then, as an implementation option, the SNDCP layer may wait for an implementation-specific amount of time and re-invoke the XID negotiation procedure.

********** NEXT MODIFIED SECTION **********

6.10 Possible combinations of SNDCP protocol functions and their connection to service access points

The following combinations of SNDCP protocol functions are allowed:

- One or several NSAPIs may use one SAPI.
- Only one SAPI shall be used by one NSAPI.
- One or several NSAPIs may use the same protocol control information compression entity.
- One NSAPI may use zero~~,~~ or one~~, or several~~ protocol control information compression entities.
- One or several NSAPIs may use the same data compression entity.
- One NSAPI may use zero~~,~~ or one~~, or several~~ data compression entities.
- Separate data compression entities shall be used for SN-DATA and SN-UNITDATA PDUs.
- Separate protocol control information compression entities shall be used for SN-DATA and SN-UNITDATA PDUs.
- One data compression entity shall be connected to one SAPI.
- One protocol control information compression entity shall be connected to one SAPI.
- One or several protocol control information compression entities may be connected to the same data compression entity.
- One protocol control information compression entity shall be connected to zero, one, or several data compression entities.

********** NEXT MODIFIED SECTION **********

8 SNDCP XID parameters

The SNDCP XID parameters are shown in table 8:

Table 8: SNDCP XID parameters

Parameter name	Parameter Type	Length	Format	Range	Default value	Units	Sense of negotiation
Version number	0	1	0000bbbb	0-15	0	-	down
Data Compression	1	variable	See subclause 6.6.1				
Protocol Control Information Compression	2	variable	See subclause 6.5.1				

NOTE: The current version of SNDCP is 10. This is also tThe default value for the version number is 0. It is assumed that the future versions are backward compatible with former ones.

3GPP TSG-CN1 Meeting #35
Sophia Antipolis, France, 16-20 August 2004

Tdoc N1-041636
(rev of Tdoc N1-041618)

CR-Form-v7.1

CHANGE REQUEST

⌘ **44.065** **CR** **017** ⌘ **rev** **2** ⌘ Current version: **6.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Negotiation of compression entities with unknown algorithm type
Source:	⌘	Siemens AG
Work item code:	⌘	TEI4
Date:	⌘	03.08.2004
Category:	⌘	**A**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change: ⌘ Ambiguity in the specification of the negotiation of compression entities:

According to subclauses 6.8.1 and 6.8.2:
"If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity."
and
"If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator."

I.e. an SNDCP entity can accept a compression entity proposed by its peer SNDCP entity by not including it in the responding SNDCP XID block.

On the other hand, according to subclause 6.8.3, exception handling:
"In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.
If the originating SNDCP XID block includes a parameter with **unrecognised Type field**, the parameter shall be ignored by the responder."

Some manufacturers have interpreted this requirement in such a way that it applies also to the parameter "algorithm type" in the SNDCP XID parameters "protocol control information compression" and "data compression".

	With such an interpretation, however, in some scenarios the SNDCP entity originating the XID negotiation cannot decide whether the peer SNDCP entity did not respond to a proposed compression entity in order to accept it or because it did not know the algorithm and therefore ignored the proposed compression entity. Example: Some R97/98 MS implementations ignore a proposal for a protocol control information compression with algorithm type RFC2507, because this algorithm was only introduced in R99. Since the MS does not explicitly reject the proposed compression entity for RFC2507, a R99 SGSN following subclauses 6.8.1 and 6.8.2 will assume that the MS has accepted the proposal and will start using this compression entity. The MS will then discard the downlink packets compressed by the SGSN with RFC2507. For the algorithm RFC2507 which was introduced in R99, the SNDCP entity originating the XID negotiation can resolve this ambiguity only by evaluating the revision level of its peer entity (revision level in the MS network capability IE or SGSNR bit in the BCCH or PBCCH, respecitvely). In order to remove this ambiguity for algorithms introduced in Rel-4 or later, the following is proposed:
Summary of change: ⌘	1) It is clarified that in the "protocol control information compression" field and "data compression" field the two octets following the PCOMP or DCOMP values always contain the "Applicable NSAPIs" parameter. 2) The parameter "algorithm type" is renamed to "algorithm identifier" so that 6.8.3 is no longer applicable to unknown compression algorithms. 3) It is clarified that if an SNDCP entity in the **MS** receives a proposal for a compression entity containing an unknown algorithm, it **should** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0. From Rel-5 onwards, with SNDCP version 1 this behaviour is mandatory. If an SNDCP entity in the **SGSN** receives a proposal for a compression entity containing an unknown algorithm, it **shall** reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0, from Rel-4 onwards . 4) From Rel-5 onwards, the SDNCP version number is raised to 1 so that the entity originating the XID negotiation can determine whether its peer entity is a R99 implementation or a 'Rel-4 or later' implementation. 5) If an SNDCP entity in Rel-4 wants to accept a proposal for a data compression entity with algorithm V.44, it shall include in the responding SNDCP XID block at least the corresponding compression entity number. 6) The peer entity of the XID negotiation shall accept only one header compression algorithm and one data compression algorithm for an NSAPI.
Consequences if not approved: ⌘	It will not be possible to use the compression algorithms introduced after R99 (V.44 and ROHC), since it will not be possible for the entity originating the XID negotiation to determine whether its peer entity is a legacy R99 implementation or a 'Rel-4 or later' implementation. Consequently the entity originating the XID negotiation will not be able to decide whether the peer SNDCP entity did not respond to a proposed compression entity for V.44 or ROHC in order to accept it or because the peer entity did not know the algorithm and therefore ignored the proposed compression entity.

Clauses affected: ⌘	6.5.1.1, 6.5.1.1.4, 6.5.2.1, 6.5.3.1, 6.5.4.1, 6.6.1.1, 6.6.1.1.4, 6.6.2.1, 6.6.3.1,

		6.8.1, 6.8.1a (new), 6.8.3, 8			
		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.5 Protocol control information compression

Protocol control information compression is an optional SNDCP feature.

Negotiation of the supported algorithms and their parameters is carried out between MS and SGSN using the SNDCP XID parameters (see clause 8).

6.5.1 Negotiation of multiple protocol control information compression types

Each SNDCP entity that supports protocol control information compression shall be able to negotiate one or several protocol control information compression entities with the compression field format shown in figure 7. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one protocol control information compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

6.5.1.1 Format of the protocol control information compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier ~~type~~				
Octet 3	Length=n-3							
Octet 4	PCOMP1				PCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 7: Protocol control information compression field format for SNDCP XID negotiation

6.5.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.5.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of PCOMP values needed by the corresponding compression algorithm (e.g. PCOMP3 and PCOMP4 shall not be included if the number of PCOMP values needed by a compression algorithm is one or two). If an odd number of PCOMP values are used by a compression algorithm, then the last PCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.5.1.1.3 Entity number

The entity number shall be used to identify a protocol control information compression entity on a SAPI. The entity number shall be assigned using the following rules:

- The entity number shall be an integer from 0 to 31.
- The entity number shall be assigned independently on each of the SAPIs.
- An entity number shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, an unassigned entity number shall become selected. If there is no unassigned entity number left, the compression entity shall not be proposed.
- A selected entity number shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned entity number shall become unassigned when the corresponding compression entity is deleted as a result of an XID negotiation, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4) in which an entity number is currently selected:
  - If the selected entity number is included with the P bit set to 0 in the incoming SNDCP XID block, then it shall be assumed that the peer SNDCP entity agreed to the creation of the proposed entity but the response was lost. Therefore the selected entity number shall become assigned, any selected PCOMP and DCOMP values for the algorithm of the entity shall become assigned, and the compression entity shall be created, before the incoming SNDCP XID block is processed. After the incoming SNDCP XID block is processed, the compression entity shall be negotiated again if necessary, as defined in subclause 6.2.1.4.
  - Otherwise (i.e. if the selected entity number is not included, or is included with the P bit set to 1 in the incoming SNDCP XID block), the selected entity number shall become unassigned, and any selected PCOMP and DCOMP values for the algorithm of the entity shall become unassigned, before the incoming SNDCP XID block, if any, is processed. Following the collision resolution procedure, the originally-proposed compression entity shall be proposed again (i.e. the originally-proposed compression entity shall not be considered created even if the originally-selected entity number is proposed in the incoming SNDCP XID block) by sending the appropriate primitive (LL-ESTABLISH.request or LL-XID.request). The originally-selected entity number, PCOMP and DCOMP values shall be used for the compression entity being re-proposed if they are unassigned, otherwise a new entity number, PCOMP or DCOMP value shall be selected.
- In the case of a collision in which an entity number is currently assigned:
  - If the peer SNDCP entity proposes a new compression entity with the same entity number, then it shall be assumed that the peer SNDCP entity negotiated the deletion of the entity but the response was lost, and the entity number is being reused. Therefore the original compression entity shall be deleted, the entity number shall become unassigned, PCOMP and DCOMP values shall be unassigned if necessary (see subclause 6.5.1.1.5), and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned entity number is not included, or is included with the P bit set to 0 in the incoming SNDCP XID block), the usual rules regarding collision handling shall apply.
- In the case of a collision in which a PCOMP or DCOMP value is currently assigned to a compression algorithm:
  - If the peer SNDCP entity proposes a new compression entity with the same PCOMP or DCOMP assigned to a different algorithm, then it shall be assumed that the peer SNDCP entity negotiated the deletion of all entities using the algorithm to which the PCOMP or DCOMP value was assigned, but the response was lost, and the PCOMP or DCOMP value is being reused. Therefore, all compression entities using that algorithm shall be deleted, all corresponding entity numbers shall become unassigned, and all PCOMP or DCOMP values assigned to the algorithm shall become unassigned, and then the proposed compression entity shall be responded to as usual.
  - Otherwise (i.e. if the assigned PCOMP or DCOMP is not included, or is included and assigned to the same algorithm), the usual rules regarding collision handling shall apply.

6.5.1.1.4 Algorithm identifier~~type~~

Table 4 show the list of protocol control information compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 4 shall be updated.

Table 4: List of protocol control information compression algorithms supported by SNDCP

Compression algorithm	Algorithm identifier~~type~~ (Range 0 to 31)
RFC 1144	0
RFC 2507	1
ROHC	2
-	Other values Reserved

6.5.1.1.5 PCOMP

One or more PCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for protocol control information compression. Each of the assigned PCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the PCOMP values follows the following general rules:

- PCOMP shall be an integer from 0 to 15.
- PCOMP value 0 is reserved permanently for no compression.
- PCOMP shall be assigned independently on each of the SAPIs.
- An assigned PCOMP value applies to all NSAPIs mapped to the same SAPI.
- PCOMP values shall be assigned to compression algorithms, not to compression entities (i.e. the same PCOMP value(s) shall be used by different compression entities on the same SAPI using the same compression algorithm).
- A PCOMP value shall be in one of the three states: unassigned, selected, or assigned.
- When a new compression entity is to be proposed, and if PCOMP values have not yet been assigned to the corresponding compression algorithm, then the appropriate number of unassigned PCOMP values shall be selected. If there is not enough unassigned PCOMP values left, the compression entity shall not be proposed.
- A selected PCOMP value shall become assigned if the corresponding proposed compression entity is created as a result of the XID negotiation, otherwise it shall become unassigned.
- An assigned PCOMP value shall become unassigned when the corresponding compression algorithm is no longer in use by any compression entity, or upon the receipt of the LL-RESET.indication primitive.
- In the case of a collision (see subclause 6.2.1.4), the handling of PCOMP values shall be in accordance with subclause 6.5.1.1.3.

While transferring data, the compressed frame type for an N-PDU is conveyed in the PCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.5.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of protocol control information compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the protocol control information compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.5.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.5.2 TCP/IP header compression (RFC1144)

The protocol control information compression method is specific for each network layer protocol type. TCP/IP (IPv4) header compression is specified in RFC 1144 [9].

6.5.2.1 Parameters

Table 5 contains the parameters defined for a compression entity using TCP/IP header compression. They may be negotiated during SNDCP XID negotiation.

Table 5: RFC 1144 TCP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 1144	0	0, 2 or 3 if P bit is 0, 1, 3 or 4 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			S_0 - 1	bbbbbbbb	0 through 255	down	15

6.5.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.2.1.2 S_0

The number of state slots, as defined in [9]. The S_0 range is 1 through 256, with 16 as default value.

6.5.2.2 Assignment of PCOMP values

The underlying service shall be able to distinguish the three types of compressed N-PDUs (i.e. Type IP, Uncompressed TCP, and Compressed TCP), as defined in RFC 1144 [9]. These three N-PDU types are differentiated by using different PCOMP values.

Two PCOMP values shall be assigned to the TCP/IP header compression algorithm. PCOMP1 shall contain the PCOMP value for the frame type "Uncompressed TCP", and PCOMP2 shall contain the PCOMP value for the frame type "Compressed TCP".

The PCOMP value of 0 shall be used for the frame type "Type IP".

6.5.2.3 Error Recovery

When TCP/IP header compression is used with unacknowledged peer-to-peer LLC operation, the decompression entity shall be notified in case an N-PDU is dropped, so that error recovery procedure (see RFC 1144 [9]) can be invoked.

6.5.3 TCP/IP and UDP/IP header compression (RFC 2507)

Detailed operation of the RFC 2507 header compression for IPv4 and IPv6 is described in clause 3 of the IETF specification RFC 2507 [10].

6.5.3.1 Parameters

Table 6 contains the parameters defined for a compression entity using RFC2507 header compression. They may be negotiated during SNDCP XID negotiation.

Table 6: RFC 2507 TCP/IP and UDP/IP header compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
RFC 2507	1	0, 2, 4, 5, 6, 7 or 9 if P bit is 0, 3, 5, 7, 8, 9, 10 or 12 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			F_MAX_PERIOD	bbbbbbbb bbbbbbbb	1-65535	down	256
			F_MAX_TIME	bbbbbbbb	1-255	down	5
			MAX_HEADER	bbbbbbbb	60-255	down	168
			TCP_SPACE	bbbbbbbb	3-255	down	15
			NON_TCP_SPACE	bbbbbbbb bbbbbbbb	3-65535	down	15

The explanation of the individual parameters can be found in the clause 14 of the IETF specification RFC 2507 [10].

6.5.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.3.2 Assignment of PCOMP values for RFC2507

The following PCOMP values shall be assigned to the RFC 2507 header compression. The PCOMP value 0 shall be used for regular IPv4 and IPv6 packets.

Table 7: PCOMP values assigned to RFC 2507 header compression algorithm

PID value	Packet type
PCOMP1	Full header
PCOMP2	Compressed TCP
PCOMP3	Compressed TCP non-delta
PCOMP4	Compressed non-TCP
PCOMP5	Context state

6.5.3.3 Error Recovery

The mechanisms related to error recovery and packet reordering are described in clauses 10 and 11 of the RFC 2507[10].

6.5.4 Robust Header Compression (ROHC)

Robust Header Compression (ROHC) is a framework for header compression, on top of which compression schemes can be defined for the compression of various protocol headers. Both the SNDCP ROHC negotiation mechanisms and the SN-PDU ROHC identifiers are generally defined for the ROHC framework, and therefore capable of handling both existing and future ROHC compression protocols (profiles). RFC 3095 [12] defines the ROHC framework, as well as the compression schemes and profiles for RTP/UDP/IP, UDP/IP, ESP/IP and uncompressed.

6.5.4.1 Parameters

Table 10 contains the parameters defined for a compression entity using ROHC. They may be negotiated during SNDCP XID negotiation.

Table 10: Robust Header Compression (ROHC) parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
ROHC	2	0, 2, 4, 6, 8, 8+n*2 if P bit is 0, 1, 3, 5, 7, 9, 9+n*2 if P bit is 1. (where n is the number of profiles, the max. number of profiles is 16)	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			MAX_CID	00bbbbbb bbbbbbbb	0-16383	down	15
			MAX_HEADER	00000000 bbbbbbbb	60-255	down	168
			PROFILE 1	bbbbbbbb bbbbbbbb	0-65535	(see 6.5.4.1.5)	0
			PROFILE 2	bbbbbbbb bbbbbbbb	0-65535	(see 6.5.4.1.5)	0
			...	...	...	...	...
			PROFILE 16	bbbbbbbb bbbbbbbb	0-65535	(see 6.5.4.1.5)	0
NOTE: ROHC parameter MRRU is not negotiated and shall be set to 0, i.e. ROHC segmentation shall not be used.							

6.5.4.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.5.4.1.2 MAX_CID

The MAX_CID parameter indicates the maximum context ID number the compressor is allowed to use. A MAX_CID value of N means CID values of 0 through N are valid. Thus, the number of contexts allowed is N+1, e.g. MAX_CID of 15 means 16 contexts are allowed.

6.5.4.1.3 MAX_HEADER

The MAX_HEADER parameter indicates the maximum number of octets of the protocol control information that may be compressed.

6.5.4.1.4 (Void)

6.5.4.1.5 PROFILE

The PROFILE parameter indicates the profile identifier. A list of up to 16 PROFILEs, indicating which ROHC profiles [14] are supported may be included. The negotiated list which is used for compression consists of the list of profiles supported by both peer entities, reduced to include at most ONE profile identifier with the same 8-bit LSB part. If both peer entities support more than one profile with the same 8-bit LSB part in its profile identifier, the set of these profiles shall be reduced to the profile with the highest MSB-value in its profile identifier.

Note: The reason for this is that the 8-bit MSB part of the profile identifier indicates the "variant" of the profile, and since only the 8-bit LSB part is sent in compressed headers, the set of available profiles must not include two profiles with the same 8-bit LSB part of the profile identifier.

6.5.4.2 Assignment of PCOMP values for ROHC

As opposed to other header compression schemes, the whole ROHC framework has only one packet type that has to be identified by the PDU format, and this packet type can be used by any ROHC compression profile. However, ROHC has two different context identification (CID) sizes. To avoid having to negotiate and potentially re-negotiate CID size, the mechanism from ROHC-over-PPP [13] is adopted in SNDCP, i.e. as shown in table 9, two packet types are defined for ROHC, one for small and one for large CIDs.

This implies that all CIDs within one ROHC packet shall be of the same size as indicated by the PID value, either small or large. In particular, embedded feedback shall have a CID of the same size as indicated by the PID value. For piggybacking feedback, a compressor must be able to control the feedback CID size used by the associated decompressor, ensure that all CIDs are of the same size, and indicate this size with the appropriate PID value. To make CID interpretation unambiguous when ROHC segmentation is used, all packets that contribute to a segment shall be sent with the same PID value, either PCOMP1 or PCOMP2, which then also applies to the CID size in the reconstructed unit. A unit reconstructed out of packets with PID values that differ shall be discarded.

Table 9: PCOMP values assigned to Robust Header Compression (ROHC)

PID value	Packet type
PCOMP1	ROHC small-CIDs
PCOMP2	ROHC large-CIDs

6.5.4.3 Error Recovery and other feedback

ROHC has built-in robustness mechanisms to avoid error events, as well as error recovery mechanisms using decompressor to compressor feedback. Such ROHC feedback is carried according to alternative 6) in section 5.2.1 of RFC 3095 [12].

6.6 Data compression

Data compression is an optional SNDCP feature. Data compression applies to both SN-DATA and SN-UNITDATA primitives.

Data compression, if used, shall be performed on the entire N-PDU, including the possibly compressed protocol control information.

Figure 8 shows an example how the SNDCP functions may be used. Several NSAPIs may use a common data compression entity, i.e. the same compression algorithm and the same dictionary. Separate data compression entities shall be used for acknowledged (SN-DATA) and unacknowledged (SN-UNITDATA) data transfer. Several NSAPIs may be associated with one SAPI, i.e. they may use the same QoS profile.

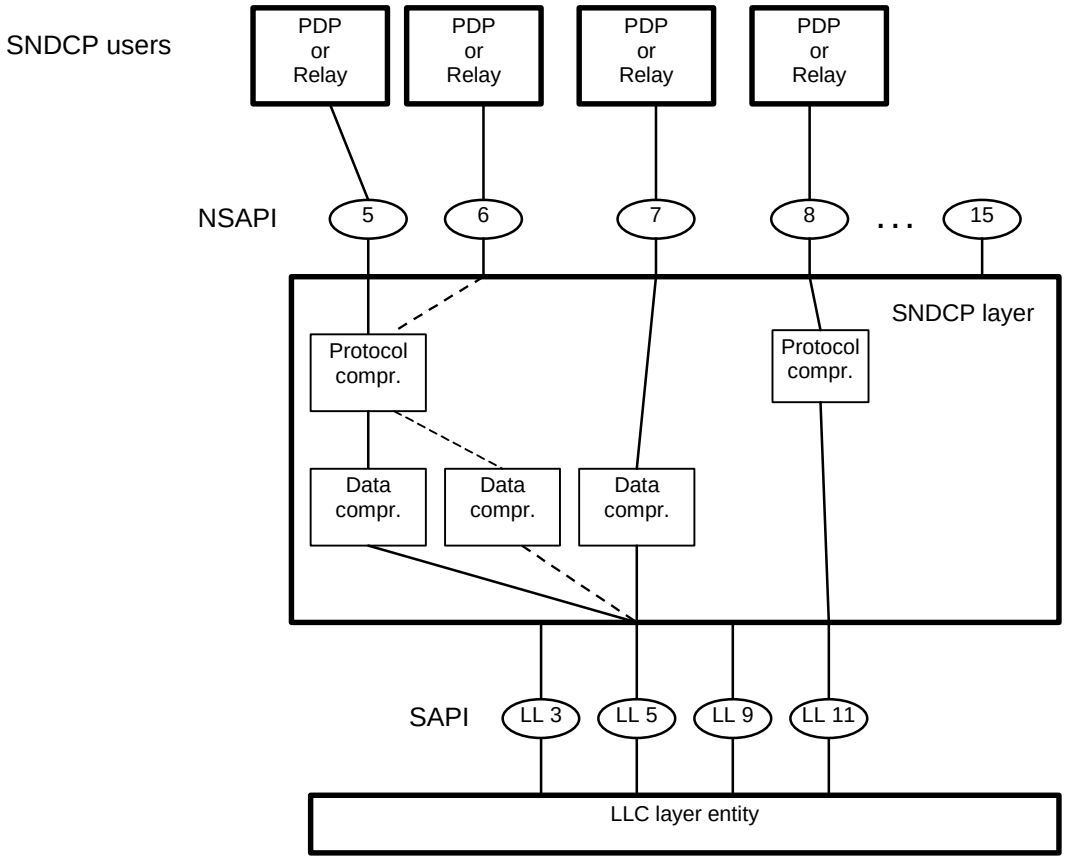

Figure 8: An example for the usage of NSAPIs, SNDCP functions, and SAPIs

6.6.1 Negotiation of multiple data compression types

Each SNDCP entity that supports data compression shall be able to negotiate one or several data compression entities with the compression field format shown in figure 9. The negotiation shall be carried out using the XID parameter negotiation specified in subclause 6.8. The initiating entity defines a set of requested compression entities, together with the algorithm and parameters for each compression entity. The set of entities and their algorithms and parameters shall be transmitted to the peer entity. The peer entity responds with the set of negotiated entities and their algorithms and parameters. The peer entity shall select the proposed parameter values or other appropriate values for the negotiated entities. If more than one data compression algorithm for a specific NSAPI is proposed during the XID negotiation then the receiving peer entity shall only choose one algorithm for that NSAPI.

6.6.1.1 Format of the data compression field

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier~~type~~				
Octet 3	Length=n-3							
Octet 4	DCOMP1				DCOMP2			
…	…				…			
Octet x	High-order octet							
…	…							
Octet n	Low-order octet							

NOTE: The two octets x and x+1, if included, contain the "Applicable NAPIs" parameter (see subclause 7.1.3).

Figure 9: Data compression field format for SNDCP XID negotiation

6.6.1.1.1 Spare bit (X)

The X bit shall be set to 0 by the transmitting SNDCP entity and shall be ignored by the receiving SNDCP entity.

6.6.1.1.2 Propose bit (P)

The P bit shall be set to 1 if a new compression entity is being proposed, otherwise it shall be set to 0. If the P bit is set to 1, then all octets shall be included, otherwise octet 2 and octets 4 to x-1 shall not be included. If the P bit is set to 1, then only enough number of octets shall be included to contain the number of DCOMP values needed by the corresponding compression algorithm (e.g. DCOMP3 and DCOMP4 shall not be included if the number of DCOMP values needed by a compression algorithm is one or two). If an odd number of DCOMP values are used by a compression algorithm, then the last DCOMP value shall be set to 0 in the compression field by the transmitting SNDCP entity, and it shall be ignored by the receiving SNDCP entity.

6.6.1.1.3 Entity number

The entity number shall be used to identify a data compression entity on a SAPI. See subclause 6.5.1.1.3 for the rules for assigning entity numbers. The assignment of entity numbers for protocol control information compression entities and data compression entities shall be independent.

6.6.1.1.4 Algorithm identifier~~type~~

Table 6a shows the list of data compression algorithms supported by the SNDCP layer. When new compression algorithms are needed for SNDCP, table 6a shall be updated.

Table 6a: List of data compression algorithms supported by SNDCP

Data compression algorithm	Algorithm identifier~~type~~ (Range 0-31)
V.42 bis	0
V.44	1
-	Other values Reserved

6.6.1.1.5 DCOMP

One or more DCOMP values shall be assigned dynamically to a compression algorithm, based on the negotiation of the XID parameters for data compression. Each of the assigned DCOMP values denotes one compressed frame type of that compression algorithm.

The assignment of the DCOMP values shall follow the rules for the assignment of PCOMP values in subclause 6.5.1.1.5.

While transferring data, the compressed frame type for an N-PDU is conveyed in the DCOMP field of the SNDCP header of the first SN-PDU belonging to the N-PDU. Any successfully negotiated algorithm may be used for compression of an N-PDU.

6.6.1.2 Resetting compression entities following SNDCP XID negotiation

The LL-Establish primitives shall be used for the negotiation of data compression if:

- one or more parameters, excluding the applicable NSAPIs, of existing compression entities used with acknowledged peer-to-peer LLC operation are changed by the originator of the negotiation; or
- one or more NSAPIs are removed, by the originator of the negotiation, from existing compression entities used with acknowledged peer-to-peer LLC operation, except when all NSAPIs using the compression entity are removed, or when LLC is already in ADM.

Otherwise, either the LL-Establish primitives or the LL-XID primitives may be used.

If the LL-XID primitives are used for XID negotiation, then in addition to restrictions specified elsewhere in the present document, the following parameters of the data compression entities are non-negotiable by the responding SNDCP entity:

- any parameter of existing compression entities used with acknowledged peer-to-peer LLC operation.

If one or more parameters, other than the applicable NSAPIs, of a compression entity used with unacknowledged peer-to-peer LLC operation are changed, the compression entity shall be reset locally upon completion of the SNDCP XID negotiation.

6.6.1.3 Parameters for compression entities

On negotiating a compression entity, not all the parameters of the entity have to be specified. If a parameter is to be included, all the preceding parameters shall also be specified, and the length field shall be set to the sum of the lengths of all the parameters specified. If any of the parameters is not specified, the rules in subclause 6.8.2 shall apply.

6.6.2 Management of V.42 bis data compression

ITU-T Recommendation V.42 bis [8] data compression may be used with SN-DATA primitives and SN-UNITDATA primitives.

6.6.2.1 Parameters

Table 7a contains the parameters defined for a compression entity using ITU-T Recommendation V.42 bis data compression. They may be negotiated during SNDCP XID negotiation.

Table 7a: V.42 bis data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.42 bis	0	0, 2, 3, 5, or 6 if P bit is 0, 1, 3, 4, 6, or 7 if P bit is 1.	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_1	bbbbbbbb bbbbbbbb	512 through 65535	down	2048
			P_2	bbbbbbbb	6 through 250	down	20

6.6.2.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.2.1.2 P_0

Two bits are used to indicate the usage of compression, one bit for each direction.

00 compress neither direction.

01 compress MS-to-SGSN direction only.

10 compress SGSN-to-MS direction only.

11 compress both directions.

6.6.2.1.3 P_1

Maximum number of codewords in the compressor dictionary (see [8]).

6.6.2.1.4 P_2

Maximum number of characters in an uncompressed data string that is accepted to be encoded.

6.6.2.2 Assignment of DCOMP values

One DCOMP value shall be assigned (as DCOMP1) to the V.42 bis data compression algorithm.

6.6.2.3 Operation of V.42 bis data compression

When V.42 bis is used with SN-DATA primitives, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent.

When V.42 bis is used with SN-UNITDATA primitives, the compression entity shall be reset (using the C-INIT primitive defined in [8]) before an N-PDU is compressed or decompressed. After compression, the data in the compression entity shall be flushed (using the C-FLUSH primitive defined in [8]) and added to the compressed N-PDU before the compressed N-PDU is sent. The LLC protocol shall operate in the protected mode of operation.

When V.42 bis is used with SN-DATA primitives and an error is detected by the decoder, the SNDCP entity shall use LL-ESTABLISH.request primitive to reset the acknowledged peer-to-peer LLC operation for the SAPI used.

6.6.3 Management of V.44 data compression

ITU-T Recommendation V.44 data compression, as described in [11], may be used with SN-DATA primitives and SN-UNITDATA primitives. Annex B of ITU-T Recommendation V.44 describes two methods of implementation and operation of V.44 in packet networks: Packet Method and Multi-Packet Method. Multi-Packet Method is a superset of Packet Method and an MS or SGSN that supports Multi-Packet Method must also support Packet Method.

6.6.3.1 Parameters

Table 7c contains the parameters defined for a compression entity using V.44 data compression. They may be negotiated during SNDCP XID negotiation. During V.44 data compression negotiation, unless both the MS and SGSN support Multi-Packet Method, Packet Method is used. Parameter C_0 indicates support of Packet Method (10000000) or both methods (11000000).

NOTE 1: V.44 data compression negotiation is not required. If V.44 is selected and no compression parameters are specified, then Packet Method with defaults as defined in subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2, is used.

Table 7c: V.44 data compression parameters

			Parameters				
Algorithm Name	**Algorithm Identifier ~~Type~~**	**Length**	**Parameter Name**	**Format**	**Range**	**Sense of Negotiation**	**Default Value**
V.44	1	0, 2, 3, 4, 6, 8, 10, or 12 if P bit is 0, 1, 3, 4, 5, 7, 9, 11, or 13 if P bit is 1	Applicable NSAPIs	bbbbbbbb bbb00000	0, 32, 64, … , 65504	down (each bit separately)	0
			C_0	bb000000	10000000 or 11000000	11000000 down to 10000000	10000000
			P_0	000000bb	0 through 3	down (each direction separately)	3
			P_{1T}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.4
			P_{1R}	bbbbbbbb bbbbbbbb	256 through 65535	down	Refer to subclause 6.6.3.1.5
			P_{3T}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1T})$	down	$3 \times P_{1T}$
			P_{3R}	bbbbbbbb bbbbbbbb	$\geq (2 \times P_{1R})$	down	$3 \times P_{1R}$

NOTE 2: V.44 parameters P_{2T} and P_{2R} are set to 255 and not negotiated in packet networks.

6.6.3.1.1 Applicable NSAPIs

See subclause 7.1.3.

6.6.3.1.2 C_0

Two bits are used to indicate the V.44 method of operation supported (refer to [11] Annex B).

10 Packet Method supported.

11 Packet Method and Multi-Packet Method supported.

If parameter C_0 is not specified then Packet Method is selected with its default parameter values (refer to subclauses 6.6.3.1.4 and 6.6.3.1.5 and in [11] annex B, clause B.1.2).

********** NEXT MODIFIED SECTION **********

6.8 XID parameter negotiation

Negotiation of XID parameters between peer SNDCP entities may be carried out to ensure optimal information transfer. The parameters are called SNDCP exchange identity (XID) parameters.

SNDCP XID parameter negotiation may be initiated by the SNDCP entity at the MS or at the SGSN. If SNDCP XID parameters are to be changed, SNDCP XID negotiation shall be initiated prior to data transfer - the MS shall initiate SNDCP XID negotiation upon receipt of SNSM-ACTIVATE.indication; the SGSN shall initiate SNDCP XID negotiation upon receipt of the SNSM-MODIFY.indication primitive if an NSAPI has been put into use (in the case of an Inter-SGSN Routeing Area Update), or if the change in QoS profile to an existing NSAPI results in a change in compressor(s) used by the NSAPI.

When an NSAPI no longer uses a compression entity due to a PDP context deactivation or a PDP context modification, an SNDCP XID negotiation shall be performed to remove the NSAPI from the Applicable NSAPIs of the compression entity. The negotiation shall be initiated by the MS upon receipt of the SNSM-DEACTIVATE.indication with XID Negotiation Indicator in the case of explicit PDP context deactivation by peer-to-peer signalling between the SM entities. The negotiation shall be initiated by the SGSN upon receipt of the SNSM-MODIFY.indication in the case of PDP context modification. If a PDP context is deactivated locally, without peer-to-peer signalling between the SM entities, the SM entity at the MS and the SM entity at the SGSN each shall send an SNSM-DEACTIVATE.indication without XID Negotiation Indicator to its respective SNDCP entity. Upon receipt of this primitive without XID Negotiation Indicator the SNDCP entity shall remove the NSAPI from the Applicable NSAPIs of the compression entity without any XID negotiation.

The XID negotiation is a one-step procedure; i.e. the initiating end proposes parameter values, and the responding end either accepts these or offers different values in their place according to the XID negotiation rules described in the present document; the rules limit the range of parameter values as well as the sense of negotiation. The initiating end accepts (or rejects) the values in the response; this concludes the negotiation.

The block format for the SNDCP XID parameter negotiation is shown in figure 10. Not all parameters have to be included in the XID block, only parameters that are negotiated. Parameters may be included in any order. Also it shall be possible to negotiate parameters for more than one NSAPI in one XID block since more than one NSAPI can use the same SAPI.

Bit	8	7	6	5	4	3	2	1
Octet 1	Parameter type=0							
Octet 2	Length=1							
Octet 3	Version number							
Octet 4	Parameter type=1							
Octet 5	Length=n-5							
Octet 6	P	X	X	Entity number				
Octet 7 (optional)								
Octet 8	Length=k-8							
Octet 9 … (optional)								
Octet j	High-order octet							
…	…							
Octet k	Low-order octet							
Octet k+1	P	X	X	Entity number				
Octet k+2 (optional)								
Octet k+3	Length=m-(k+3)							
Octet k+4… (optional)								
Octet k+y	High-order octet							
…	…							
Octet m	Low-order octet							
…	…							
Octet n	Low-order octet							
Octet n+1	Parameter type=2							
Octet n+2	Length=r-(n+2)							
Octet n+3	P	X	X	Entity number				
Octet n+4 (optional)								
Octet n+5	Length=p-(n+5)							
Octet n+6… (optional)								
Octet n+w	High-order octet							
…	…							
Octet p	Low-order octet							
Octet p+1	P	X	X	Entity number				
Octet p+2 (optional)								
Octet p+3	Length=q-(p+3)							
Octet p+4… (optional)								
Octet p+v	High-order octet							
…	…							
Octet q	Low-order octet							
…	…							
Octet r	Low-order octet							

Figure 10: Example of SNDCP XID block format

The SNDCP user uses SN-XID.request to initiate the negotiation of the XID parameters. The SNDCP entity sends the proposed SNDCP XID parameters to the LLC SAP with the LL-XID.request or LL-ESTABLISH.request. The LLC SAP shall issue an XID command containing the SNDCP XID parameters (see 3GPP TS 44.064 [6]). The peer LLC SAP shall, upon receipt of the XID command, indicate the SNDCP XID parameters to SNDCP entity using LL-XID.indication or LL-ESTABLISH.indication. The peer SNDCP entity shall select appropriate values for the proposed parameters or negotiate the appropriate values with the SNDCP user entity with the SN-XID.indication and SN-XID.response primitives. When the appropriate parameter values are known by the peer SNDCP entity, it shall use the LL-XID.response or LL-ESTABLISH.response primitive to continue negotiation. Upon reception of the response, the LLC SAP shall send the received parameters to the SNDCP entity using the LL-XID.confirm or LL-ESTABLISH.confirm primitive. The SNDCP entity delivers the negotiated parameters to the SNDCP user. This is illustrated in figure 11. The originator of the negotiation shall apply the new parameter values after it has received the 'confirm' primitive. The responding end of the negotiation shall apply the new parameter values after it has sent the replying 'response' primitive.

Following the sending of the LL-XID.request primitive, the SNDCP layer shall suspend the transfer of SN-DATA and SN-UNITDATA primitives to the LLC SAP to which the LL-XID.request is sent. Transfer of SN-DATA and SN-UNITDATA primitives shall resume when the SNDCP XID negotiation ends through one of the following means:

- successful (receiving LL-XID.confirm);
- failure (receiving LL-RELEASE.indication, or LL-STATUS.indication); or
- successful following collision resolution (receiving LL-ESTABLISH.indication and sending LL-ESTABLISH.response, or receiving LL-XID.indication and sending LL-XID.response, see subclause 6.2.1.4).

LLC may also initiate LLC XID negotiation, in which case LLC may send an LL-XID.indication to inform SNDCP the values of N201-I and N201-U. This is illustrated in figure 12. If the SNDCP entity receives an LL-XID.indication without an SNDCP XID block, it shall not respond with the LL-XID.response primitive.

Negotiation of SNDCP version number is always between the peer SNDCP entities. The version number is not known by the SNDCP user. However, negotiation of the parameters for compression algorithms may be carried out between the SNDCP user entities.

Negotiation of SNDCP XID parameters for an NSAPI shall be carried out in the SAPI to which the NSAPI is mapped.

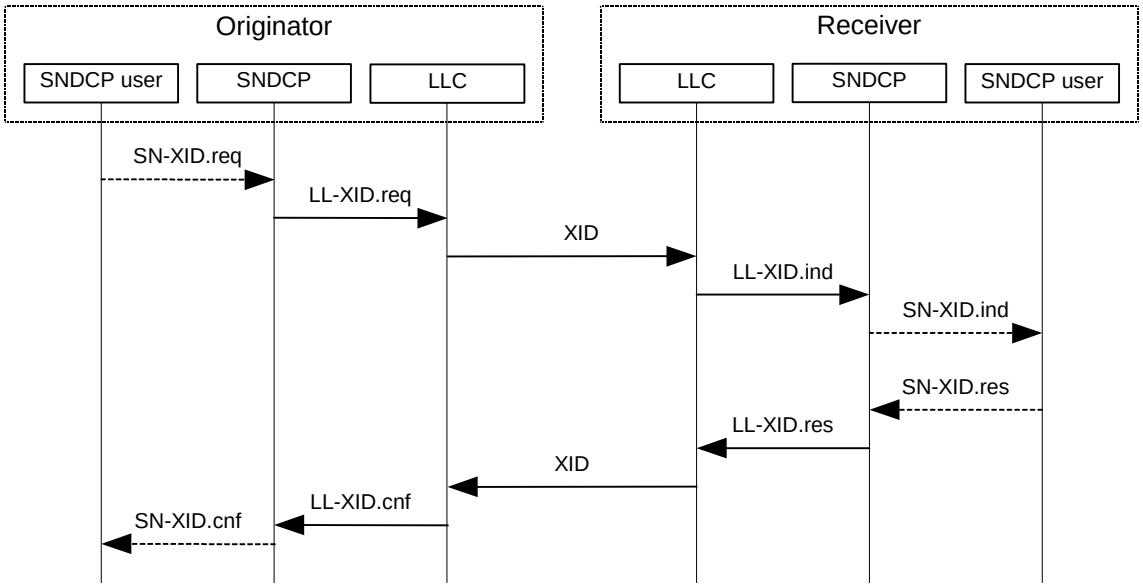

Figure 11: SNDCP XID negotiation procedure

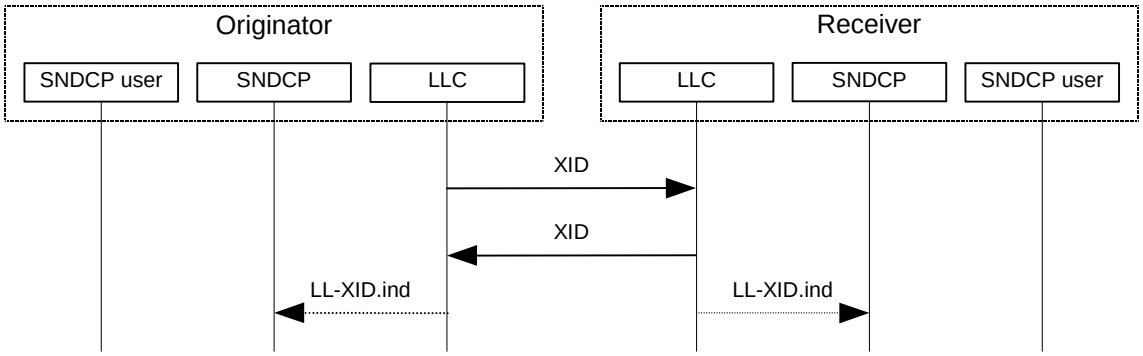

Figure 12: LLC XID negotiation procedure

6.8.1 Negotiation of compression entities

For parameter type 1 and 2, multiple compression fields (as shown in figure 7 and figure 9) may be specified. Each compression field corresponds to a compression entity.

In each compression field, the "Applicable NSAPIs" parameter indicates the NSAPIs that uses the compression entity. The parameter, if included, shall consist of 2 octets. Multiple NSAPIs may share the same compression entity by setting multiple bits in the parameter. NSAPIs requiring acknowledged peer-to-peer LLC operation and unacknowledged peer-to-peer LLC operation shall not share the same compressor (see subclause 6.10).

During SNDCP XID negotiation or re-negotiation, if a parameter type is specified in the SNDCP XID block, compression entities currently in use and compression entities proposed to be added may be included in the SNDCP XID block. Not all entities need to be included in the SNDCP XID block. If a compression entity is not included, the value of its parameters shall be determined by the rules defined in subclause 6.8.2.

If, implicitly or explicitly (see subclause 6.8.2), a compression entity is specified in the responding SNDCP XID block with one or more bits set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be created (if it does not exist yet).

If, implicitly or explicitly, a compression entity is specified in the responding SNDCP XID block with no bit set to 1 in the "Applicable NSAPIs" parameter, the compression entity shall be deleted (if it currently exists).

If an SNDCP entity receives a proposal for a compression entity containing an unknown algorithm, it shall reject this compression entity explicitly by setting the bits in the "Applicable NSAPIs" parameter in the responding SNDCP XID block to 0.

NOTE: Some legacy MS implementations of SDNCP version 0 will ignore a proposed compression entity containing an unknown algorithm and will not reply with an explicit rejection.

If, implicitly or explicitly, one or more bits are set to 1 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall start using (or continue to use) the compression entity.

If, implicitly or explicitly, one or more bits are set to 0 in the "Applicable NSAPIs" parameter of a compression entity in the responding SNDCP XID block, the NSAPIs corresponding to these bits shall release the compression entity (if they have been using the compression entity).

6.8.1a Negotiation of SNDCP version

An SNDCP entity supporting this version of SNDCP shall set the version number in the originating SNDCP XID block to 1. For the sense of negotiation of the version number see clause 8, table 8.

6.8.2 Values of SNDCP XID parameters

In this subclause, the term "parameter" refers to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If an SNDCP XID parameter has not been negotiated, default values shall apply. The default value for a compression field (entity) is "non-existing".

If the originating SNDCP XID block does not include a parameter (implicit command), it shall be treated as equivalent to requesting for the current value for the parameter. The responder may explicitly include this parameter in its response. If the responder explicitly includes the parameter in the response, then it shall also explicitly include this parameter in every SNDCP XID response until the parameter has been explicitly negotiated, either by responding to an SNDCP XID command that included the parameter, or by explicitly including the parameter the next time an SNDCP XID command is transmitted.

If a parameter is included in the originating SNDCP XID block and the responder does not include the parameter in its response (implicit response), it shall be treated as equivalent to responding with the value proposed by the originator.

If both the originator and the responder do not include a parameter in the negotiation, the value of the parameter is not changed.

6.8.3 Exception handling

In this subclause, the term "parameter" may refer, wherever applicable, to an SNDCP XID parameter, a compression field (for parameter type 1 or 2), or a parameter for a compression field.

If the originating SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (as specified in clause 8, table 8), the parameter shall be ignored by the responder.

If the originating SNDCP XID block includes a parameter with unsupported length or an out-of-range value, then the responder shall respond to the parameter with lengths and values set according to the responder's preference.

If the originating SNDCP XID block includes parameter type 1 or 2 which violates the rules in subclause 6.8.1, the responder shall treat the parameter as not transmitted by the originator, and responds according to subclause 6.8.2.

If the originating SNDCP XID block includes a parameter with duplicated instances, the subsequent instances of the duplicated parameter shall be ignored.

If the originating SNDCP XID block is sent on LL-XID primitives and contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation, then the responder shall respond with these parameters set to their previously-negotiated values.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, when an assigned entity number is included with the P bit set to 1, the algorithm and the PCOMP and DCOMP fields shall be ignored if they are the same as the previously-assigned values. If the algorithm and PCOMP or DCOMP fields are not the same as the previously-assigned values, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if an unassigned entity number is included with the P bit set to 0, then the Applicable NSAPIs field in the response shall be set to 0.

In the originating SNDCP XID block, excluding the collision scenarios described in subclause 6.5.1.1.3, if one or more of the PCOMP or DCOMP specified is already assigned to a different compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

In the originating SNDCP XID block, if one or more new PCOMP or DCOMP values are specified for an existing compression algorithm, then the Applicable NSAPIs field of the compression field in question shall be set to 0 in the response, and an SNSM-STATUS.request primitive with Cause "invalid XID command" shall be sent to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the responding SNDCP XID block includes an SNDCP XID parameter with unrecognised Parameter Type field (see clause 8, table 8), unsupported length, an out-of-range value or a value violating the sense of negotiation, a parameter type 1 or 2 which violates the rules in subclause 6.8.1, a parameter with duplicated instances, contains prohibited changes (see subclauses 6.5.1.2 and 6.6.1.2) to the parameters of compression entities used with acknowledged peer-to-peer LLC operation when the SNDCP XID block is sent on LL-XID primitives, or a compression field with the P bit set to 1, then the originator shall ignore the block and reinitiate the negotiation. If the renegotiation fails for an

implementation-specific number of times, the originating SNDCP layer shall send an SNSM-STATUS.request primitive with Cause "invalid XID response" to the SM sub-layer. SM shall then deactivate all PDP contexts for this SAPI.

If the LLC layer indicates that the XID parameter negotiation failed, by sending an LL-RELEASE.indication with Cause "no peer response" or an LL-STATUS.indication with Cause "no peer response", then, as an implementation option, the SNDCP layer may wait for an implementation-specific amount of time and re-invoke the XID negotiation procedure.

********** NEXT MODIFIED SECTION **********

8 SNDCP XID parameters

The SNDCP XID parameters are shown in table 8:

Table 8: SNDCP XID parameters

Parameter name	Parameter Type	Length	Format	Range	Default value	Units	Sense of negotiation
Version number	0	1	0000bbbb	0-15	0	-	down
Data Compression	1	variable	See subclause 6.6.1				
Protocol Control Information Compression	2	variable	See subclause 6.5.1				

NOTE: The current version of SNDCP is 1~~0~~. ~~This is also t~~The default value for the version number is 0. ~~It is assumed that the future versions are backward compatible with former ones.~~