Presentation of Specification to TSG or WG

Presentation to:	**TSG-RAN Meeting #15**
Document for presentation:	**TR 25.860 Version 1.0.0**
Presented for:	**Information**

Abstract of document:

This technical report captures the results of the Rel-5 work on the work item “Radio access bearer support enhancement”. The TR details the solution made so far on RFC3095 context relocation.

Changes since last presentation to TSG-RAN Meeting:

This is the first presentation of the TR to RAN.

Outstanding Issues:

The outstanding issues and progress made is fully captured in the TR.

Contentious Issues:

During RAN WG2#27 there was no time to discuss the issues raised in R2-020377. The final details of the solution will be discussed in RAN WG2#28 and #29.

3GPP TR 25.860 V1.0.0 (2002-03)

Technical Report

3rd Generation Partnership Project;
Technical Specification Group Radio Access Network;
Radio Access Bearer Support Enhancements
(Release 5)

Keywords

UMTS, radio, bearer

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document is a technical report that summarises the work on the UTRAN Release 5 work item "Radio Access Bearer Support Enhancements". The work items comprises of two areas of study:

1) Robust header compression

2) SRNS context relocation

Each study area includes the requirements of the proposed feature, a description of the basic mechanism and a discussion of the issues involved. A recommended solution is provided and impacts to other RAN WGs are analysed.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 25.322: "RLC Protocol Specification".

[2] 3GPP TS 25.323: "PDCP Protocol Specification".

[3] 3GPP TS 25.331: "Radio Resource Control (RRC); protocol specification".

[4] 3GPP TS 25.413: "UTRAN Iu Interface: RANAP Signalling".

[5] 3GPP TS 25.415: "UTRAN Iu Interface: CN-RAN User Plane Protocol".

[6] IETF RFC 3095: "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed".

[7] IETF RFC 1144, February 1990: "Compressing TCP/IP Headers for Low-Speed Serial Links".

[8] IETF RFC 2507, February 1999: "IP Header Compression".

[9] IETF RFC 2508, February 1999: "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links".

[10] IETF RFC 2509, February 1999: "IP Header Compression over PPP".

[11] 3GPP TS 23.060: "General Packet Radio Service (GPRS); Service description; Stage 2".

[12] 3GPP TS 25.303: "Interlayer Procedures in Connected Mode".

[13] Internet-Draft (work in progress), 7 February 2001, <draft-ietf-rohc-rtp-lower-layer-guidlines-01.txt>: "Lower Layer Guidelines for Robust RTP/UDP/IP Header Compression".

http://www.ietf.org/internet-drafts/ draft-ietf-rohc-rtp-lower-layer-guidlines-01.txt

[14] IETF RFC 3096: "Requirements for robust IP/UDP/RTP header compression".

[15] M. Degermark, H. Hannu, L.E. Jonsson, K. Svanbro, "Evaluation of CRTP Performance over Cellular Radio Networks", IEEE Personal Communication Magazine, Volume 7, number 4, Aug. 2000 pp. 20-25

[16] 3GPP TS 43.051: "GERAN Overall Description – Stage 2".

3 Abbreviations

3.1 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AS	Access Stratum
CID	Context ID
ESP	Encapsulating Security Payload Header
GERAN	GSM/EDGE Radio Access Network
HC	Header Compression
ID	Identifier
IE	Information Element
IETF	Internet Engineering Task Force
IP	Internet Protocol
PID	Packet ID
RNC	Radio Network Controller
ROHC	Robust Header Compression
RTP	Realtime Transmission Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network

4 Requirements

Void.

5 Study Areas

5.1 Robust header compression

5.1.1 Introduction

Header compression of IP flows is possible due to the fact that the fields in the headers of IP packets are either constant or changing in a known pattern between consecutive packets in the same flow. It is possible to send only information regarding the nature of the changing fields of the headers with respect to the reference packet in the same IP flow.

The benefit is a significant reduction in header overhead and hence an increase in bandwidth efficiency. For example, IP based voice applications require an IP header; 20 octets for IPv4 and 40 octets for IPv6, a UDP header of 8 octets and a RTP header of 12 octets. A total of 40 octets for the headers are required to transport the voice payload for IPv4 and 60 octets for IPv6. When this is compared to the size of the payload, which is of the order of 15-32 bytes (depending on the codec and frame size/rate), the gains from compressing the headers is quite apparent.

In order for header compression to work there must be a compressor and a decompressor. A context is basically a snapshot of the complete (uncompressed) headers of an IP flow. This context is always exchanged from the compressor to decompressor at initialisation of the header compression scheme. After that the context is updated according to some criteria that is dependent on the header compression scheme.

During normal operation, the compressor will always try to send compressed headers instead of full headers. The compressed header represents the relative changes to a reference packet in the same IP flow and therefore the changes are relatively small.

Some header compression schemes may employ feedback from the decompressor to the compressor to indicate the current context state in the decompressor. A result of this could be to send sufficient information to update the context in the decompressor. With the basic tools of header compression it is possible to define a protocol that will work on any link layer technology.

5.1.2 Existing header compression schemes

The former header compression schemes that are standardised in the IETF are specified in references [7], [8] and [9]. However, these schemes were not designed for cellular usage and especially [9] that compress real-time IP headers do not cope well over unreliable links such as the cellular environment. Also, wireless links exhibit long round trip times (RTT) and therefore loss of synchronisation of contexts between the compressor and decompressor can result in a large loss of packets until synchronisation is achieved [15].

As IP based multimedia services are increasing rapidly, a need has arisen to support real-time IP services in UTRAN. However, with the added difficulties due the radio interface as described earlier there is a need for header compression to be robust in the cellular environment.

5.1.3 IETF Robust header compression working group

It is the task of the IETF WG called "Robust Header Compression" or ROHC to standardise a header compression protocol that is suitable for wireless links. A robust scheme should tolerate errors on the link over which header compression takes place (including both frame losses and residual bit errors) without losing additional packets, introducing additional errors or using more bandwidth.

The ROHC protocol is currently the only protocol that is being standardised by the ROHC WG. ROHC framework handles several compression profiles. Currently it contains profiles that are able to compress RTP/UDP/IP, UDP/IP and ESP/IP streams for both IPv4 and IPv6. This study area in this technical report will investigate how the ROHC protocol will be realised for Release 4.

5.1.4 ROHC compression and decompression states

The compressor starts in the lowest compression state and gradually transitions to higher compression states. The general principle is the compressor will always operate in the highest possible compression state, under the constraint that the compressor has sufficient confidence that the decompressor has the information necessary to decompress a compressed header.

In the reliable mode, that confidence comes from receipt of ACKs from the decompressor. Otherwise, that confidence comes from sending the information a certain number of times, utilising a CRC calculated over the uncompressed RTP/UDP/IP header, and from not receiving NACKs (negative acknowledgements).

The compressor may also transition back to a lower compression state when necessary.

For IP/UDP/RTP, IP/UDP and ESP/IP compression profiles, the three compressor states are:

- Initialisation/Refresh (IR);
- First Order (FO);
- Second Order (SO).

5.1.4.1 IR State

The purpose of this state is to set up or refresh the context between the compressor and decompressor. The information that is sent from the compressor may contain static and non-static fields in uncompressed form (full refresh), or just non-static fields in uncompressed form (dynamic refresh).

The compressor enters this state at initialisation, upon request from decompressor, or upon Refresh Time-out. The compressor leaves the IR state when it is confident that the decompressor has correctly received the refresh information.

5.1.4.2 FO State

The compressor operates in the FO state when the header stream does not conform to a uniform pattern (i.e. constant changes), or when the compressor is not confident that the decompressor has acquired the parameters of the uniform pattern. The compressor will leave this state and transition to the SO state when the header conforms to a uniform pattern and when the compressor is sufficiently confident that previous non-uniform changes have reached the decompressor.

5.1.4.3 SO State

In this state the compressor is sufficiently confident that the decompressor has also acquired the parameters of the uniform pattern. In the SO state, the compressor sends headers, which mainly consist of a sequence number. While in the SO state, the decompressor does a simple extrapolation based on information it knows about the pattern of change of the header fields and the sequence number contained in the SO header in order to regenerate the uncompressed header. The compressor leaves this state to go back to FO state when the header no longer conforms to the uniform pattern or to IR state if counter so indicates in unidirectional mode.

5.1.5 Modes and mode transitions

There are three modes of operation, each with the three states as described in 5.1.4:

- Uni-directional;
- Bi-directional optimistic;
- Bi-directional reliable.

and the possible transitions are shown in Figure 1 below.

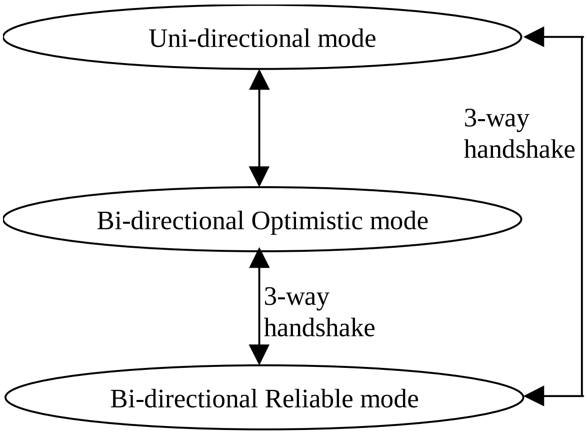

Figure 1: ROHC modes

Compression always starts in the unidirectional mode and transits to any of the bi-directional mode depending on feedback from the decompressor. The uni-directional mode implies that there is no feedback from the decompressor to the compressor while for the bi-directional optimistic there is irregular feedback and periodic feedback for the bi-directional reliable mode.

A brief description of the modes is given below:

- U-mode is used when a feedback channel is not present or undesirable to use, and it should be known that the robustness and efficiency can never be as good as with feedback (if the channel is not completely error-free).
- O-mode is aiming for highest compression efficiency while providing reasonable robustness.
- R-mode is almost completely robust but has slightly higher overhead and more feedback messages.

5.1.6 State transitions

The allowed state transitions are shown in Figure 2 and the rules and packets formats that are required are briefly described in the following subclause of this subclause. A more detailed description can be found in [6].

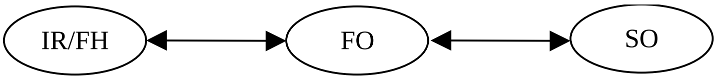

Figure 2: State transitions

5.1.6.1 Uni-directional mode

5.1.6.1.1 Compressor

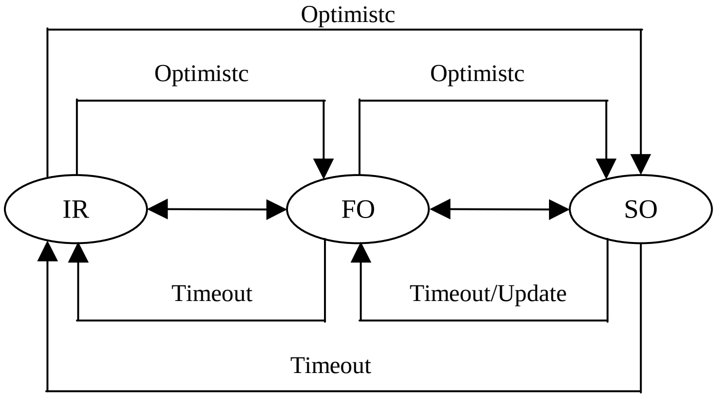

Figure 3: Uni-directional mode compressor logic

5.1.6.1.2 Decompressor

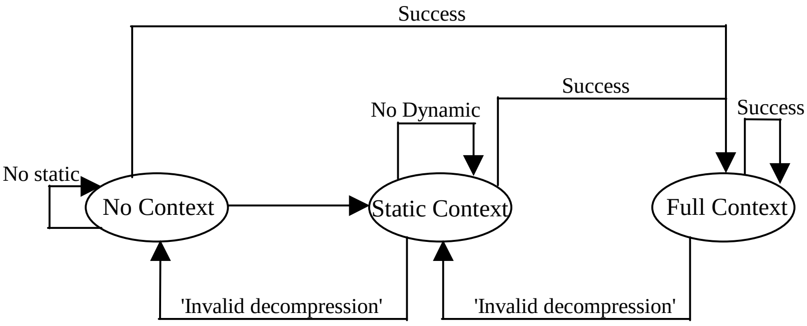

Figure 4: Uni-directional mode decompressor logic

5.1.6.2 Bi-directional optimistic

5.1.6.2.1 Compressor

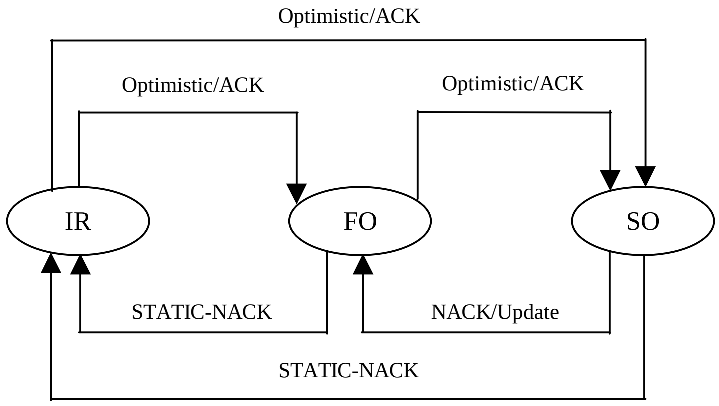

Figure 5: Optimistic mode compressor logic

5.1.6.2.2 Decompressor

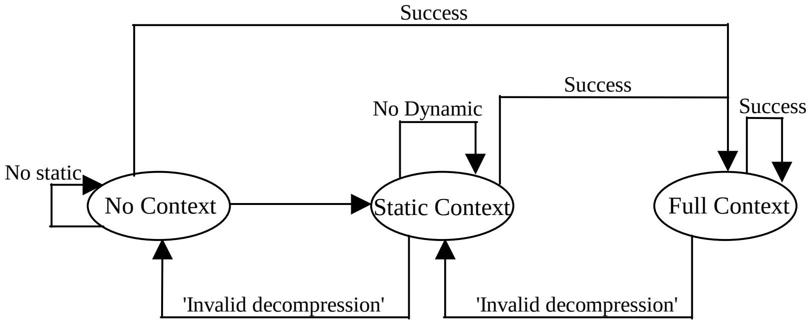

Figure 6: Bi-directional optimistic mode decompressor logic

5.1.6.3 Bi-directional reliable

5.1.6.3.1 Compressor

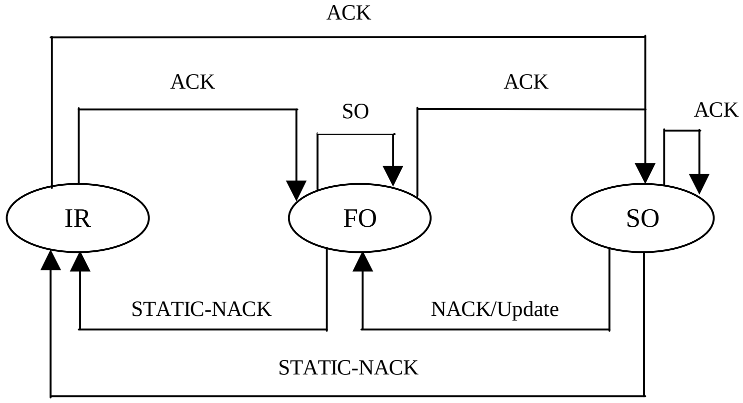

Figure 7: Reliable mode compressor logic

5.1.6.3.2 Decompressor

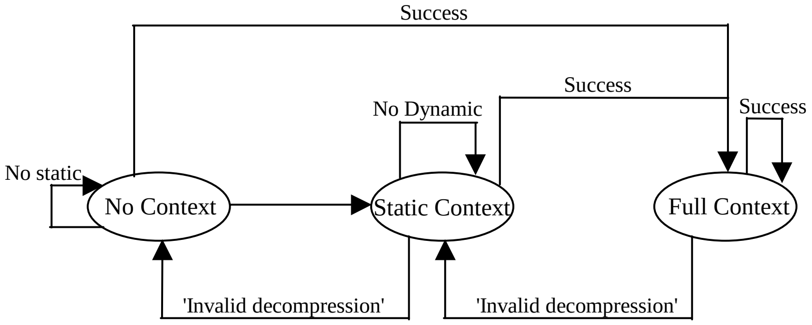

Figure 8: Reliable mode decompressor logic

5.1.6.4 Compressor and decompressor logic

It can be seen from the subclause 5.1.6 that the decompressor logic for all modes is the same as the uni-directional decompressor logic. What differs is the feedback logic, which states and what feedback messages are sent due to different events in each operating state.

5.1.7 Packet types

A ROHC RTP packet starts with a packet type identifier. The packet type indicates the format of the (first part of) packet. Packet types basically allow distinguishing between IR-DYN packets, compressed packets and feedback. Packet types can also be used to signal presence of segmented packets, padding octets, piggybacked feedback and context identifying bits in a compressed packet. As the packet type has been included in the ROHC packet it is not required to be carried by the link layer header as e.g. in RFC 2507 [10].

5.1.8 Packet formats

The ROHC protocol defines a set of packet formats. A general format for all packets is presented in Figure 9 below. Packet formats are organised in different categories:

- packet type 0 (UO-0, R-0, R-0-CRC);
- packet type 1 (R-mode, (R-1, R-1-TS, R-1-ID));
- packet type 1 (UO-modes, (UO-1, UO-1-ID, UO-1-TS));
- packet type 2 (UOR-2, UOR-2-ID, UOR-2-TS);
- FEEDBACK-1; and
- FEEDBACK-2.

U, O and R refer to U-mode, O-mode and R-mode, respectively. Different categories illustrate the length and the usage of the packet.

0 1 2 3 4 5 6 7	
Add-CID octet	If for CID 1-15 and small CIDs
First octet of base header	(with type indication)
0,1 or 2 octet of CID	1-2 octect is large CIDs
Reminder of base header	Variable number of bits
Extension	Extension, if X=1 in base header
IP-ID of inner IPv4 header	2 octets, if value (RND)=1
IP-ID of outer IPv4 header	2 octets, if value (RND2)=1
UDP checksum	2 octets, if context (UDP checksum)!=0
AH data for outer list	variable
AH data for inner list	variable
GRE checksum	2 octets, if GRE flag C=1

Figure 9: Packet formats from the compressor to the decompressor

5.1.9 Packet format description

This is described and specified in [6].

5.1.10 ROHC Configuration

ROHC has two kinds of parameters; configuration parameters that are mandatory and must be configured between compressor and decompressor peers, and implementation parameters that are optional and when used, mandate how a ROHC implementation operates.

Configuration parameters are mandatory and must be configured (signalled by RRC) between compressor and decompressor, so that they have the same values at compressor and decompressor. An example of a configuration parameter is context identification (CID).

Implementation parameters (ROHC primitives) make it possible to mandate how the ROHC compressor or decompressor should operate. Implementation parameters have local significance, are optional to use and are thus not necessary to be negotiated between compressor and decompressor.

This does not preclude that implementation parameters may be signalled or negotiated using lower layer functionality in order to set the way a ROHC implementation operates. Some implementation parameters are valid only at either the compressor or decompressor. Implementation parameters may further be divided into parameters that describe the way an implementation operates and into parameters that trigger a specific event, i.e., signals.

5.1.10.1 Profiles

As mentioned previously RFC 3095 [6] supports 4 different profiles. A profile describes exactly how to do compression and decompression. A profile is specific for each context and it is established with the IR header. An IR header contains a profile identifier, which determines how the rest of the header is to be interpreted. Note that the profile parameter determines the syntax and semantics of the packet type identifiers and packet types used for a specific context. Profiles have to be negotiated during link establishment. The decompressor indicates which profiles it supports and compressor may not compress using other ones. Currently, the following profiles have been defined:

- Profile 0 is for sending uncompressed IP packets.
- Profile 1 is for RTP/UDP/IP compression.
- Profile 2 is for UDP/IP compression, i.e., compression of the first 12 octets of the UDP payload is not attempted.
- Profile 3 is for ESP/IP compression, i.e., compression of the header chain up to and including the first ESP header, but not subsequent subheaders.

5.1.10.1.1 Context for Uncompressed Packets

It is possible also in the wireless environment to exceed the maximum number of supported contexts per radio bearer. Following possibilities are foreseen and have to be evaluated.

1. Only ROHC protocol used in PDCP entity. PDCP-No-Header PDU is used.
    - Different contexts are separated within ROHC protocol and CID is carried in the ROHC packet. The ROHC has its internal mechanisms to reserve one CID for sending uncompressed data over the radio link. This solution does not put any requirements on PDCP layer. All required mechanisms exist in release 99.
2. Only ROHC protocol used in PDCP entity. PDCP-Data PDU is used. This combination enables the separation of different contexts in ROHC or in PDCP.
    - If the CIDs are carried within the ROHC protocol and the case equals to point 1 above.
    - If the CIDs are carried out in PDCP. The ROHC protocol has to be configured to support only one context. The PDCP requires functionality before ROHC protocol to separate different flows and pass them through ROHC protocol. PDCP is also required to filter out data flows that exceed the maximum number of supported contexts. Such contexts should by-pass header compression and would be indicated with a dedicated CID in the PDCP header to enable bypassing the decompressor.
3. ROHC+RFC2507 protocols used in PDCP entity. As two different protocols are used it implies that PDCP-Data PDU is used.
    - This case is equal to point 2 above. A ROHC packet or PDCP header can be used to carry CID information. The CID should be carried in the PDCP headers as this utilises the one octet introduced by the PDCP header for ROHC, without having to use CID bits in the ROHC.

5.1.10.2 Context Identifier

The ROHC scheme has the possibility to support several contexts or unique IP flows. In order to track or identify these flows a Context ID (CID) is required in the ROHC packet format. This CID field is 0 (i.e. not present), 1 or 2 octets in length. Each CID has a context and must maintain enough information in order to correctly compress and decompress these IP flows. This context will be referred to as, CONTEXT_SIZE. Typically, CONTEXT_SIZE will include at least the full/uncompressed IP header. This would be the IP/UDP/RTP header, which is 40 octets for IPv4 and 60 octets for IPv6. The CONTEXT_SPACE is the sum of all CONTEXT_SIZEs.

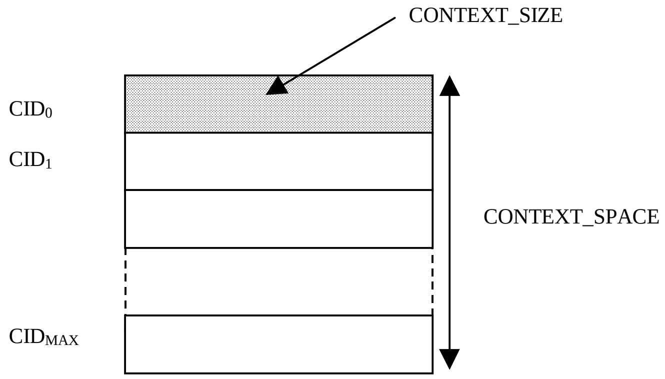

Figure 10: Header compressor contexts

The CID length of 2 octets is encoded to allow a maximum CID value of 16383 and would imply significant memory requirements. The need for such a large number of CIDs is that ROHC could be quite easily used in fixed IP backbone networks. In such cases there are several flows that a router would need to manage. This has been designed for the general case and to be future proof.

However, the requirements on the number of CIDs for each radio bearer in UTRAN pose less of a requirement and the maximum that is seen today is in the order of 5 to 15 flows per radio bearer for mobile terminals in this wireless environment.

Therefore, it is more efficient to signal the actual maximum CID value (CID_{MAX}) by RRC to the UE to configure ROHC with this value. In order for UTRAN to make a suitable choice for the maximum CID value certain capability parameters need to be indicated by the UE.

The UE, could for example indicate the CID_{MAX} in the IE "PDCP Capability". UTRAN could then configure ROHC with a maximum CID value in the range [0, CID_{MAX}]. This would be the simplest approach.

As explained in subclause 5.1.5 of this TR, the ROHC protocol has three modes; Uni-directional (U-mode), bi-directional optimistic (O-mode) and bi-directional reliable (R-mode). Each of these modes has different requirements on the typical amount of information that is required in the stored context, with the first two modes requiring the least amount of memory resources. If the UE were to indicate the CID_{MAX} then this would have to be for the mode that required the most context space, thus providing a worst case estimate.

There are, however, several alternatives:

I) CID_{MAX} (for all modes) is given in combination with $CONTEXT\_SPACE_{MAX}$ (for all modes).

II) CID_{MAX} is given for each mode. In addition a one bit indicator could be included in the IE "PDCP info" to indicate that the reported maximum CID is the same for all modes.

III)$CONTEXT\_SIZE_{MODE}$ is given for each MODE and a $CONTEXT\_SPACE_{MAX}$ also indicated.

IV)$CID_{MAX,1}$ is given for (U-mode and O-mode) and $CID_{MAX,2}$ is given for (R-mode). In this way there is no need for UTRAN to know the UE $CONTEXT\_SPACE_{MODE}$. UTRAN can then choose a maximum CID value (Max_UO_CID) between [0, $CID_{MAX,1}$] for U-mode and O-mode and a maximum CID value (Max_R_CID) [0, $CID_{MAX,2}$] for R-mode only. In addition a one bit indicator could be included in the IE "PDCP info" to indicate that the reported maximum CID is the same for all modes.

It is recommended that alternative II is chosen. Therefore Max_U_CID, Max_O_CID and Max_R_CID are indicated in the RRC IE "PDCP Capability". However, it has to be evaluated what effect different CID sizes for different modes have on the ROHC protocol. As the decompressor makes the decision of the mode changes, different CIDs in different modes will have an effect on the mode change criteria.

5.1.10.3 0-octet-CID

In subclause 5.1.10.2 it was stated that the CID field in the ROHC packet format could be 0, 1 or 2 octets in length. Large CID spaces are not typically required for terminals in the wireless scenario.

It is possible to include the CID value on the PDCP layer instead of ROHC headers. In 3GPP Release 99, the PDCP layer can be configured to introduce a PDCP header or not to introduce a PDCP header. The PDCP header is typically required for HC protocols that require link layer (i.e. the layer that 'carries' HC) identification of the HC packet types.

3GPP Release'99 includes only one HC protocol (RF2507) and this requires link level identification of the HC packet formats. Header compression protocols that do not require link layer identification do not require PDCP to include a PDCP header. ROHC is one such protocol. PDCP configured to support only ROHC would typically be configured not to introduce a PDCP header.

With the introduction of an additional HC protocol to the suite of HC protocols to be used in UTRAN it is currently not possible to configure PDCP entity to accommodate HC protocols that do and do not require link layer packet identification (or any other header compression information).

It is important that the HC algorithms that will be included in Release 4 can be used on the same PDCP entity. RFC2507 can compress UDP/IP and/or TCP/IP streams while the current ROHC profile can compress RTP/UDP/IP, UPD/IP and ESP/IP streams. It is quite typical that different types of IP streams will be mixed together, for example in streaming applications.

There are two ways that this can be achieved and these are discussed below:

- Introducing a new PDCP PDU type for ROHC packets
- Using the same PDCP PDU type for ROHC and RFC2507

5.1.10.3.1 Introducing a new PDU Type for ROHC packets

The PDCP data PDU is shown below in Table 1. This PDCP header is 1 octet and consists of 2 fields; the PDU type field (3 bits) and the PID field (5 bits).

Table 1: PDCP-Data-PDU format

PDU type	PID
Data	

The PDU Type field in the PDCP PDU header indicates the type of PDCP data PDU and is shown below in Table 2. The PID field identifies the exact header compression packet type and the setting of the PID values is described in [2].

Table 2: PDCP-Data-PDU format

Bit	PDU Type
000	PID field used for header compression information (PDCP-PDU format described in table 5)
001	PID field used for header compression information and the PDCP PDU sequence number included (PDCP-PDU format described in table 6)
010	ROHC CID packet (PDCP-PDU).
011-111	Reserved (PDUs with this encoding are invalid for this version of the protocol)

In order to mix RFC2507 and ROHC there needs to be a header in PDCP as RFC2507 requires link level identification of the HC packet formats. It is only a matter of introducing a PDU type (010: ROHC CID packet) for ROHC packets to allow the possibility to mix RFC2507 and ROHC. The PID field for this PDCP Type (010: ROHC CID packet) would be defined for use as CID identification. The benefit here is that 32 CIDs can be identified and that the ROHC packet format would operate without a CID i.e. in 0-octet-CID mode, thereby saving, at best 2 octets overhead. Therefore, the PDCP PDU format would be as described in Table 3.

Table 3: PDCP-Data-PDU format

PDU type (010)	CID
Data	

The data part as indicated in Table 3 would be the ROHC packet when ROHC would be configured to have CID_{MAX} = 1 context (it will ensure that the CID field is not present in the ROHC packet).

If the number of CIDs is greater than what can be accommodated in the PID field (referred as CID field in table y) of the PDCP PDU header then the CID of 1 or 2 octets in the ROHC packet is/are used. The PID field is therefore unused in this case.

5.1.10.3.2 Using the same PDCP PDU type for ROHC and RFC2507

In this case ROHC and RFC2507 protocols are identified by the same PDU Type = '000' but the packet formats for these protocols are distinguished by the 5 bit PID field in the PDCP Data PDU. For RFC2507, the PID values identify the packet formats and for ROHC the PID values are used to identify the CONTEXT ID for the flow. The maximum number of CIDs that can be accommodated in the 5-bit CID/PID field is dependent on the number of PID values that are used for identification of packet types for other header compression protocols. The PID field is not used to identify the ROHC packet formats as this is done within the ROHC

If ROHC is the only protocol that is configured then up to 32 CIDs could be identified. The variables Max_U_CID , Max_O_CID and Max_R_CID are configured by the RRC protocol during configuration or reconfiguration of the header compression protocol. Refer to Alternative II, in subclause 5.1.10.1 above.

The assignment of PIDs for CIDs is shown in Table 4 below.

Table 4: PID values assigned to ROHC header compression protocol

PID value	Optimisation method	Packet type
N+1	ROHC	CID1
N+2	ROHC	CID2
…	ROHC	…
…	ROHC	…
N+n	ROHC	CID$_n$

If the number of CIDs is greater than what can be accommodated in the PID field of the PDCP PDU header then the CID of 1 or 2 octets in the ROHC packet is used and the PID field is therefore unused in this case.

The 0-octet-CID option is the recommended CID configuration for ROHC in UTRAN.

5.1.10.4 Segmentation

ROHC protocol supports segmentation. The segmentation can vary packet by packet and it does not cause any overhead to packets that are not segmented. The segmentation may not be used when ROHC is run on top of UM RLC and then MRRU (maximum reconstructed reception unit) shall be equal to 0. The only case when the usage of ROHC segmentation is allowed is when ROHC is run on top of Tr RLC and the Packet_Sizes_Allowed is used to configure ROHC packet sizes. Furthermore, in that case segmentation may only be applied if the produced packet does not fit to the largest packet indicated by Packet_Sizes_Allowed.

5.1.10.5 Packet_Sizes_Allowed

This is a list of positive integer values that mandate the packet sizes that are allowed to be produced by ROHC. If this parameter list is set it has to contain packet size that allows a transmission of an entire IR header. Otherwise ROHC segmentation has to be used. It is also recommended to use packet sizes that are most frequently used in SO and FO states. If segmentation is not allowed/configured, then one of the Packet sizes allowed must accommodate the largest ROHC packet size possible, which typically is the IR packet. NOTE: ROHC has an in-built segmentation functionality, which takes place when the packets do not fit to the defined packet size. Therefore, extra attention has to be paid if the header sizes will be defined e.g. by RRC so that reasonable values are selected. Otherwise, segmentation will be performed in ROHC and possibly in RLC. However, it has to be noted that ROHC segmentation does not add any overhead to the packets that are not segmented (that should be the case in UTRAN).

5.1.10.6 Feedback

Feedback from decompressor to the compressor can be realised in several ways.

1) Feedback is provided as in release 99 when the feedback packets are transmitted with data packets on the same logical channel. This is same as the RLC AM model of operation for control and data RLC PDUs.

2) Another RLC link could be set up under one PDCP entity, which would be used as a dedicated feedback-only link.

3) The feedback packets are piggybacked to the compressed packets in the compressor. This option will require also mechanism 1 or 2 if the traffic to opposite direction does not exist or is very infrequent.

The recommended method of providing feedback is as in option 1) above.

5.1.10.7 Example PDCP Configuration

```
CHOICE Algorithm type: RFC2507 and ROHC
PDCP_Header             Present         PDCP will introduce a 1 octet header (PDU Type = 000 or 010)
RFC2507
        …                                                   …                   …
ROHC
        MAX_CID                                 10                  Maximum CID value for ROHC
        PROFILES SUPPORTED                      0,                  Uncompressed IP
                                                1,                  RTP/UDP/IP
                                                2,                  UDP/IP
                                                3                   ESP/IP
        PACKET_SIZES_ALLOWED                    1, 2, 6, 28, 40     In octets
        REVERSE_DECOMPRESSION_DEPTH             0                   No reverse decompression
```

```
    MRRU                                0                   ROHC segmentation not used.
```

5.1.11 ROHC primitives and parameters

5.1.11.1 ROHC Primitives at the compressor

5.1.11.1.1 CONTEXT_REINITIALIZATION

The parameter triggers the initialisation of static and dynamic parts of the context at the decompressor. In this state, the compressor sends complete header information. This includes all static and non-static fields in uncompressed form plus some additional information. When triggered the compressor shall reinitialise all contexts.

5.1.11.2 ROHC Primitives at the decompressor

5.1.11.2.1 MODE

This parameter triggers a mode transition using the mechanism described in ROHC protocol specification [6]. The possible parameter values are U, O and R and correspond to the U-mode, O-mode and R-mode, respectively.

The mode transition is made from the current mode to the new mode as signalled in the MODE primitive. MODE should not only serve as a trigger for mode transitions, but to also make it visible which mode ROHC operates in.

5.1.11.2.2 CLOCK_RESOLUTION

This parameter indicates the resolution of the system clock. A zero value will indicate that system clock is not available and thus timer based TS compression nor SN wrap-around detection are not possible.

5.1.11.2.3 Reverse_decompression_DEPTH

The parameter indicates whether reverse decompression is performed and to what extent. Value 0 indicates that reverse decompression shall not be performed.

5.1.11.3 Summary of ROHC Parameters

A ROHC implementation may have two kinds of parameters: configuration parameters that are mandatory and must be negotiated between compressor and decompressor peers, and implementation parameters that are optional and, when used, stipulate how a ROHC implementation is to operate.

Configuration parameters are mandatory and must be configured between compressor and decompressor, so that they have the same values at both compressor and decompressor.

Implementation parameters make it possible for an external entity to stipulate how an implementation of a ROHC compressor or decompressor should operate. Implementation parameters have local significance, are optional to use and are thus not necessary to negotiate between compressor and decompressor. As such, this does not affect the interoperability between ROHC implementations.

Some implementation parameters are valid only at either of compressor or decompressor. Implementation parameters may further be divided into parameters that allow an external entity to describe the way the implementation should operate and parameters that allow an external entity to trigger a specific event, i.e., signals.

The parameters in subclause 5.1.11.3.1 and 5.1.11.3.2 are classified in the following way:

M: Mandatory and configured by upper layers.

MO: Parameters that must be supported and when used can only be configured or triggered by upper layers.

O: Optional ROHC parameters that are not configured by upper layers. They may be used locally (i.e. UTRAN and/or in UE) for the ROHC implementation.

N/A: These are not used in the ROHC implementation/configuration.

The usage and definition of the parameters below are specified in [6].

5.1.11.3.1 Configuration Parameters

- MAX_CID (M)
- LARGE_CIDS: This is not configured by upper layers but inferred from the configured value of MAX_CID according to the following rule:

  If MAX_CID > 15 then LARGE_CIDS = TRUE else LARGE_CIDS = FALSE.

- PROFILES (M): Profiles are used to define which profiles are allowed to be used by the UE in uplink. All profiles defined in [6] shall be supported by UE.
- FEEDBACK_FOR (N/A):
- MRRU (M): Segmentation is not used by default.

5.1.11.3.2 Implementation Parameters

5.1.11.3.2.1 Compressor

Operational

(PDCP → ROHC):

- NO_OF_PACKET_SIZES_ALLOWED (O)
- PACKET_SIZES_ALLOWED (MO): When this option is used then no others packet sizes are allowed to be used.
- PAYLOAD_SIZES (O)

(ROHC → PDCP):

- NO_OF_PACKET_SIZES_USED (O)
- PACKET_SIZES_USED (O)

Event Triggers

- CONTEXT_REINITIALIZATION (MO)

5.1.11.3.2.2 Decompressor

Operational

(PDCP→ROHC):

- MODE (O)
- CLOCK_RESOLUTION (O)
- REVERSE_DECOMPRESSION_DEPTH (M): Default value is that reverse decompression is not used.

(ROHC→ PDCP):

- None.

Event Triggers

- None.

5.1.13 ROHC Reconfiguration

ROHC is reconfigured in the same way as PDCP in Release 99, by the RRC RADIO BEARER RECONFIGURATION message [3].

5.1.14 ROHC with SRNS relocation

As ROHC is part of the PDCP layer, there is a compressor and decompressor pair in the RNC and a corresponding compressor and decompressor pair in the UE. During SRNS relocation the source RNC gives the role of the serving RNC (SRNC) to the target RNC. This is described in [11] and in [12].

The impact of SRNS relocation on header compression is only on the state of the context information. It needs to be defined how the contexts are handled as a result of the relocation. Two methods exist for when the target RNC supports RFC 3095:

1) No transfer of context information from source RNC to target RNC

2) Transfer of context information from source RNC to target RNC

On the selection of the two methods, as stated above, the following have to be considered:

- No context transfer will cause additional overhead to the radio interface after relocation, as the contexts have to be initialised in UE and in UTRAN. This may cause e.g. speech quality degradation as the header parts may steal resources from the payload.

- Context transfer option does not cause any additional overhead to the radio interface. However, The Iur is more loaded as the contexts have to be transferred over that. Furthermore, the compressor/decompressor relocation will cause a break to real time traffic.

In the selection of the method of handling contexts during relocation, the main selection criteria should be to minimise degradation of traffic quality during relocation. Other criteria may be complexity and overhead in radio and Iur interface.

For when the target RNC does not support RFC 3095 is FFS.

5.1.14.1 No context transfer

Immediately after relocation, the compressor in the UE and the target RNC must switch to the IR (Initialisation and Refresh state) in U-mode. The ROHC CONTEXT_REINITIALIZATION primitive as described in subclause 5.1.11.1.1 is used for this purpose.

PDCP uses the ROHC CONTEXT_REINITIALIZATION primitive after receiving the CPDCP-CONFIG-Req with the R/I parameter in the UE and the target RNC.

This is shown in the figures below. Figure 11 and Figure 13 show the combined Cell/URA update with SRNS relocation examples for a lossless and seamless radio bearer respectively as shown in [12]. Similarly, Figure 12 and Figure 14 show the Hard Handover with SRNS relocation examples for a lossless and seamless radio bearer respectively as shown in [12]. The benefit of this approach is that there is no impact on RAN WG3 specifications. However, this scheme will temporarily add overhead to the radio interface and degrade e.g. voice quality as the initialisation of contexts is being performed and so full headers are sent in form of static and dynamic parts.

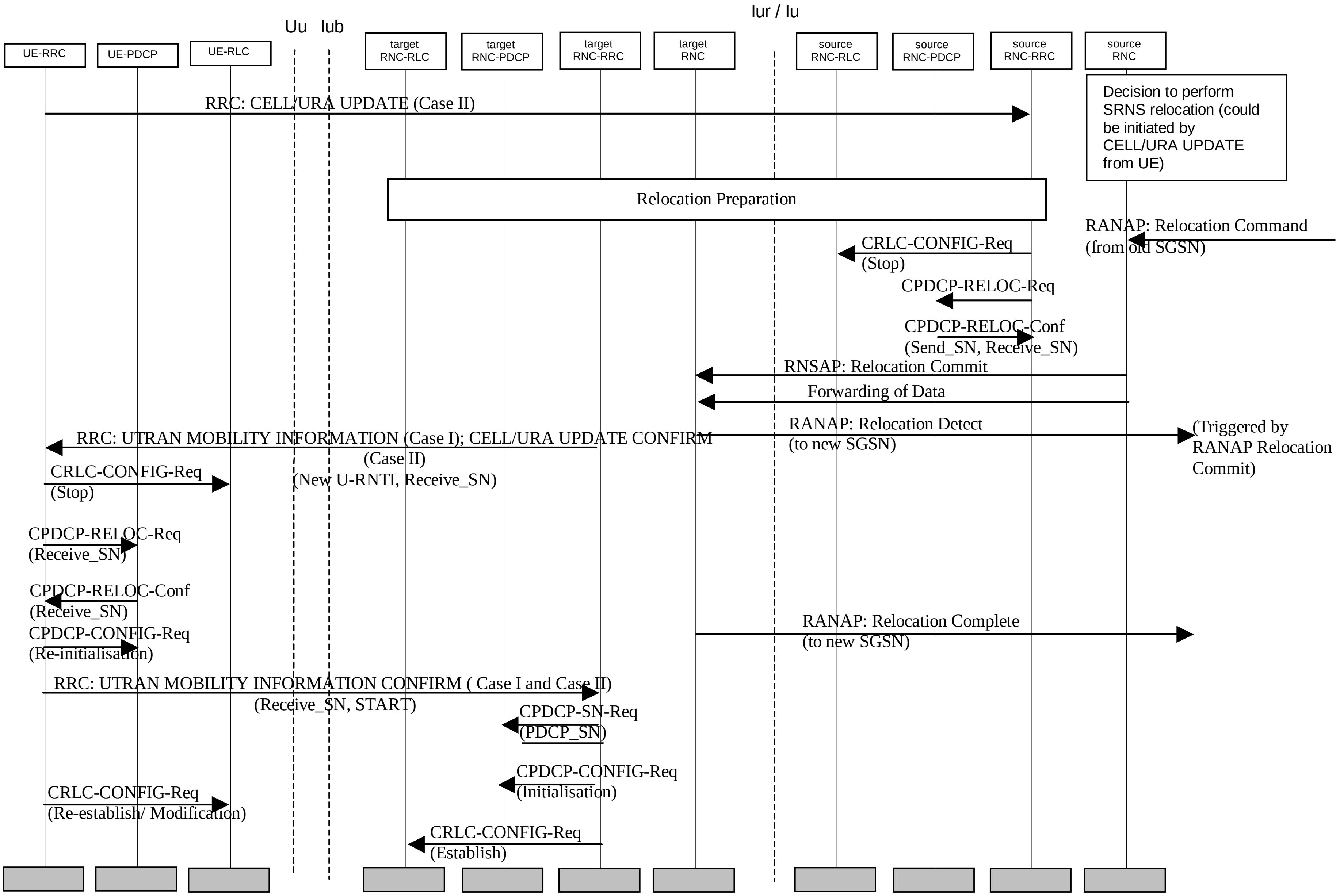
Iur / Iu
Uu
Iub
UE-RRC
UE-PDCP
UE-RLC
target RNC-RLC
target RNC-PDCP
target RNC-RRC
target RNC
source RNC-RLC
source RNC-PDCP
source RNC-RRC
source RNC
RRC: CELL/URA UPDATE (Case II)
Decision to perform SRNS relocation (could be initiated by CELL/URA UPDATE from UE)
Relocation Preparation
RANAP: Relocation Command (from old SGSN)
CRLC-CONFIG-Req (Stop)
CPDCP-RELOC-Req
CPDCP-RELOC-Conf (Send_SN, Receive_SN)
RNSAP: Relocation Commit
Forwarding of Data
RANAP: Relocation Detect (to new SGSN)
(Triggered by RANAP Relocation Commit)
RRC: UTRAN MOBILITY INFORMATION (Case I); CELL/URA UPDATE CONFIRM (Case II) (New U-RNTI, Receive_SN)
CRLC-CONFIG-Req (Stop)
CPDCP-RELOC-Req (Receive_SN)
CPDCP-RELOC-Conf (Receive_SN)
CPDCP-CONFIG-Req (Re-initialisation)
RANAP: Relocation Complete (to new SGSN)
RRC: UTRAN MOBILITY INFORMATION CONFIRM (Case I and Case II) (Receive_SN, START)
CPDCP-SN-Req (PDCP_SN)
CPDCP-CONFIG-Req (Initialisation)
CRLC-CONFIG-Req (Re-establish/ Modification)
CRLC-CONFIG-Req (Establish)

Figure 11: ROHC handling during combined Cell/URA Update and SRNS relocation (lossless radio bearers)

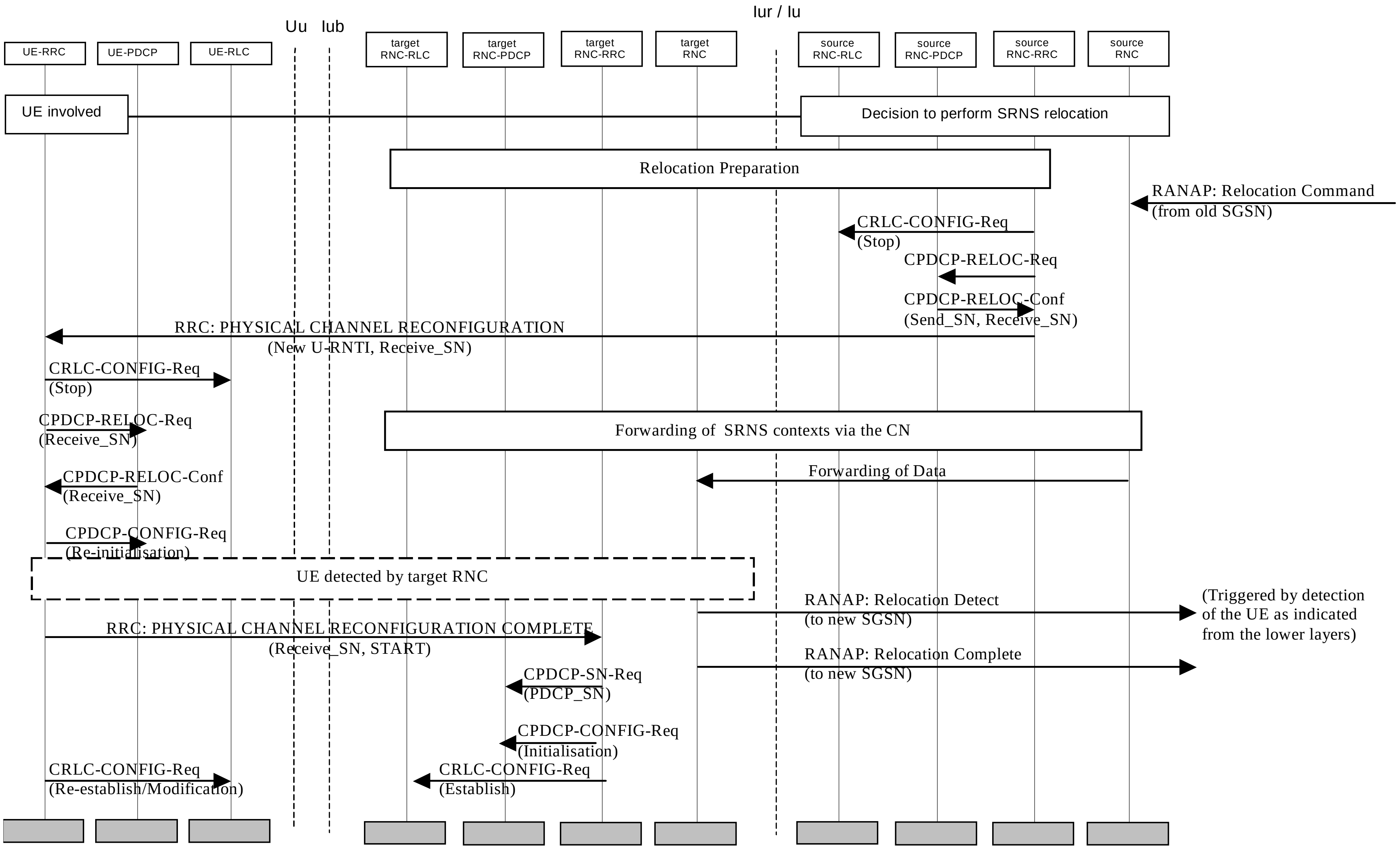
Iur / Iu
Uu
Iub
UE-RRC
UE-PDCP
UE-RLC
target RNC-RLC
target RNC-PDCP
target RNC-RRC
target RNC
source RNC-RLC
source RNC-PDCP
source RNC-RRC
source RNC
UE involved
Decision to perform SRNS relocation
Relocation Preparation
RANAP: Relocation Command (from old SGSN)
CRLC-CONFIG-Req (Stop)
CPDCP-RELOC-Req
CPDCP-RELOC-Conf (Send_SN, Receive_SN)
RRC: PHYSICAL CHANNEL RECONFIGURATION (New U-RNTI, Receive_SN)
CRLC-CONFIG-Req (Stop)
CPDCP-RELOC-Req (Receive_SN)
Forwarding of SRNS contexts via the CN
CPDCP-RELOC-Conf (Receive_SN)
Forwarding of Data
CPDCP-CONFIG-Req (Re-initialisation)
UE detected by target RNC
RANAP: Relocation Detect (to new SGSN)
(Triggered by detection of the UE as indicated from the lower layers)
RRC: PHYSICAL CHANNEL RECONFIGURATION COMPLETE (Receive_SN, START)
RANAP: Relocation Complete (to new SGSN)
CPDCP-SN-Req (PDCP_SN)
CPDCP-CONFIG-Req (Initialisation)
CRLC-CONFIG-Req (Re-establish/Modification)
CRLC-CONFIG-Req (Establish)

Figure 12: ROHC handling during Hard Handover with SRNS relocation (lossless radio bearers)

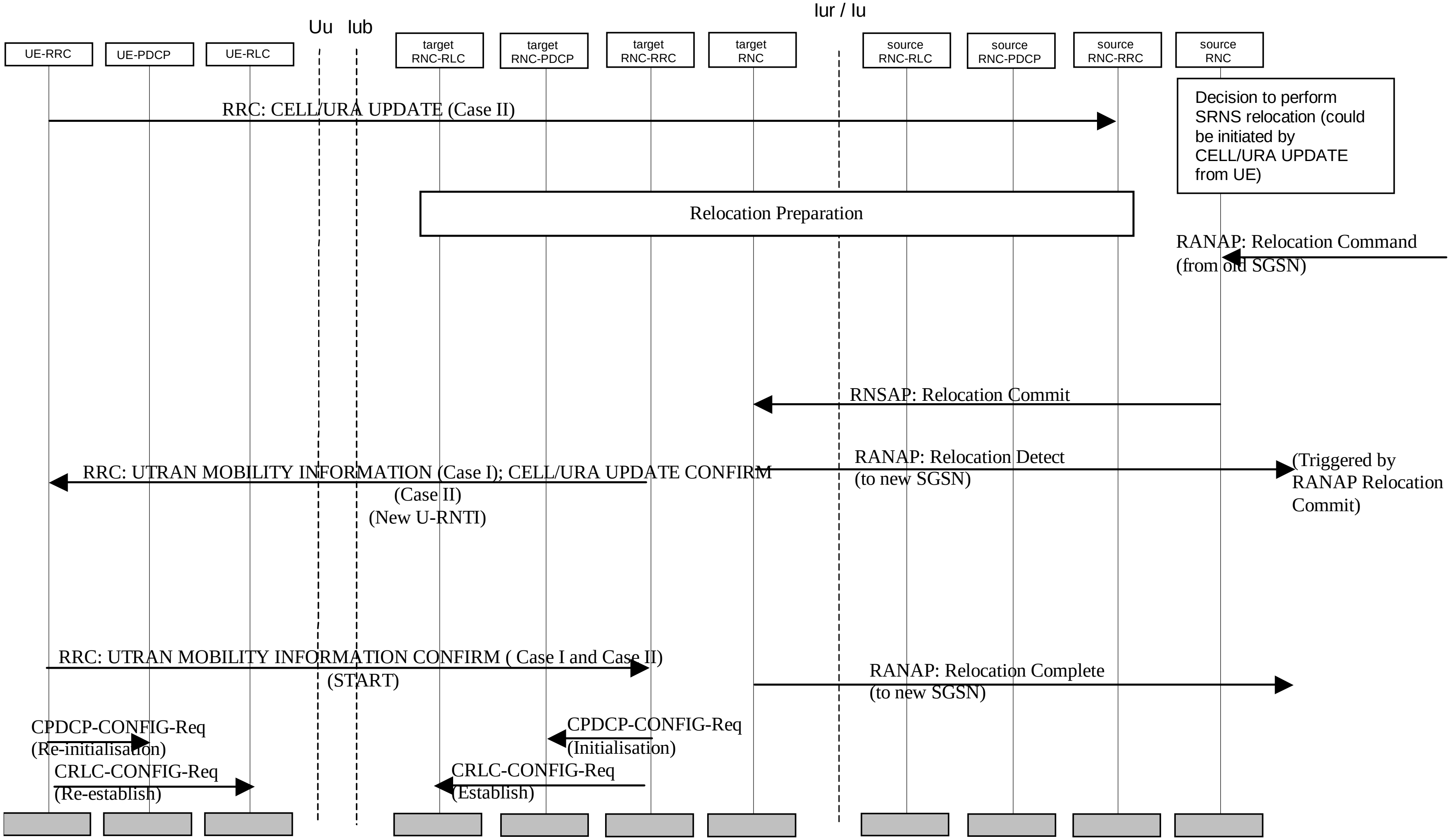
Uu
Iub
Iur / Iu
UE-RRC
UE-PDCP
UE-RLC
target RNC-RLC
target RNC-PDCP
target RNC-RRC
target RNC
source RNC-RLC
source RNC-PDCP
source RNC-RRC
source RNC
Decision to perform SRNS relocation (could be initiated by CELL/URA UPDATE from UE)
RRC: CELL/URA UPDATE (Case II)
Relocation Preparation
RANAP: Relocation Command (from old SGSN)
RNSAP: Relocation Commit
RANAP: Relocation Detect (to new SGSN)
(Triggered by RANAP Relocation Commit)
RRC: UTRAN MOBILITY INFORMATION (Case I); CELL/URA UPDATE CONFIRM (Case II) (New U-RNTI)
RRC: UTRAN MOBILITY INFORMATION CONFIRM (Case I and Case II) (START)
RANAP: Relocation Complete (to new SGSN)
CPDCP-CONFIG-Req (Re-initialisation)
CRLC-CONFIG-Req (Re-establish)
CPDCP-CONFIG-Req (Initialisation)
CRLC-CONFIG-Req (Establish)

Figure 13: ROHC handling during combined Cell/URA Update and SRNS relocation (seamless radio bearers)

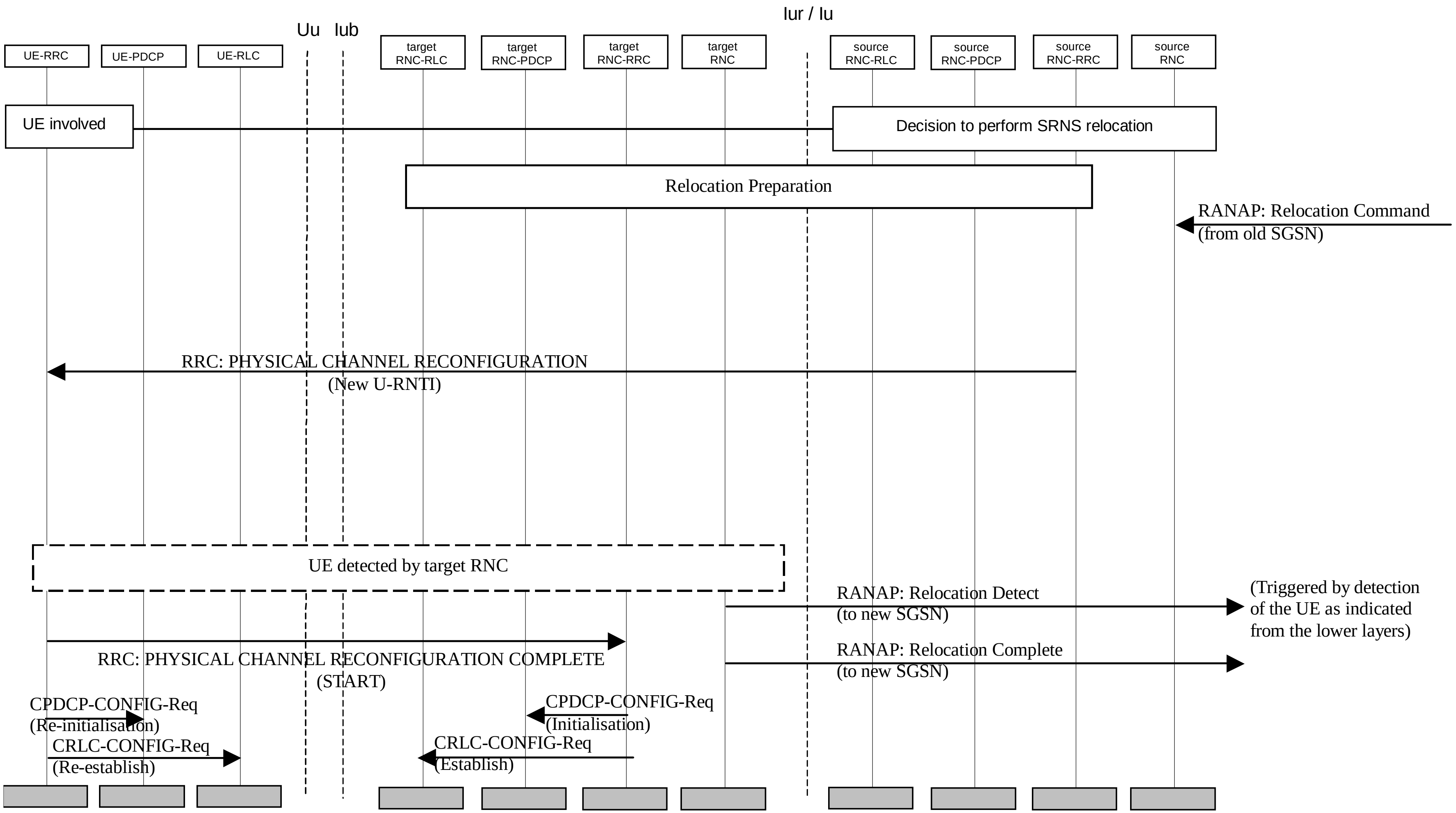
Uu
Iub
Iur / Iu
UE-RRC
UE-PDCP
UE-RLC
target RNC-RLC
target RNC-PDCP
target RNC-RRC
target RNC
source RNC-RLC
source RNC-PDCP
source RNC-RRC
source RNC
UE involved
Decision to perform SRNS relocation
Relocation Preparation
RANAP: Relocation Command (from old SGSN)
RRC: PHYSICAL CHANNEL RECONFIGURATION (New U-RNTI)
UE detected by target RNC
RANAP: Relocation Detect (to new SGSN)
(Triggered by detection of the UE as indicated from the lower layers)
RRC: PHYSICAL CHANNEL RECONFIGURATION COMPLETE (START)
RANAP: Relocation Complete (to new SGSN)
CPDCP-CONFIG-Req (Re-initialisation)
CRLC-CONFIG-Req (Re-establish)
CPDCP-CONFIG-Req (Initialisation)
CRLC-CONFIG-Req (Establish)

Figure 14: ROHC handling during Hard Handover with SRNS relocation (seamless radio bearers)

5.1.14.2 Context transfer

The context relocation for RFC3095 is described in a separate subclause 5.2 dealing with the SRNS context relocation in general case.

5.2 SRNS context relocation

This subclause describes how SRNS relocation is handled when the header compression context is transferred from source to target SRNC in order to minimise the degradation of traffic quality during relocation. This technical report discusses only the following case, since it is seen the most important from the real-time UTRAN service point of view:

- UTRAN source-RNC using RFC 3095 (ROHC) => UTRAN target-RNC using RFC 3095 (ROHC)

All the other cases are handled as specified in Release 4, i.e. no context transfer is applied but the header compression is initialised during the relocation.

5.2.1 Overview of the feature

The straightforward working solution in Release 4 has been to initialise the header compression in both peers after relocation (see subclause 5.1.14.1) resulting in problems of e.g. high probability of lost speech frames. This could be avoided by not initialising compression but continuing it in the target SRNC from the place in which the compression ended in the source SRNC.

This feature is based on the principle that a snapshot of the header compressor context for downlink is first taken in the source SRNC. The context information is then transferred to target SRNC while performing the relocation signalling between source and target SRNC. The compressor context in the target SRNC is then initialised with the transferred information. Eventually when the control is really given to the target SRNC, it starts compressing with that context. The decompressor context is similarly transferred between source and target SRNC. As a consequence, neither uplink nor downlink user plane data transmission is seriously affected by the SRNS relocation.

5.2.2 Some terminology

Context synchronisation A compression context is synchronised with a decompression context when a header compressed according to the compression context is correctly decompressed with the decompression context.

Link switching In terms of UTRAN procedures the link switching in this document means the process when target SRNC is assigned the responsibility of serving RNC. I.e. it is the point of time when layer 2 protocols of target SRNC become active and UE is notified about the changed SRNC by the presence of the IE "Downlink counter synchronization info" in the received RRC message. The link switching is typically preceded by a preparation phase, during which the information necessary to carry out the switching is exchanged between source and target SRNCs.

N-context Refers collectively to both *N-context-C* and *N-context-D.*

N-context-C The compression context for downlink in SRNC at any given point of time.

N-context-D The decompression context for uplink in SRNC at any given point of time.

*N-context** Refers collectively to both *N-context-C** and *N-context-D*.*

*N-context-C** The frozen snapshot of the compression context for downlink taken by SRNC

*N-context-D** The frozen snapshot of the decompression context for uplink taken by SRNC

N-HC Entity located in the network that performs header compression for downlink
(i.e. RNC PDCP in UTRAN)

N-HD Entity located in the network that performs header decompression for uplink
(i.e. RNC PDCP in UTRAN)

N-HCD Refers collectively to both *N-HC* and *N-HD*.

M-context Refers collectively to both *M-context-C* and *M-context-D*.

M-context-C The compression context for uplink in UE at any given point of time.

M-context-D The decompression context for downlink in UE at any given point of time.

*M-context** Refers collectively to both *M-context-C** and *M-context-D**.

*M-context-C** The frozen snapshot of the compression context for uplink taken by UE.

*M-context-D** The frozen snapshot of the decompression context for downlink taken by UE.

M-HC Entity located in the mobile terminal that performs header compression for uplink (i.e. UE PDCP in UTRAN)

M-HD Entity located in the mobile terminal that performs header decompression for downlink (i.e. UE PDCP in UTRAN)

M-HCD Refers collectively to both *M-HC* and *M-HD*.

5.2.3 Challenges of context transfer

There are some generic challenges in context transfer, which need to be addressed by the method that realises the concept. These challenges are briefly discussed here in order to give some requirements for the relocation concept.

5.2.3.1 Stale context

There is some elapsed time between t_1, the time when the source SRNC takes a snapshot of its context for transfer to the target SRNC, and t_2, the time when the target SRNC starts to use the contexts. Between t_1 and t_2, the source SRNC and UE may have continued to exchange traffic on the old link. The source SRNC may have updated its compression context, and the UE updated its decompression context accordingly to stay in synchronization. In the same fashion, the UE may have updated its compression context, and the source SRNC updated its decompression context accordingly. When that happens, the *N-context* used by the target SRNC may be out of sync with the *M-context*, and consequently, decompression is incorrect. In particular, the header compression modes defined by RFC3095 have to be in sync after handover.

5.2.3.2 Skew of timer

When the timer-based compression/decompression of RTP time stamp is in effect, a timer is used at the compressor and decompressor to measure the packet arrival times (refer to RFC 3095, subclause 4.5.4). For the target SRNC to continue timer-based compression/decompression, a snapshot of the timer value at the source SRNC is taken and transferred to the target SRNC, so that the target SRNC can initialise its timer with the transferred value. The transfer latency time, *T_transfer*, may create a skew in the calculation of the elapsed time for the first few headers compressed or decompressed by the target SRNC, since that elapsed time is the difference between two arrival times measured by two different timers (one, *a_i*, at the target SRNC, and the other, *a_ref*, at the source SRNC).

The skew may or may not be a problem, depending on the magnitude of *T_transfer*.

5.2.4 Technical approach of the feature

5.2.4.1 RFC3095

5.2.4.1.1 R mode

For RFC3095 context relocation to work, the challenge of stale context must be addressed. It can be done easily by stopping to update the context at the compressor while relocation is taking place. In the case of RFC3095 **R mode**, it is relatively straightforward, because the old RNC can inhibit its compressor context update for the downlink, and stop sending ack for the uplink. By stopping the acks, it will cause the mobile station to stop updating its compressor context.

5.2.4.1.2 O mode

The case of RFC3095 **O mode** is more tricky, since the compressor (decompressor) continuously updates its compression (decompression) context at every packet, regardless of the signalling from the other side. A few options can be seen as a solution for RFC3095 O mode:

1) to prevent the whole SRNS relocation for radio bearers running RFC3095 in O mode

2) to accept problems of RFC3095 initialisation for radio bearers running RFC3095 in O mode (i.e. no RFC3095 context relocation during SRNS relocation)

3) to temporarily mimic the R mode during the relocation for radio bearers running RFC3095 in O mode

The first one of these options is easy to understand but is not really a solution for the problem, since it disables the whole SRNS relocation and is against the basic assumption of the scenario where SRNS relocation is performed.

The second option is clearly a solution for the uplink problems of O mode and stale context. It is supported by all Rel4 terminals supporting RFC3095 by default, so it doesn't imply any additional implementation requirements. On the other hand it doesn't bring any improvements into Release 4 SRNS relocations.

Consequently the third option seems to be the only way to support context relocation in RFC3095 O mode bringing really some additional value to the relocation. So the proposed concept solution for RFC3095 O mode is for the old RNC to send signalling to the mobile station so it can mimic the R mode for both uplink and downlink during context relocation. Once the context relocation is completed, and the compression/decompression has resumed at the new RNC, the system can switch back to O mode, if desired.

5.2.4.1.3 U mode

Even though it is the most tricky to solve the challenges of reliable context transfer for RFC3095 **U mode** due to its unreliable nature, some consideration needs to be addressed also for that since the compressor/decompressor may well be running in U mode when SRNS relocation is performed. As in O mode, the problem is caused by the fact that every packet in U mode updates the compression context at sender side and the decompression context at receiver side. Once again three totally different kind of approaches could be proposed for U mode:

1) to prevent the whole SRNS relocation for radio bearers running RFC3095 in U mode

2) to accept problems of RFC3095 initialisation for radio bearers running RFC3095 in U mode (i.e. no RFC3095 context relocation during SRNS relocation)

3) to transfer the static part of the context during SRNS relocation and force the update of the dynamic part of the context right after SRNS relocation in RFC3095 U mode in order to address stale context challenge.

When considering these approaches it becomes clear that only the third option is valid with the same reasoning as with O mode. The proposed concept solution for RFC3095 U mode is for the old RNC to send signalling to the mobile station so that right after SRNS relocation the M-HC initiate a transition back to the IR state and send IR-DYN as defined in RFC 3095 to re-synchronise the dynamic part of the uplink compressor and decompressor contexts. Note that for downlink, the target N-HC will automatically go to IR state and send IR-DYN following the normal logic in U mode.

5.2.4.1.4 How challenges are addressed

The presented problems in subclause 5.2.3 are addressed by the approach described above. Here is a short explanation of why:

- The stale context problem is avoided for R and O modes by inhibiting any context update once the context transfer has started, and for U mode by transferring only static part of the context.

- The skew of timer does not exist since the transfer latency is considered to be bounded in UTRAN. While being so, it can be compensated in the elapsed time calculation. The latency time is considered as an additional jitter in RTP timestamps. In order to compensate the jitter the compressor adds a maximum latency time to the total jitter and when sending the next RTP TS header info to the decompressor after the relocation it calculates how many bits are needed to deliver this RTP TS header. The amount of needed bits is dependent on the maximum latency (i.e. the selected upper limit), i.e. the bigger latency is selected, the more extra bits are always needed. Thus it is beneficial to select as low upper limit as possible.

5.2.4.1.5 Generic procedure

In this subclause the generic procedure is given that combines the approaches drafted above and gives a generic framework for realising the procedure in UTRAN.

The relocation proceeds through the following steps **for uplink**:

	R mode	**O mode**	**U mode**
1.	The network determines that the relocation has to be performed and notifies the source N-HCD about this.		
2.	The **source N-HD** does the following in an atomic fashion: - takes a snapshot of its - *N-context-D* (denoted *N-context-D**) - transfers snapshot to the new N-HCD - stops generating ack.	The **source N-HD** does the following in an atomic fashion: - takes a snapshot of its - *N-context-D* (denoted *N-context-D**) - transfers snapshot to the new N-HD - switches operationally to R mode without interaction with the compressor - stops generating ack.	The **source N-HD** does the following in an atomic fashion: - takes a snapshot of - the static part of its *N-context-D* (denoted *N-context-D-static**) - transfers snapshot to the new N-HD.
3.	The **source N-HD** sends an indication **to the M-HC** that this is an SRNS relocation (presence of IE "Downlink counter synchronization info").		
4.	-	Upon the reception of the relocation indication, **the M-HC** shall switch operationally to R mode without interaction with the decompressor.	-
5.	Target RNC becomes the source RNC.		
6.	Right after link switching, - **New N-HD** decompresses by using *N-context-D**; acks are again generated normally - **M-HC** continues to compress normally.	Right after link switching, - **New N-HD** decompresses in R mode by using *N-context-D** - **M-HC** compresses in R mode - **New N-HD** may command M-HC back to O mode by using feedback.	Right after link switching, - **M-HC** transits to IR state and sends IR-DYN to re-synchronise the dynamic part of the uplink context. - **New N-HD** decompresses by using *N-context-D**

NOTE: for uplink (only when the system was in R and O modes before SRNS relocation):
In phase 2, when the source N-HD is unsure of its context integrity (e.g. due to multiple detected errors), it just transfers the static part of its N-context-D (denoted *N-context-D-static*). Upon receiving N-context-D-static instead of N-context-D* in the normal case, the target N-HD will send a request for IR-DYN as soon as the new link to M-HCD is up.

The relocation proceeds through the following steps **for downlink**:

	R mode	**O mode**	**U mode**
1.	The network determines that the relocation has to be performed and notifies the source N-HCD about this.		
2.	The **source N-HC** does the following in an atomic fashion: - takes a snapshot of its - *N-context-C* (denoted *N-context-C**) - transfers snapshot to the new N-HC - inhibits its *N-context-C* update (i.e. will not update the context).	The **source N-HC** does the following in an atomic fashion: - takes a snapshot of its - *N-context-C* (denoted *N-context-C**) - transfers snapshot to the new N-HC - switches operationally to R mode without interaction with the decompressor - inhibits its *N-context-C* update.	The **source N-HC** does the following in an atomic fashion: - takes a snapshot of - the static part of its *N-context-C* (denoted *N-context-C-static**) - transfers snapshot to the new N-HC.
3.	The **source N-HC** sends an indication **to the M-HD** that this is an SRNS relocation (presence of IE "Downlink counter synchronization info").		
4.	Upon the reception of the relocation indication, **- M-HD** shall flush the reverse decompression buffer, if such was used.	Upon the reception of the relocation indication, **- M-HD** shall flush the reverse decompression buffer, if such was used **- M-HD** shall switch operationally to R mode without interaction with the compressor.	Upon the reception of the relocation indication, **- M-HD** shall flush the reverse decompression buffer, if such was used.
5.	Target RNC becomes the source RNC.		
6.	Right after link switching, - **New N-HC** compresses by using *N-context-C** - **M-HD** continues to decompress normally - **New N-HC** resumes updating its context.	Right after link switching, - **New N-HC** compresses in R mode by using *N-context-C** - **M-HD** decompresses in R mode - **New N-HC** resumes updating its context - **M-HD** may command the new N-HC back to O mode by using feedback.	Right after link switching, - **New N-HC** compresses by using *N-context-C** (based on U mode definition, N-HC goes to IR state and sends IR-DYN first to resynchronise the dynamic part of the downlink context) - **M-HD** continues to decompress normally.

6 Impacts on RAN WGs

This clause reviews the impacts on other RAN WGs in relation to the study areas as outlined in the Scope clause of the Technical Report for the Release 5 work item, "Radio Access Bearer Support Enhancements".

Where a study area has been recommended as not for Release 5 (refer to clause 7), the impacts to the other RAN WGs are indicated as "Not applicable for Release 5".

6.1 WG1

6.1.1 Robust header compression

None.

6.1.2 SRNS context relocation

None.

6.2 WG2

6.2.1 Robust Header Compression

None.

6.2.2 SRNS context relocation

The 3GPP TSG RAN WG2 specifications that are affected are described in the following subclauses:

6.2.2.1 Interlayer Procedures in Connected Mode 25.303

The figures and textual descriptions of SRNS relocation procedures need to be updated according to new primitives and parameters needed.

6.2.2.2 UE Radio Access Capabilities 25.306

The new capability concerning the support for RFC3095 context relocation needs to be added.

6.2.2.3 Packet Data Convergence Protocol 25.323

Exact behaviour, i.e. all specific actions of PDCP and RFC3095, in case of context relocation needs to be specified based on steps shown in the subclause 5.2.4.1.5.

New primitives and their parameters to enable SRNS context relocation need to be added.

6.2.2.4 Radio Resource Control 25.331

The context information ("RFC3095 Context Info") container to be transferred between source and target SRNC needs to be specified so that it can be referred to in the RANAP signalling; i.e. "RFC3095 Context Info" (defined in 25.331 RRC) will be included

1) in "RANAP Relocation Information" (defined in 25.413 RANAP) which is referred to in 25.423 RNSAP, i.e. included in RNSAP Relocation Commit message over Iur, or

2) in "RANAP Forward Context" message (defined in 25.413 RANAP) sent via Iu.

The new capability concerning the support for RFC3095 context relocation needs to be added in UE radio access capability information as specified in TS 25.306.

All messages involved in SRNS relocation need to be updated to contain information whether the relocation is performed with or without context relocation.

6.3 WG3

6.3.1 Robust header compression

None.

6.3.2 SRNS context relocation

The 3GPP TSG RAN WG3 specifications that are affected are described in the following subclauses:

6.3.6.1 UTRAN Iu Interface RANAP Signalling 25.413

RANAP information elements "Forward SRNS Context" and "RANAP Relocation Information" need to be modified to contain "RFC3095 Context Info" container.

6.4 WG4

6.4.1 Robust header compression

None.

6.4.2 SRNS context relocation

None.

7 Recommendations

7.1 Robust header compression

None.

7.2 SRNS context relocation

The RFC3095 changes to SRNS context relocation are included in Release 5 as outlined in subclause 6 of this technical report.

8 Release '99 Specification impacts

No impacts on Release '99 specifications.

Annex A: Change history

Document history		
Date	**Version**	**Comment**
2001-11	V 0.0.1	First proposal
2001-11	V 0.1.0	
2002-01	V 0.2.0	Updates based on RAN WG2#26
2002-03	v 1.0.0	editorial clean-up for RAN#15
Editor for 3GPP TR 25.860 is Juha Mikola, Nokia:		
Tel. : +358 50 48 37412 Fax : +358 71 80 36850 Email : juha.mikola@nokia.com		
This document is written in Microsoft Word version.		

NOTE: The table above will be deleted when this TS is presented to the TSG-RAN plenary in version 2.0.0 and be replaced by the table below.

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**