TR 25.933 V1.0.0 (2001-03)

Technical Report

3rd Generation Partnership Project (3GPP);
Technical Specification Group (TSG) RAN;

IP Transport in UTRAN Work Task Technical Report

UMTS <spec>

Reference

<Workitem> (<Shortfilename>.PDF)

Keywords

<keyword[, keyword]>

3GPP

Postal address

Office address

Internet

secretariat@3gpp.org
Individual copies of this deliverable
can be downloaded from
http://www.3gpp.org

Contents

Intellectual Property Rights

Foreword

This Technical Report (TR) has been produced by the 3rd Generation Partnership Project (3GPP), Technical Specification Group RAN.

The contents of this TR are subject to continuing work within 3GPP and may change following formal TSG approval. Should the TSG modify the contents of this TR, it will be re-released with an identifying change of release date and an increase in version number as follows:

Version m.t.e

where:

m indicates [major version number]

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

y the third digit is incremented when editorial only changes have been incorporated into the specification.

1 Scope

The purpose of the present document is to help the TSG RAN WG3 group to specify the changes to existing specifications, needed for the introduction of "IP Transport" option in the UTRAN for Release 2000. It is intended to gather all information in order to trace the history and the status of the Work Task in RAN WG3. It is not intended to replace contributions and Change Requests, but only to list conclusions and make reference to agreed contributions and CRs. When solutions are sufficiently stable, the CRs can be issued.

It describes agreed requirements related to the Work Task, and split the Work Task into "Study Areas" in order to group contributions in a consistent way.

It identifies the affected specifications with related Change Requests.

It also describes the schedule of the Work Task.

This document is a 'living' document, i.e. it is permanently updated and presented to all TSG-RAN meetings.

~~1~~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.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1.] IP-Transport in UTRAN Work Task Description, TSGRP#6(99)836

[2.] TS 25.401, UTRAN Overall Description

[3.] TS 25.410, UTRAN I_u Interface: General Aspects and Principles

[4.] TS 25.412, UTRAN I_u Interface Signalling Transport

[5.] TS 25.420, UTRAN I_{ur} Interface: General Aspects and Principles

[6.] TS 25.422, UTRAN I_{ur} Interface Signalling Transport

[7.] TS 25.430, UTRAN I_{ub} Interface: General Aspects and Principles

[8.] TS 25.427, UTRAN Iur and Iub interface user plane protocols for DCH data streams.

[9.] "Requirements for IP Version 4 Routers", RFC1812, June 1995.

~~[5]~~[10.] R. Pazhyannur, I. Ali, Craig Fox, "PPP Multiplexed Frame Option", <draft-ietf-pppext-pppmux-01.txt>, October 2, 2000.

~~[5]~~[11.] W. Simpson, Ed., "The Point-To-Point Protocol (PPP)", STD 51, RDF 1661, July 1994.

~~[5]~~[12.] W. Simpson, Ed., "PPP in HDLC-like Framing", STD 51, RFC 1662, July 1994.

~~[5]~~[13.] S. Casner, V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

~~[5]~~[14.] M. Engan, S. Casner, C. Bromann, "IP Header Compression over PPP", RFC 2509, February 1999.

~~[5]~~[15.] G. Gross, M. Kaycee, A. Lin, J. Stephens, "PPP Over AAL5", RFC 2364, July 1998.

~~[5]~~[16.] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC2661, August 1999.

~~[5]~~[17.] Bruce Thompson, Tmima Koren, Dan Wing, "Tunneling multiplexed Compressed RTP (TCRTP)", <draft-ietf-avt-tcrtp.01.txt>, July 12, 2000.

~~[5]~~[18.] Andrew J. Valencia, "L2TP Header Compression (L2TPHC), <draft-ietf-l2tpext-l2tphc-01.txt>, April 2000.

~~[5]~~[19.] Tmima Koren, Stephen Casner, Patrick Ruddy, Bruce Thompson, Alex Tweedly, Dan Wing, John Geevarghese, "Enhancements to IP/UDP/RTP Header Compression", <draft-koren-avt-crtp-enhance-01.txt>, March 9, 2000.

~~[5]~~[20.] "The PPP Multilink Protocol (MP)", IETF RFC 1990.

~~[5]~~[21.] " The Multi-Class Extension to Multi-Link PPP", IETF RFC 2686

~~[5]~~[22.] A Lightweight IP Encapsulation Scheme, draft-chuah-avt-lipe-02.txt, M. Chuah, E. J. Hernandez-Valencia, December 2000

~~[5]~~[23.] “Multi-Protocol Label Switching Architecture”, http://www.ietf.org/internet-drafts/draft-ietf-mpls-arch-07.txt, IETF Work in Progress.

~~[5]~~[24.] Framework Architecture for Signaling Transport, RFC 2719, October 1999

~~[5]~~[25.] Stream Control Transmission Protocol, RFC 2960, October 2000

~~[5]~~[26.] J. Loughney,G. Sidebottom, Guy Mousseau, S.Lorusso, SS7 SCCP-User Adaptation Layer (SUA), <draft-ietf-sigtran-sua-02.txt>, 04 October 2000

~~[5]~~[27.] “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, December 1998.

~~[5]~~[28.] “IPv6 Stateless Address Autoconfiguration”, RFC 2462, December 1998.

~~[5]~~[29.] “An overview of the introduction of IPV6 in the Internet”, IETF draft-ietf-ngtrans-introduction-to-ipv6-transition-04, July 2000.

~~[5]~~[30.] “Transition Mechanisms for IPv6 Hosts and Routers”, draft-ietf-ngtrans-mech-06, March 2000.“Multi-Protocol Label Switching Architecture”, http://www.ietf.org/internet-drafts/draft-ietf-mpls-arch-07.txt, IETF Work in Progress, July 2000

~~[5]~~[31.] “MPLS Support of Differentiated Services”, http://www.ietf.org/internet-drafts/draft-ietf-mpls-diff-ext-07.txt, IETF work in progress, August 2000

~~[5]~~[32.] “Tunneling Multiplexed Compressed RTP in MPLS”, http://www.ietf.org/internet-drafts/draft-theimer-tcrtp-mpls-00.txt, IETF work in progress, June 2000

~~[5]~~[33.] “Frame Relay Fragmentation Implementation Agreement, FRF.12” http://www.frforum.com/5000/Approved/FRF.12/frf12.doc.

~~[5]~~[34.] “Simple Header Compression”, draft-swallow-mpls-simple-hdr-compress-00.txt, March 2000, work in progress

~~[5]~~[35.] “PPP in a Real-time Oriented HDLC-like Framing”, RFC 2687

~~[5]~~[36.] “COPS Usage for MPLS/Traffic Engineering”, http://search.ietf.org/internet-drafts/draft-franr-mpls-cops-00.txt, July 2000, work in progress

~~[5]~~[37.] “Constraint-Based LSP Setup using LDP”, http://www.ietf.org/internet-drafts/draft-ietf-mpls-cr-ldp-04.txt, July 2000, work in progress

~~[5]~~[38.] “RSVP-TE: Extensions to RSVP for LSP Tunnels”, http://www.ietf.org/internet-drafts/draft-ietf-mpls-rsvp-lsp-tunnel-07.txt, August 2000, work in progress

~~[5]~~[39.] “MPLS/IP Header Compression”, draft-ietf-mpls-hdr-comp-00.txt, July 2000, work in progress

~~[5]~~[40.] “Comparison CIP/MPLS” , R3-010181

~~[5]~~[41.] “MPLS/IP Header Compression over PPP”, draft-ietf-mpls-hdr-comp-over-ppp-00.txt, July 2000, work in progress

~~[5]~~[42.] "User Datagram Protocol", IETF RFC 768 (8/1980)

~~[5]~~[43.] 3G TS 21.133: “3G Security; Security Threats and Requirements”.

~~[5]~~[44.] RFC 2401: “Security Architecture for the Internet Protocol”, November 1998.

~~[5]~~[45.] RFC 2408: “Internet Security Association and Key Management Protocol (ISAKMP)”, November 1998.

~~[5]~~[46.] TS 29.060, “GPRS Tunnelling Protocol (GTP)”.

~~[5]~~[47.] draft-larzon-udplite-03, “The UDP Lite protocol”.

~~1~~3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply.

3.2 Symbols

3.3 Abbreviations

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

~~1~~4 Introduction

4.1 Task Description

The work task is described in the contribution [1], which has been agreed at TSG-RAN#6. The purpose of this new work task is to enable the usage of IP technology for the transport of signalling and user data over Iu, Iur and Iub in the UTRAN.

4.2 Rationale for IP Transport

This section will describe some rationale for IP Transport option in the UTRAN.
Some mobile operators require a UTRAN transport solution for IP as an alternative to ATM.
This is partly due to the following reasons:

1. IP is developing to allow the support of a mix of traffic types and to support low speed links.
2. The popularity of the Internet/World Wide Web and corporate LANs puts price pressure on IP networking equipment.
3. IP is the technology to the “desktop” (terminals) so most applications will be based on IP.
4. Operation and maintenance networks will be based on IP. To have networks with homogeneous technology can save management and operations costs.
5. IP, like ATM, is a packet-switched technology and provides the opportunity to use transport resources in an efficient manner.
6. IP is Layer 2 independent.
7. Autoconfiguration capabilities.
8. Dynamic update of routing tables.

It’s clear that there will be IP data traffic in the mobile networks. It should be a matter of an operator‘s choice whether IP or ATM is used in the transport network to carry the various types of traffic from the circuit and packet domains.

~~1~~5 Requirements

This section detail high level requirements for the IP UTRAN option.

5.1 General requirements

Whenever possible, preference for already standardised protocols should be used, e.g. IETF protocols for the IP related parts, in order to have wide spread acceptance and avoid double work. Relevant UTRAN recommendations may also be standardised in the IETF.
By “IETF protocols”, it is meant standards RFCs and working group internet drafts.
The use of IPv6 shall not be precluded.

5.2 Independence to Radio Network Layer

The changes should only be made to the Transport Network Layer (TNL) since the Radio Network Layer should be independent of the TNL. The impact on the RNL shall be minimised but there could be some minor changes to the Radio Network Layer, e.g. addressing.
Not requiring the end point RNL user plane frame protocols to be aware of the underlying multiplexing, i.e., transparency.

5.3 Services required by the upper layers of user planes of Iu

For the Iu_CS the requirement is transfer of user data (TS25.415) and in-sequence delivery is not required.
It is a requirement that the Radio Network Layer (RNL) functional split shall not be changed depending on the TNL technology. This is in line with the architectural principle of separation of the RNL and TNL stated in [2.]. If the RNL is different for different transport technologies, backward compatibility is lost or complicated and an implementation is potentially complicated when changing transport. The RNL shall be independent from the transport type.
In order to be compatible with the release ’99 IuCS, Iur, and Iub, the following requirements for setting up transport bearers shall apply for IP transport:

The SRNC (Iu/Iur) /CRNC (Iub) TNL receives a request from the RNL to establish a bidirectional transport bearer. The request includes the end system address and transport bearer association received from the peer. It also includes the quality of service and resources required from the transport network.

5.4 Services required by the upper layers of user planes of Iur and Iub

In the current specifications the AAL2/ATM provides the services to radio network layer. The services required by the radio network layer are:

- connection identification.
- in-sequence delivery of PDUs to upper layers (TS25.425, TS25.427). If this means re-ordering of PDUs or simply not sending data that have been received out-of-sequence is not clearly stated.

It is a requirement that the Radio Network Layer (RNL) functional split shall not be changed depending on the TNL technology. This is in line with the architectural principle of separation of the RNL and TNL stated in [2.]. If the RNL is different for different transport technologies, backward compatibility is lost or complicated and an implementation is potentially complicated when changing transport.

In order to be compatible with the release ’99 IuCS, Iur, and Iub, the following requirements for setting up transport bearers shall apply for IP transport:

The SRNC (Iu/Iur) /CRNC (Iub) TNL receives a request from the RNL to establish a bidirectional transport bearer. The request includes the end system address and transport bearer association received from the peer. It also includes the quality of service and resources required from the transport network.

5.5 Coexistence of the two transport options

In Release 00, UTRAN(s) may have both ATM and IP transport networks. Following requirements with regards to ATM and IP transport network coexistence shall be met:

- The specifications shall ensure the co-existence of ATM and IP Transport options within UTRAN, i.e. parts of UTRAN using ATM and parts of UTRAN using IP transport.
- In Release 2000, ATM and IP Transport Options shall rely on the same functional split between Network Elements

The transport technology choices of an UMTS operator will vary. Some will use AAL2/ATM. Others will use IP and others will use both AAL2/ATM and IP. Interoperability between release ’99 and later UTRAN ATM interfaces and UTRAN IP interfaces (for example, IP Iur to ATM Iur) is an important function for operators deploying both types of transport networks. An interworking solution shall be included in the specification.

The following are requirements for the interworking solution:

1. It shall be possible for a UTRAN to support release ’99 and later ATM interfaces and UTRAN IP interfaces. One means of assuring that UTRAN nodes can communicate with each other is for nodes to have both ATM and IP interfaces.
2. Where Node terminating Iu, Iur or Iub does not support ATM interfaces (R99 and later releases) and UTRAN IP interfaces, an TNL interworking function shall be required to enable the nodes to inter-operate between ATM and IP technologies.

5.6 Quality of Service

The mechanisms to secure the quality of service parameters, timing aspects, and packet loss have to be considered. Quality of service parameters include service class definition and congestion control requirements. Timing aspects include delay and delay-variation requirements.

TNL shall provide the appropriate QoS requested by the RNL. However, the way the end-to-end transport network actually implements the QoS shall not be specified below IP.

Mechanisms that provide QoS or efficient bandwidth utilization must take into account UTRAN traffic (Control plane, user plane, O&M) and non-UTRAN traffic.

5.7 Efficient utilisation of transport resources

Efficient use of the bandwidth of the transport network shall be considered, e.g. by reducing the protocol overhead (via Header compression, multiplexing, …).

Iub/Iur protocols shall operate efficiently on low speed point to point links which may be shared with other traffic (e.g. GSM/GPRS Abis, UMTS R99 compliant interfaces).

The TNL shall provide the functionality of sufficiently de-coupling the bandwidth optimisation techniques such that they can be used independently of each other.

The TNL shall provide the means to enable or disable the schemes for efficient bandwidth usage (e.g. header compression, multiplexing, etc…).
In addition, for high-speed routed segments, it is important that specific bandwidth optimisation is not required at every hop.
Mechanisms that provide efficient bandwidth utilisation must take into account the QoS requirements of all UTRAN traffic (Control plane, user plane, O&M) also in case of non UTRAN traffic.

5.8 Layer 2 / Layer 1 independence

The functionality of the higher layers shall be independent from the Layer2 and Layer1 technologies. The higher layers refer both to the higher protocol layers of the Transport Network Layer and to all Radio Network Layer.
The Layer2 and Layer1 shall be capable to fulfill the QoS requirements set by the higher layers. IP Transport Flexibility
By defining protocol stacks on Iur, Iub and Iu, one may not make any restrictive assumption on IP transport network topology. They shall adapt to a wide range of networks (LAN to WAN) and no preference shall be expressed on routed vs. point to point networks.

5.9 Transport Bearer Identification

In R’99 UTRAN, ATM transport provides the ability to uniquely address individual flows. In an IP based UTRAN, the transport network has to provide the means to uniquely address individual flows – both in the user as well as signaling planes.

5.10 Transport Network Architecture and Routing

~~6.9.1~~5.10.1 Network elements

Network elements e.g. RNC, Node B need to be identified by one or more IP addresses.

5.11 Radio Network Signalling Bearer

The following are requirements on the signalling transport protocol:

1. It shall be possible for a UTRAN node to support multiple signalling bearers of different transport technologies at the same time.
2. A signalling transport shall allow multiple RNL signalling protocol entities terminating on a node to use a common physical interface.
3. A signalling transport shall provide a means of uniquely identifying the originating and terminating signalling entities.

~~1~~6 Study Areas

This section gives a summary of areas that have been identified where work needs to be performed to complete the work item.

As work proceeds in R00 with regard to IP in the UTRAN, the Work Task is divided in the following Study Areas:

6.1 External standardisation

There is a need for identifying supporting work required by other Standards Bodies. Certain protocols and /or QoS mechanisms may be indicated which are not currently supported in the industry. Appropriate liaisons should be identified. Procedure for LS's with IETF should be defined. RAN3 needs to start the IETF official communication channels.

6.2 User plane proposed solutions

This study area is intended to describe the various proposed solutions for Iur and Iub, Iu-cs and Iu-ps.

~~6.9.1~~6.2.1 CIP solution

~~6.9.1~~6.2.1.1 CIP Container

The aggregation functionality allows to multiplex CIP packets of variable size in one CIP container, also of variable size. This is necessary for an efficient use of the bandwidth of the links. It is achieved by amortising the IP/UDP overhead over several CIP packets. The resulting packet structure is depicted below:

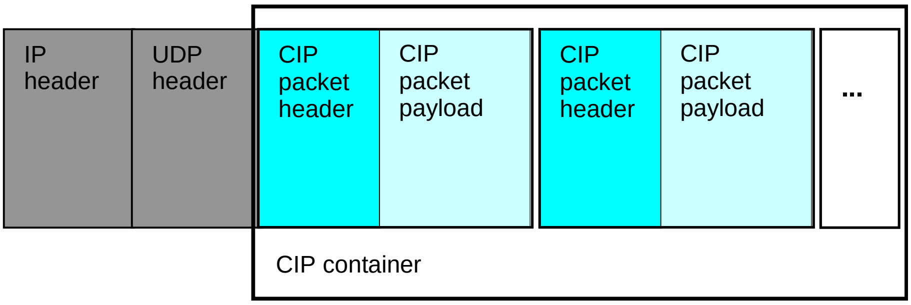


Figure 1: Generic CIP Container format

~~6.9.1~~6.2.1.2 CIP Packets

~~6.9.1~~6.2.1.2.1 Segmentation and Re-assembly

A segmentation/re-assembly mechanism allows to split large FP PDUs in smaller segments. There has to be a trade-off between efficiency (IP header / payload ratio) and transmission delay. Large data packets have to be segmented in order to avoid IP fragmentation and to keep transmission delays low.

The following figure shows the segmentation process from a FP PDU to several CIP packet payloads.

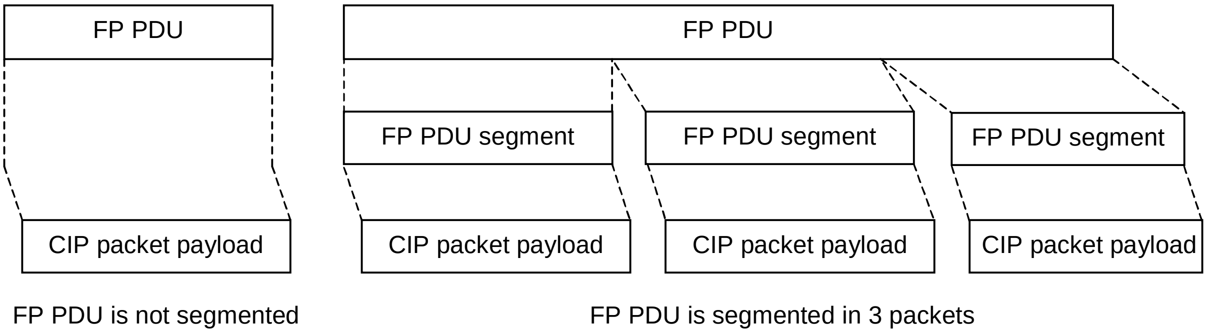


Figure 2: CIP segmentation

~~6.9.1~~6.2.1.2.2 CIP Packet Header Format

The proposed CIP packet header format is shown in the following figure.

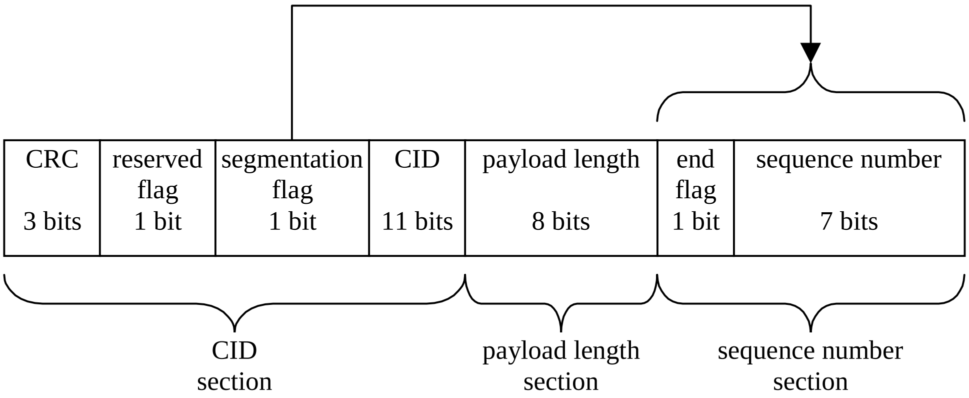


Figure 3: CIP packet header format

~~6.9.1~~6.2.1.2.3 The CIP Packet Header Fields in Detail.

The CIP packet header is composed of three sections:

1. The **CID section**, also containing CRC and flags is used for multiplexing. This section is mandatory.
    - The **CRC** protects the reserved flag, the segmentation flag and the CID.
    - The **reserved flag** is for further extensions.
    - The **segmentation flag** indicates that the sequence number field and the end flag are present. These fields are only needed for segmented packets. Because also the aggregation of non-segmented PDUs is a frequent case, e.g. voice, these fields can be suppressed by means of the segmentation flag to save bandwidth.
    - The **CID** is the Context ID. This is the identifier of the multiplex functionality, e.g. to distinguish the flows of different calls or users by the higher layers.
2. The **payload length section** is used for aggregation. This section is mandatory.
    - The **payload length** is the length of the CIP packet payload. So, CIP packets, containing e.g. FP-PDUs with voice or FP-PDU segments with data, can be between 1 and 256 octets in size.
3. The **sequence number section**, also containing the end-flag is used for segmentation. This section is optional. It exists if the segmentation flag is set.
    - The **end-flag** marks the last segment of a packet in a sequence of segments. This field is only present if the segmentation flag is set.
    - The **sequence number** is to reassemble segmented packets. This field is only present if the segmentation flag is set. It is incremented for each segment (modulo) and is not reset if the segments of a new packet start. The sequence numbers are maintained for each CID individually.

~~6.9.1~~6.2.1.2.4 Discussion of the CIP Packet Header Field Sizes

One aim is to have byte aligned boundaries where possible. So, adding a few bits to some fields would increase the header size by at least 1 byte. The proposed CIP packet header has a length of 3 bytes for non-segmented packets and 4 bytes for segmented packets.

- The **CID field** size determines how many flows between a pair of network elements can be supported at the same time. The proposed size of 11 bits allows 2048 CIDs. This is more than 8 times the amount that AAL2 offers. It can be extended by additional UDP ports, each having its own CID address space.
- The size of the **Payload Length** field. This choice determines the maximum size of a CIP packet payload, containing either a whole FP-PDU or a segment of a FP-PDU. Typically, these packets are either small by nature or they are made small intentionally. So, to stay on byte boundaries, the length field for the CIP packet payload size is proposed to be 1 byte.
- The size of the **Sequence Number** field determines in how many segments a FP PDU can be split before this modulo-incremented field wraps around and becomes ambiguous. The proposed size is 7 bits i.e. 128 segments. One bit has to be reserved for the end-flag. These two fields are combined together

because they are both optional and are needed only in case of segmentation. The segment numbers also protect segments that arrive late, from being injected in the next packet with the same CID during the reassembly process. This is the reason why the segment numbers are counted modulo over the full range and do not start with 0 at every new FP PDU. A very worst case scenario with a 2Mbit/s source would deliver 20480 bytes within 80 ms. If this PDU is cut to pieces of 256 bytes, 80 segments would result.

- The size of the **CRC** depends on how many bits need protection. A bit error in the length field would interpret the wrong bytes as the next header. But this can be detected, because the next header is again protected by its own CRC. So, the payload length needs no protection. An error in the sequence number would be detected by either placing a segment in a position where another segment with the same number already is, or would be regarded as 'too late' because it belongs to the segment number range of a PDU already processed. Even if the segment is injected in the wrong place, it would be detected by a checksum error of the higher layer. So, the only fields that need protection are the flags and the CID. An error in the CID is critical, because it would inject a formally correct (non-segmented) PDU in the flow to another CID, i.e. to the wrong destination. This might be difficult to detect by the higher layer, because the CID is not a part of the PDU of the higher layer. And so, the CRC of the higher layer alone is not a sufficient protection mechanism against the erroneous injections of formally correct PDUs. For the 13 bits to be protected, a 3 bit CRC seems to be sufficient.

~~6.9.1~~6.2.2 LIPE solution

The LIPE scheme uses either UDP/IP or IP as the transport layer. Each LIPE encapsulated payload consists of a variable number of multimedia data packet (MDP). For each MDP, there is a multiplexing header (MH) that conveys protocol and media specific information.

The format of an IP packet conveying multiple MDPs over UDP using a minimum size MH is below:

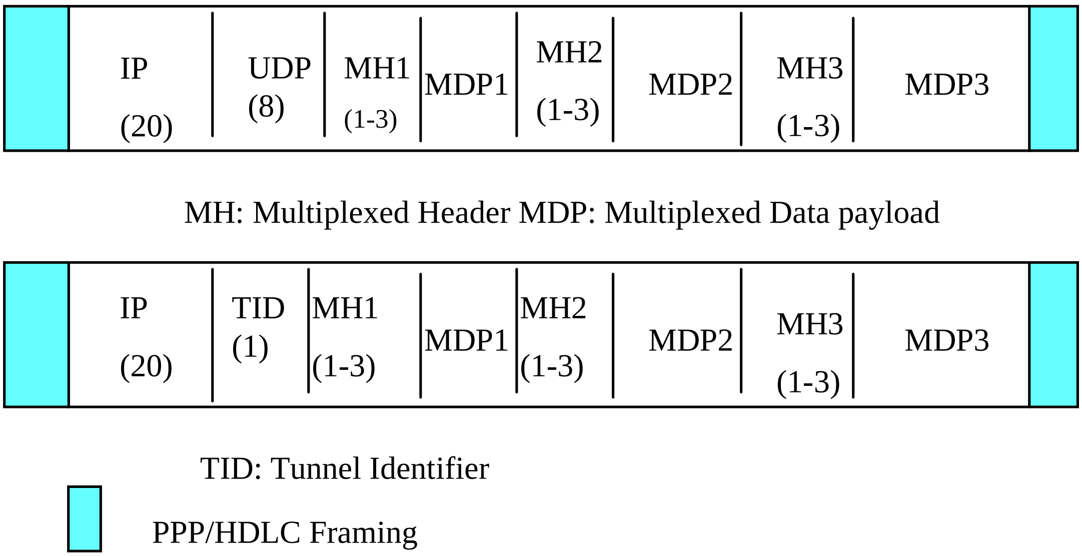
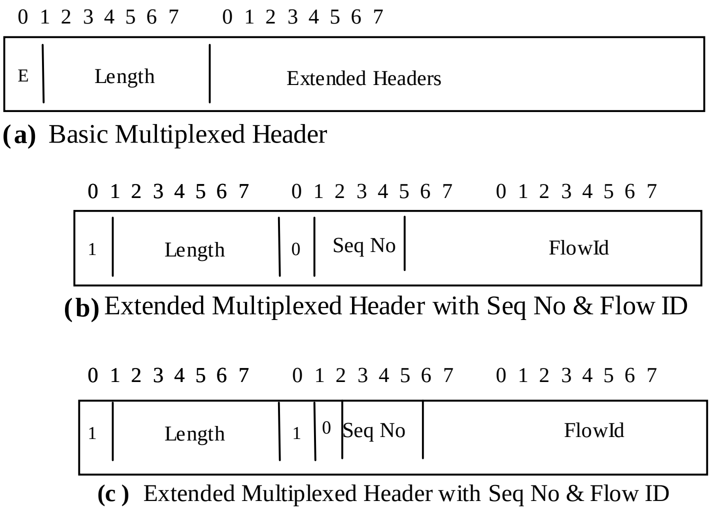


Figure 4: LIPE UDP/IP or IP Encapsulation Format

Figure 4 shows the encapsulation format of a LIPE packet. Details of the multiplexed header is described in the next section.

~~6.9.1~~6.2.2.1 Details of Multiplexed Header



(a) Basic Multiplexed Header

(b) Extended Multiplexed Header with Seq No & Flow ID

(c) Extended Multiplexed Header with Seq No & Flow ID

Figure 5: Formats of Multiplexed Header

~~6.9.1~~6.2.2.2 Basic Header

The Multiplexing Header (MH) comprises of two components: The extension bit (the E bit) and the MDP length field. Optional Extension Headers can be supported via the E bit. The MH format is shown in Figure 2 (a). The E bit is the least significant bit of the first byte of the MH header. It is set to one/zero to indicate the presence/absence of an extension header. If the E-bit is set to one, the first header extension MUST be a Extended Header Identifier field. The Length filed is 7 bit. This field indicates the size of the entire MDP packet in bytes, including the E bit, the length field and optional extension headers (if they exist).

~~6.9.1~~6.2.2.3 Extensions

Extension headers are used to convey user specific information. It also facilitates the customization of LIPE to provide additional control information e.g. sequence number, voice/video quality estimator.

The 16-bit EHI is the first field in any Extension Header. It is used to identify MDPs belonging to specific user flows. The format of a LIPE encapsulated payload with a FlowID extension header is shown in Figure 2 (b). The least significant bit of the 1st byte of EHI is the X-bit. When the X-bit is clear, it means there is a 3 bit header SEQUENCE NO. and a 12 bit FlowId. When the X bit is set to one, it indicates that the EOF bit and the 3 bit Seq Number fields exist and that the FlowID field is 11 bit. The second least significant bit is the end of fragment (EOF) indicator. When EOF is set to 0, it means this is the last fragment (for packets that are not fragmented, this bit is always 0). When EOF is set to 1, it means there are more fragments coming.

~~6.9.1~~6.2.3 PPP-MUX based solution

6.2.3.1 PPP Multiplexed Frame Option Over HDLC

PPP Multiplexing (PPPmux) [10.], Figure 6, provides a method to reduce the PPP framing [11.][12.] overhead used to transport small packets, e.g. voice frames, over slow links. PPPmux sends multiple PPP encapsulated packets in a single PPP frame. As a result, the PPP overhead per packet is reduced. When combined with a link layer protocol, such as HDLC, this offers an efficient transport for point-to-point links.

At a minimum, PPP encapsulating a packet adds several bytes of overhead, including an HDLC flag character (at least one to separate adjacent packets), the Address (0xFF) and Control (0x03) field bytes, a two byte PPP Protocol ID, and the two byte CRC field. Even if the Address and Control Fields are negotiated off and the PPP Protocol ID is compressed, each PPP encapsulated frame will include four bytes of overhead. This overhead can be reduced to one or two bytes.

The key idea is to concatenate multiple PPP encapsulated frames into a single PPP multiplexed frame by inserting a delimiter before the beginning of each frame. Each PPP encapsulated frame is called a PPP subframe. Removing the PPP framing characters can save several bytes per packet, reducing overhead.

During the NCP negotiation phase of PPP, a receiver can offer to receive multiplexed frames using a PPP Mux Control Protocol (PPPMuxCP). Once PPPMuxCP has been negotiated, the transmitter may choose which PPP frames to multiplex. Frames should not be re-ordered by either the transmitter or receiver regardless of whether they arrive as part of the PPP multiplexed frame or by themselves.

The PPP Protocol ID field of a subframe can be removed if the PPP Protocol ID of that subframe is the same as that for the preceding subframe. A Protocol Field Flag (PFF) bit and a Length Extension (LXT) field is defined as part of the length field (thus reducing the length field from an 8-bit to a 6- bit field). The PFF bit is set if the PPP Protocol ID is included in the subframe. The PFF bit is cleared if the PPP Protocol ID has been removed from the subframe. The PFF bit may be set to zero for the first subframe in a PPP multiplexed Frame if the Protocol ID is the same as the default PID, as specified by the PPPMuxCP option. The transmitter is not obligated to remove the PPP Protocol ID for any subframe.

The format of the complete PPP frame along with multiple subframes is shown in Figure 4. Note that regardless of the order in which individual bits are transmitted, i.e. LSB first or MSB first, the PFF bit will be seen to be the MSB of a byte that contains both the PFF and the subframe length field.

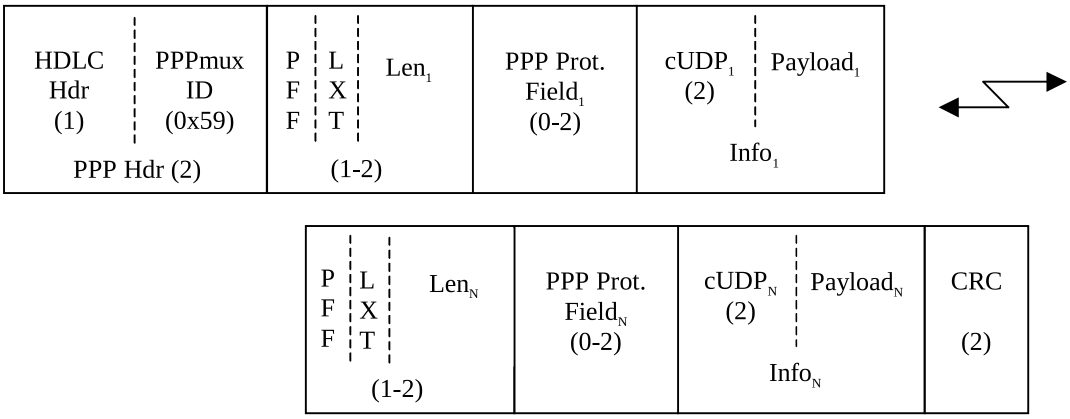


Figure 6: PPPMux frame with multiple subframes

PPP Header: The PPP header contains the HDLC header and the PPP Protocol Field for a PPP Multiplexed Frame (0x59). The PPP header compression options (ACFC and PFC) may be negotiated during LCP and could thus affect the format of this header.

Protocol Field Flag (PFF): This one bit field indicates whether the PPP Protocol ID of the subframe follows the subframe length field. PFF = 1 indicates that the protocol field is present for this subframe. PFF = 0 indicates that the protocol field is absent for this subframe. If PFF = 0 then the PPP Protocol ID is the same as that of the preceding subframe with PFF = 1, or it is equal to the default PID value of the PPPMuxCP Option for the first subframe.

Length Field:

The length field consists of three subfields:

1. Protocol Field Flag (PFF):
   The PFF refers to the most significant bit of the first byte of each subframe. This one bit field indicates whether the PPP Protocol ID of the subframe follows the subframe length field. For the first subframe, the PFF bit could be set to zero if the PPP protocol ID of the first subframe is equal to the default PID value negotiated in PPPMuxCP. PFF = 1 indicates that the protocol field is present (and follows the length field) for this subframe. PFF = 0 indicates that the protocol field is absent for this subframe. If PFF = 0 then the PPP Protocol ID is the same as that of the preceding subframe with PFF = 1, or it is equal to default PID value of the PPPMuxCP Option for the first subframe. The transmitter is not obligated to remove the PPP Protocol ID for any subframe.

2. Length Extension (LXT):
   This one bit field indicates whether the length field is one byte or two bytes long. If the LXT bit is set, then the length field is two bytes long (a PFF bit, a length extension bit, and 14 bits of sub-frame length). If the LXT bit is cleared, then the length field is one byte long (a PFF bit, a length extension bit, and 6 bits of sub-frame length).

3. Sub-frame Length (LEN):

This is the length of the subframe in bytes not including the length field. However, it does include the PPP Protocol ID if present (i.e. if PFF = 1). If the length of the subframe is less than 64 bytes (less than or equal to 63 bytes), LXT is set to zero and the last six bits of the length field is the subframe length. If the length of the subframe is greater than 63 bytes, LXT is set to one and the last 14 bits of the length field is the length of the subframe. The maximum length of a subframe is 16,383 bytes. PPP packets larger than 16,383 bytes will need to be sent in their own PPP frame. A transmitter is not required to multiplex all frames smaller than 16,383 bytes. It may chose to only multiplex frames smaller than a configurable size into a PPP multiplexed frame.

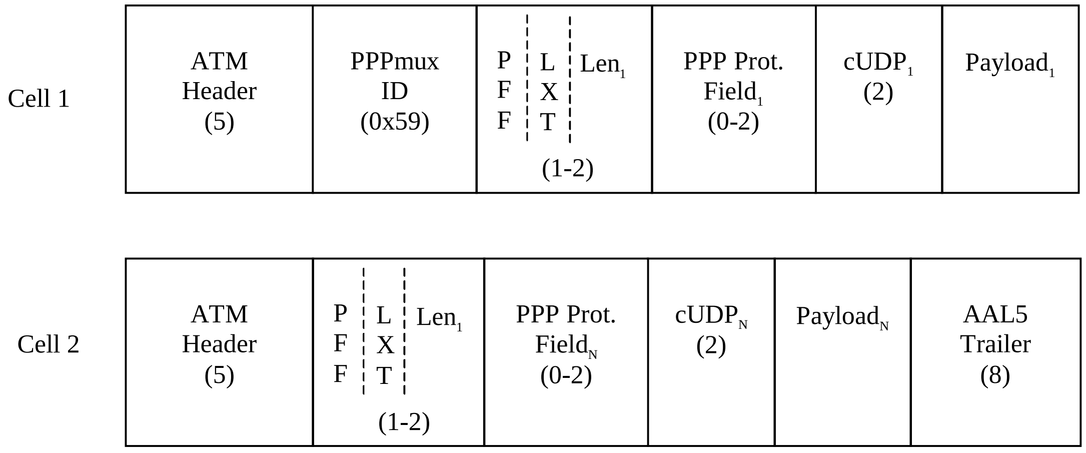
Protocol Field: This field contains the Protocol Field value for the subframe. This field is optional. If PFF = 1 for a subframe, the protocol field is present in the subframe, otherwise it is inferred at the receiver. The receiver MUST support Protocol-Field-Compression (PFC) for PPP Protocol IDs in this field. Thus the field may be one or two bytes long. The transmitter SHOULD compress PPP Protocol IDs in this field that have an upper byte of zero (i.e. Protocol IDs from 0x21 thru 0xFD). This Protocol Field Compression is not related to the negotiation of PFC during LCP negotiation, which affects the length of the PPP Multiplexed Frame Protocol ID.

Information Field: This field contains the actual packet being encapsulated. Any frame may be included here with the exception of LCP Configure Request, ACK, NAK and Reject frames and PPP multiplexed frames. If LCP is renegotiated, then PPP Multiplexing MUST be disabled until PPP Mux Control Protocol is negotiated.

In the proposed protocol stack the Information Field is comprised of a compressed IP/UDP (cUDP) [12.][13.] header (with a minimum length of 2 bytes and maximum of 5 bytes) and the payload of the packet. The PPPMuxCP default PID is 0x67, corresponding to cUDP. (A 2-byte cUDP header assumes an 8-bit CID and no UDP checksum.)

6.2.3.2 PPP Multiplexed Frame Option Over ATM/AAL5

This protocol stack uses the same PPPmux option as described above, but carries PPP over an ATM/AAL5 link layer [14.][15.], Figure 7. Here the HDLC header and CRC trailer is replaced with an ATM header and AAL5 trailer.



[Editor's note: Payload position needs to be fixed]

Figure 7: PPPMux over an ATM/AAL5

6.2.3.3 PPP Multiplexed Frame Option Over L2TP Tunnel (TCRTP)

In cases where a routed WAN interface is required, one may still use PPPmux, but tunnel it via L2TP [16.]. This protocol is called Tunnelled Compressed RTP (TCRTP) [17.],Figure 8.

L2TP tunnels should be used to tunnel the cUDP payloads end to end. This is a natural choice since cUDP payloads are PPP payloads, and L2TP allows tunnelled transport of PPP payloads. L2TP includes methods for tunnelling messages used in PPP session establishment such as NCP. This allows the procedures of RFC 2509 to be used for negotiating the use of cUDP within a tunnel and to negotiate compression/decompression parameters to be used for the cUDP flow.

A companion draft [18.] describes a method of compressing L2TP tunnel headers from 36 bytes (including the IP/UDP/L2TP headers) to 21 bytes. L2TPHC packets include an IP header, using the L2TPHC IP protocol id. The UDP header is omitted, and the L2TPHC header is reduced to 1 byte. The added overhead is now 21 bytes of the IP header. Enhancements to CRTP [19.] are not needed for cUDP header compression.

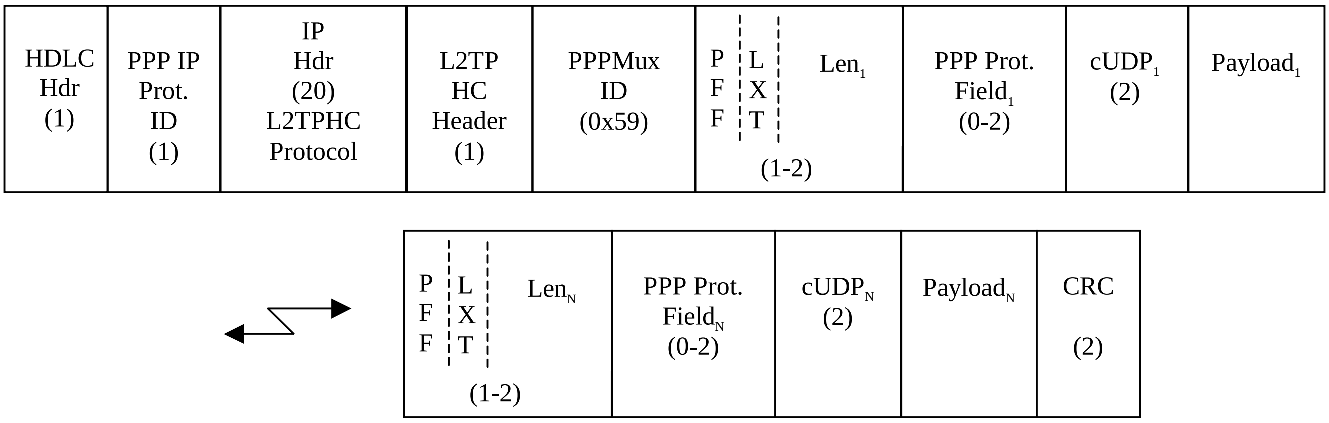


Figure 8: PPPmux tunnelled over Routed Network using L2TPHC (with PPP as Layer 2)

A more bandwidth efficient way to send TCRTP over a PPP link is to compress the L2TP IP header with cUDP (this is referred to as cTCRTP).

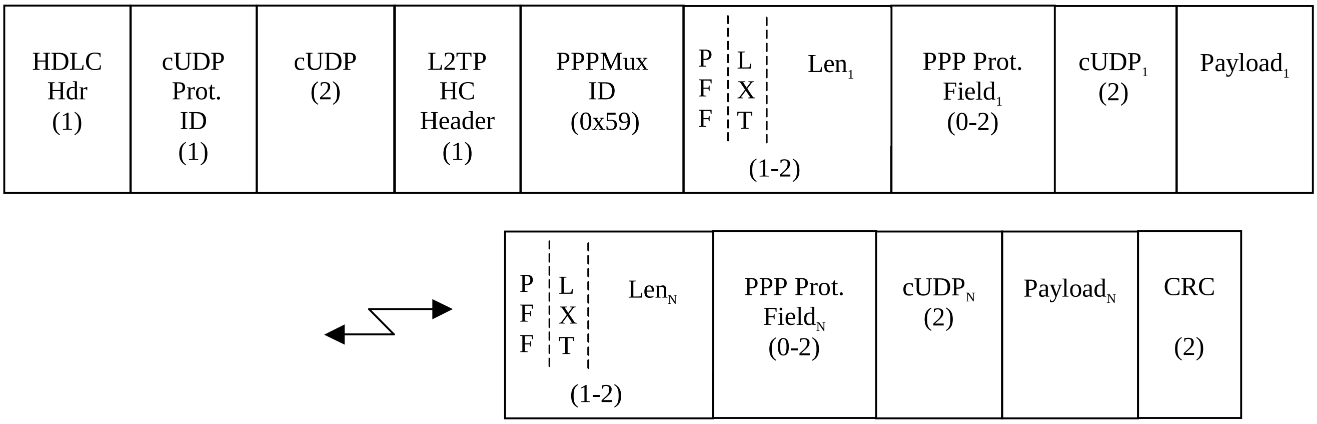


Figure 9: cTCRTP PPPMux packet tunnelled in L2TPHC over a PPP link

~~6.9.1~~6.2.4 MPLS solution

[Editor's note: Detailed reference to RFCs and other standards need to be provided, and overheads need to be calculated again according to the detailed references.]

~~6.9.1~~6.2.4.1 MPLS General Description

The Multi-Protocol Label Switching (MPLS) protocol is an interstitial, layer 2.5 protocol which complements and enhances the IP protocol, in that it offers an alternative method of forwarding IP packets, while reusing the existing IP routing protocols (e.g., OSPF, BGP).

MPLS can run on top of numerous L2 technologies (PPP/Sonet, Ethernet, ATM, FR, WDM Lambdas, etc.) .

MPLS forwards IP packets based on a 20-bit label. An ingress router at the edge of an MPLS domain, called a Label Edge Router, decides which subset of incoming packets is to be mapped to which Label-Switched Path (LSP), and then adds the corresponding label to each packet as it arrives. This subset of packets that is forwarded in the same manner over the same LSP is called a Forwarding Equivalence Class (FEC). Packets are then forwarded through the MPLS domain by the Label Switched Routers (LSRs) based on the label. At the egress edge of the MLS domain, the egress LSR removes the MPLS label from each IP packet, and subsequently the IP packets are forwarded by conventional IP forwarding.

Each pair of LSRs on the label-switched path (LSP) must agree on which label to use on that segment of the LSP. This agreement is achieved by using a set of procedures, called a label distribution protocol. The label distribution protocol associates a Forwarding Equivalence Class (FEC) with each LSP it creates. The FEC associated with an LSP specifies which packets are "mapped" to that LSP.

~~6.9.1~~6.2.4.2 Routing with MPLS

MPLS, as a complementary forwarding technique to IP forwarding, offers the following advantages :

- **Coexistence with IP Hop-By-Hop Routing.** An LSR is capable of forwarding both IP packets and MPLS frames.
- **Traffic engineering capabilities** : MPLS uses the label prefixed to an IP packet to determine the path that the packet will take through the network, regardless of the IP addresses contained in the packet. Routes through the network can be engineered to meet various network or operator requirements (such as QoS or trafic load). For example, the traffic at the edge of the MPLS domain can be segregated according to QoS class and the packets can be directed along the MPLS paths defined over the route that meets their QoS requirements (see QoS section hereafter).
- **Flexibility due to label semantics**. The meaning of the labels can be tailored to what needs to be achieved in the network. For example, labels can be used to specify treatment for QoS, multiplexing, multicasting, header compression, etc.
- **Flexibility due to label stacking**. MPLS supports the ability to stack more than one label in front of an IP packet. LSRs are capable of pushing, popping and swapping labels. This allows for :
  - □ Different addressing in different subnets
  - □ Efficient inherent support for tunnels-in-tunnels. This can be used, for example, for IP VPN and mobility support.
- **Transparent routing :** the compressed packet passes transparently through the intermediate LSRs. This is in contrast to schemes based, for example, on PPP where either header (de-)compression must occur on a hop-by-hop basis or the compressed packets must be carried inside a second, uncompressed IP tunnel packet. MPLS thereby makes network nodes much simpler.
- **Fast rerouting** MPLS protection switching mechanisms can be applied to achieve fast restoration from a node failure. Both local and end-end protection could be used to achieve fast tunnel restoration which is an essential requirement for a carrier grade network. Backup tunnels may also be combined with load sharing to allow a more even traffic distribution.
- **Match any layer 2 :** MPLS can run on top of numerous L2 technologies. When MPLS is used over ATM or Frame Relay, the LSP can be mapped onto layer 2 connections such as VCCs or PVCs.

~~6.9.1~~6.2.4.3 Support for QoS requirements

Finally, the MPLS supports a number of QoS differentiation mechanisms for IP flows :

- **QoS engineered paths.** The flows with different QoS characteristics can be separated on different LSPs. LSPs can be engineered to meet the QoS requirements for each class of traffic supported by the network. The traffic at the edge of the MPLS domain can be segregated according to QoS class and the packets can be directed along the MPLS paths defined over the route that meets their QoS requirements.
  Taking again our example over narrow-band links, QoS efficient LSPs could pave the way for real-time flows whereas user data with long payloads could be routed over separate LSP(s). By so doing, there is no risk to have big packets blocking the way of delay-sensitive small packets. Best efficiency can be achieved by combining the use of MPLS with the appropriate layer 2 mechanisms depending the technology used at layer 2. Taking again our example with ATM over such narrow-band links, the different LSPs (i.e. VCCs) are multiplexed onto the same physical link by the ATM VCC multiplexing function respecting the VCC QoS, thus the LSP Qos. Then QoS characteristics of real- time flows (such as IP Diffserv marking) can be used to select the LSP (i.e. the ATM VCC) the packet should be sent over. This is fairly easy to achieve through the VPI/VCI - label mapping defined above.
- **Integration with Differentiated Services (DiffServ)** DiffServ provides a mechanism for defining the treatment that a packet will receive as it is forwarded through an IP network. Although there are no performance guarantees with DiffServ, it can be used to improve end-to-end performance over large

scale, wide area networks. MPLS can support DiffServ by using the DiffServ marking in each packet to determine:

- □ which path the packet should be sent over. Paths can then be engineered, as mentionned above, to provide more deterministic performance guarantees than are available with pure DiffServ in a routed network.
- □ the treatment that packets will receive over a specific path. In this model, closely resembling the basic DiffServ model, packets with different QoS requirements can be carried over the same MPLS path. Within that path, the DiffServ marking is used to prioritise and schedule packets to provide "better" treatment for some packets with respect to other packets carried over that same path.

- o **In-Sequence Packet Delivery.** Because the route that a packet will travel through the network is precisely defined by the Label Switched Path, packets are guaranteed to be received in the same order that they were transmitted.

~~6.9.1~~6.2.4.4 Efficient, QoS-enabled transmission over routed domains with MPLS

Let us consider a general network configuration, which includes a broadband routed cloud as well as a narrowband link, typically on the last-mile link to the Node B. This configuration is shown in Figure 10.

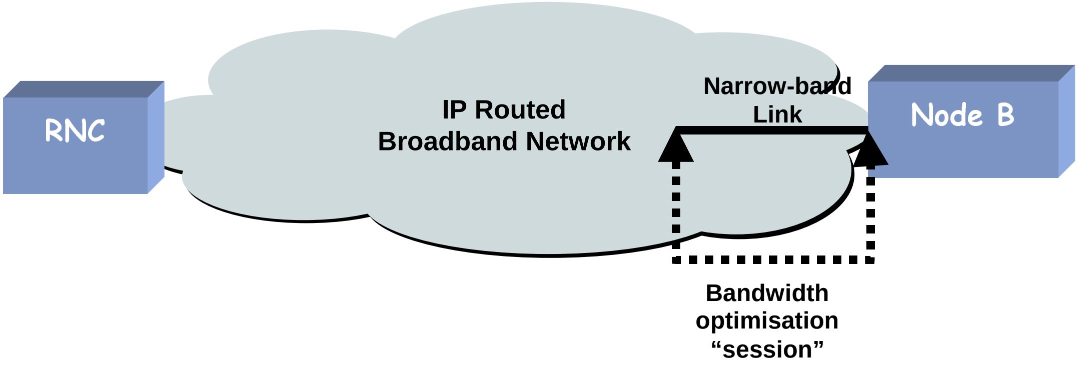


Figure 10 General UTRAN network configuration

Figure 1 also shows the most likely location of the pair of endpoints for a bandwidth optimisation "session". In this manner, bandwidth optimisation is only performed where it is really required, on the narrow-band, point-to-point link.

Figure 11 shows the protocol stacks at the relevant nodes in the network for an MPLS-based transport solution over a routed domain. On the downlink, UDP/IP packets are mapped onto MPLS paths at the RNC, and are sent uncompressed through the network to a compressing/decompressing node (CDN). The UDP/IP packets are then compressed using a technique defined in section 6.2.4.5.1, and sent compressed over the narrow-band point-to-point link. At the Node B the UDP/IP packets are restored/uncompressed. On the uplink, UDP/IP packets are compressed and sent over the narrow-band link. At the CDN, packets are uncompressed and mapped onto an MPLS path for transport to the RNC. Because the MPLS label attached to the compressed packet is used to route the frame through the network, the CDN can be located at the RNC or at any point along the path to the Node B that has sufficient processing capacity for handling the CDN functions.

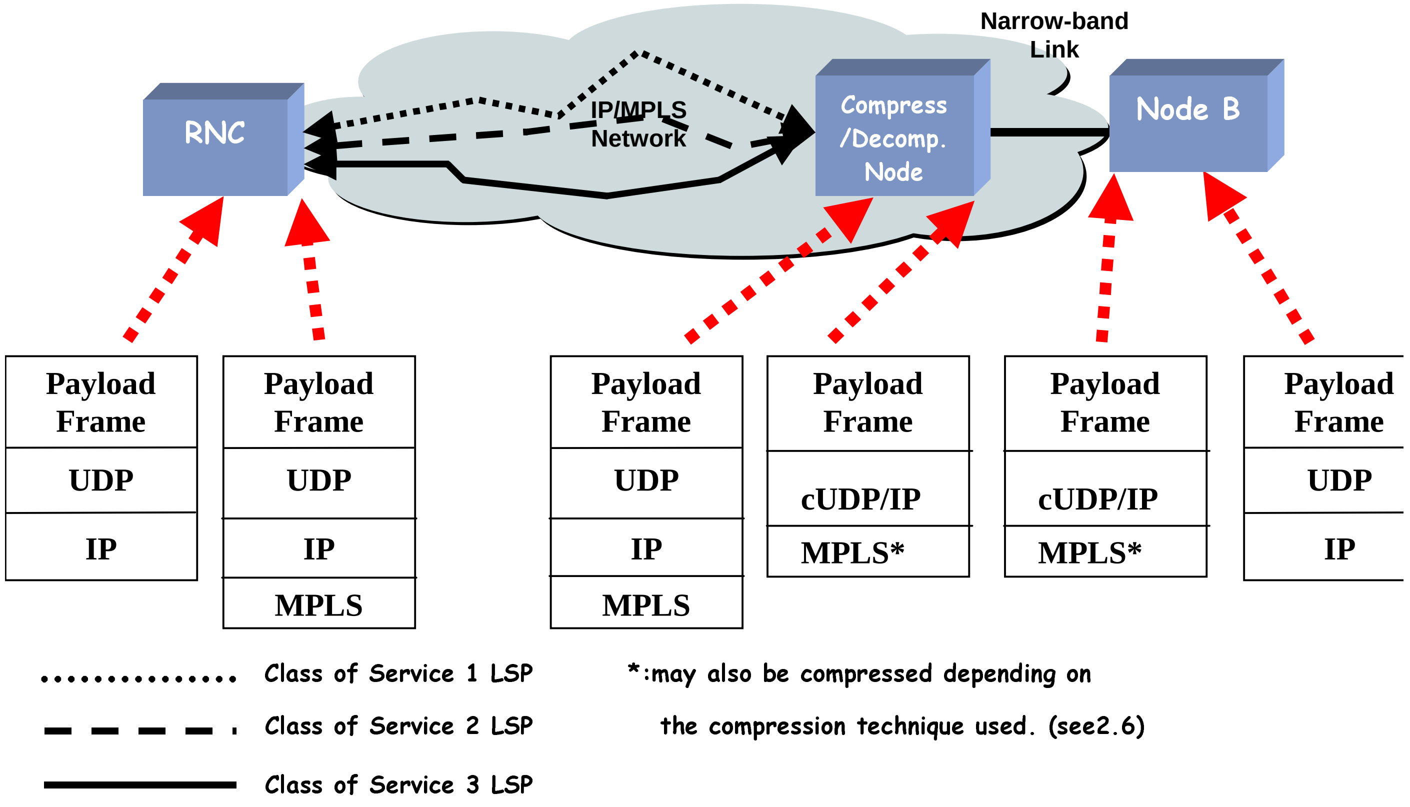


Figure 11 Protocol stacks at key nodes in the network for a MPLS-based transport solution

An MPLS-based transport solution for the UTRAN, integrated with DiffServ (or DiffServ-like) mechanisms, includes the following:

- Label-Switched Paths (LSPs) are established between an RNC and a Node B, in both directions; each LSP carries one or more class of service supported by the UTRAN. This occurs during NodeB initialisation, before user traffic is allowed to flow through the NodeB. LSPs can be pre-setup via provisioning (e.g., using COPS MPLS [40.]), or set up dynamically using CR-LDP [37.]or RSVP-TE [38.]As part of this process of setting up the LSPs, all the intermediate transit routers are provisioned to provide the desired per-hop behaviour (i.e., scheduling treatment and in some cases, drop precedence for each DS code point). By providing consistent behaviour to packets belonging to the same class of service in each transit node which is part of an LSP, the overall quality of service in that LSP is achieved. This is consistent with the approach described in [31.]
- The operator decides how many classes of service there will be supported in the UTRAN, and also how classes of service map to an LSP (i.e., one or more).
- An IP packet is mapped to the LSP with appropriate class of service based on two things: the DS code point marking in the IP header of the packet, and the FEC that the packet belongs to, (i.e. the destination IP address in the IP header). This is also consistent with [31.].
- IP packets are mapped to the appropriate LSPs at the UTRAN edge nodes, i.e., the RNCs and Node Bs.

~~6.9.1~~6.2.4.5 Efficient transmission over narrowband (point-to-point) links with MPLS

Compression of UDP/IP headers is compatible with the use of MPLS in order to provide optimized efficiency on narrow-band links. As an example, two types of techniques are currently under investigation over PPP links and available as internet drafts :

“simple IP header compression” [34.] where the emphasis is put on the flexibility on the point where the compression and decompression nodes are located : compression can be performed between any two LSRs on the LSP including compressing over the complete LSPs. In that case the compressed frame is routed through the LSP with the MPLS label. This technique is based on differential coding compared to a static template which presents the advantage of robust

synchronization between compressor and decompressor even in case of lost frames. The bandwidth efficiency calculation leads to overheads (layer 2 + layer 3) of 9 bytes per user flow.
“MPLS+IP Header compression” [39.] where the compression is only performed on a point-to-point link (such as UTRAN last mile) and the emphasis is further put on MPLS header compression. In that case, the MPLS label is compressed by sharing the UDP/IP compression context. Bandwidth efficiency is further improved by using the same differential coding as introduced in [40.]. This differential coding scheme transmits the changes between successive packets in order to keep the size of the compressed fields small. The resulting overhead (layer2 + layer3) is 7 bytes per user flow.

The detailed calculations and the comparison of bandwidth efficiency on the last mile for the different alternatives is addressed in the document [40.]. The optimization between the two techniques could be left to network engineering.

Header compression also implies a previous negotiation between the compressor and decompressor. As an example, the following section describes how this negotiation is performed for one of the above defined compression techniques over PPP [34.]. The equivalent for the second one can be found in [41.].

~~6.9.1~~6.2.4.5.1 MPLS Header Compression “Session Negotiation”

As with other header compression techniques, a header compression session negotiation is required. Here are two examples of how this can be done:

1. Using RSVP-TE messages to negotiate the header compression [34.]

2. Using the Label Distribution Protocol (LDP) to negotiate the header compression.

A fundamental concept in MPLS is that two LSRs must agree on the meaning of the labels used to forward traffic between and through them. This common understanding is achieved by using a set of procedures, called a label distribution protocol, by which one LSR informs another of label bindings it has made.

The Label Distribution Protocol, LDP [8] describes one of the label distribution protocols, by which LSRs distribute labels to support MPLS forwarding along normally routed paths. An extended version of RSVP [38.] can also be used to define and distribute labels.

~~6.2.4.5.1.2~~6.2.4.5.1.1 Using RSVP-TE to negotiate “MPLS Simple Header Compression”

The internet draft “Simple Header Compression” [34.] describes a way of negotiating a MPLS Header Compression session using RSVP-TE signalling. The compressor endpoint sends an RSVP PATH message to request an MPLS header compression session. The decompressor replies with an RSVP RESV message confirming that it will perform the decompression.

The compressor includes a SIMPLE_HEADER_COMPRESSION (SHC) RSVP object in the PATH message to communicate the header template and the set of operands. To allow multiplexing across an LSP the SHC objects also carry a one byte sub-context ID (SCID)

The decompressor includes a SIMPLE_HEADER_COMPRESSION_REPLY RSVP object in the RESV message to indicate which SCIDs it is agreeing to decompress.

The template in the SHC object consists of the first n bytes of a packet. All of the fixed fields are set to their appropriate values. The variable fields are set to zero. Fields are always delimited on byte boundaries. Each operand is simply an offset and a length. They serve to delimit the variable fields within the template.

Instructions on what to do with the variable fields (e.g., IP TTL, IP checksum, and IP length) is also signalled in the SHC object, using the T, C, and L flags, respectively.

The compressor removes the header from the packet. The term header is used loosely here. It refers to the first n bytes of the packet where n is the length of the header template. The compressor uses the operands to extract the variable fields from the header. These are concatenated together as a compressed header. The SCID is then prepended to the compressed header and the packet is sent.

The decompressor uses the incoming MPLS label and the SCID to locate the proper decompression context. The decompressor then uses the header template to reconstruct the original header. It uses the operands to populate the variable fields of the header with the contents of the compressed header.

Over the life of an RSVP session SCIDs may be added and deleted simply by refreshing the Path state with the updated set of SHC objects The SHCR object provides synchronization between the sender and receiver as to which SCIDs may be used.

6.2.4.5.1.2 Using LDP signalling for “MPLS Simple Header Compression” session negotiation

MPLS Header Compression session negotiation can be accomplished with the LDP protocol, by adding a new TLV (Type-Length-Value) that includes the header template, flags and set of operands as described in section 6.2.4.5.1.1.

The compressor requests a label for a new IP flow (i.e., 5-tuple combination source IP address, source port, destination IP address, destination port, protocol id} via the downstream on-demand method from the decompressor, which is its LDP peer in this case. The decompressor provides the MPLS label it wants to use for this FEC back to the compressor. The decompressor also stores the mapping of MPLS label to header template+flags+operands in a local table. The compressor also specifies how the IP TTL, IP checksum, and IP length fields are to be regenerated on the other end in the FEC TLV.

The compressor LSR can then compress the IP packets as per section 6.2.4.5.1.1. When the decompressor LSR receives the MPLS frame, it looks up the MPLS label in the mapping table, and uses this information to restore the UDP/IP header.

~~6.9.1~~6.2.4.5.2 Handling of large packets over narrowband links

In general, sending a large packet over a narrowband link will cause delays to subsequent real time packet(s) that would impact the QoS of the real time packet(s). Fragmenting large packets into smaller sub-packets, and then scheduling all the packets to be sent over a link (including the sub-packets) according to their QoS requirements generally solves this problem.

When MPLS is used in a UTRAN transport solution, the fragmentation can be localised over the narrowband link by relegating it to the underlying layer 2:

- ATM can provide this with AAL-5
- Multi_Link PPP can provide this [20.]
- Multi-class extension of Multi-Link PPP can provide this [21.]
- HDLC can provide this with PPP in a Real-time Oriented HDLC-like Framing [35.]
- Frame Relay can also provide this [33.]

~~6.9.1~~6.2.5 AAL2 based solution

If it is determined by RAN3 that a protocol should be used for multiplexing and/or fragmentation between the IP layer and the RNL, the AAL2 (SSSAR and CPS) user plane protocol should be used over UDP.
AAL2/UDP should be used for multiplexing and fragmentation between the IP layer and the RNL for the following reasons:

1. Using AAL2 makes interoperability between IP and AAL2/ATM nodes easier.
2. Fragmentation and multiplexing standards already exist.
3. Fewer protocols need to be supported in a UTRAN node.
4. AAL2/UDP will be terminated in the UTRAN end node.

Some changes could be made to the existing AAL2 protocol:

1. It’s not necessary to limit the UDP packet size to 48 bytes as it is for ATM.
2. There is no reason to split an AAL2 SDU between two UDP packets as is done with ATM. As a result there should be no reason for the AAL2 Start field.

~~6.9.1~~6.2.6 Usage of UDP Lite for IP UTRAN

~~6.9.1~~6.2.6.1 Background

There are a number of link technologies where data can be partially damaged. Microwave transport is one common example. For some applications, such as voice, better performance can be achieved if errored data is not discarded but is instead delivered to the application.
The current ATM UTRAN allows bit errors in the payload to be passed to the application. This is because:

- ATM only protects the ATM header with a Header Error Control (HEC) field.
- AAL2 only protects the AAL2 header with an HEC field.
- AAL2 also provides support for error detection for the payload in I.366.1. This is not used in the UTRAN, however.
- The UTRAN framing protocols include a checksum for the headers and an optional checksum for the payload.

In IPv4, the UDP checksum either covers the entire datagram or is not used at all. In IPv6, the UDP checksum is mandatory and can not be disabled. The IPv6 header does not have a header checksum so the UDP checksum was made mandatory in order to protect the IP addressing information. This means with classic UDP the entire packet must be covered for IPv6.

It would be beneficial if the error detection mechanism of the transport layer could protect vital information such as headers and to optionally ignore errors best handled by the application.

~~6.9.1~~6.2.6.2 UDP Lite

UDP Lite is an IETF Working Group draft. It provides a partial checksum that improves the flexibility over classic UDP by making it possible to define the part of a packet to be protected by the checksum.

The UDP Lite header is shown in the figure below.

0 15	16 31
Source Port	Destination Port
Checksum Coverage	Checksum
Data bytes ...	

Its format differs from classic UDP in that the UDP “Length” field has been replaced with a “Checksum Coverage” field. Information about the UDP Lite packet length can be found in the length field of the IP header so the packet length information in UDP is not required.

The fields ``Source Port'' and ``Destination port'' are the same as classic UDP (RFC-768) [42.].

“Checksum Coverage” is the number of bytes that are covered by the checksum beginning with the first byte of the UDP Lite header. A “Checksum Coverage” of zero indicates that the entire UDP Lite packet is included in the checksum.

“Checksum” is a checksum over a pseudo-header of information from the IP header and the number of bytes specified by the “Checksum Coverage”. The same pseudo-header from the IP layer used in classic UDP for inclusion in the checksum calculation is also used for UDP Lite.

UDP Lite has its own protocol number that is different than the classic UDP protocol.

6.3 QoS

This study area is related to the QOS mechanisms that may be in the upper layers. For example, an IP stack may use the IETF diffserv mechanisms to effect QOS. However, Diffserv provides the tools but does not define the policies of the QOS architecture. For example, QOS must be provided for individual user services, and packets must be marked accordingly.

At IP layer, Diffserv, RSVP or over-provisioning may be used.

In the UTRAN there are three planes involved, the User plane, the Control plane and the Management plane. Though the characteristics of the users in these planes differ (PDU size, QoS requirements, etc.), they are all sharing the same transmission and potentially interfering each other. Additionally non-UTRAN traffic will also share the transmission network. That non-UTRAN traffic can not be excluded from the IP transport network, as it could be one reason why a operator chooses IP as transport technology.

When evaluating any mechanism, one should consider its applicability for all three planes and the non-UTRAN traffic. This approach enables a unified basis for the QoS and for the efficient utilisation of transport resources.

In an IP network, the deployment of QoS features is not sufficient to ensure guarantee of service. The network shall be correctly dimensioned, so that the expected service can be provided. The provisioning of resource must be done with some over-dimensioning factor depending on the maximum packet size. The bigger the real-time packets, the more resource will be necessary.[1]

~~6.9.1~~6.3.1 Fragmentation

~~6.9.1~~6.3.1.1 General

Fragmentation is required to adjust packets to the Maximum Transmission Unit (MTU) size of the path, and, for slow links, to prevent short, time sensitive packets from being delayed by large packets in front of them on a link. For

[1] That reason is basically the same that justifies small cell size in ATM, to provide QoS.

example, with a rate of 384 kbps and a TTI of 80 ms a data payload size of 3840 bytes will result. The RLC might segment this data but all the segments (transport blocks) are multiplexed into the same packet (transport block set). Fragmentation must be performed also on the non-UTRAN traffic, if any, or the network must be oversized. The typical packet size density derivation of www traffic has its peaks at 64Byte and 1500Byte. A 1500Byte packet introduces on a E1 link the jitter of 6,25ms.

~~6.9.1~~6.3.1.2 IP fragmentation

IP fragmentation is the capability of the IP protocol to fragment a packet into multiple segments based on the Maximum Transmission Unit (MTU) size of the path the packet will traverse. The MTU of the path can be “discovered” using MTU path discovery which involves sending an ICMP message over the path and receiving the smallest MTU discovered along the path. If the packet is larger than the path MTU, it will be fragmented. The MTU is set in a router based on the link characteristics.

For PPP, the MTU size is flexible. For Ethernet links the maximum and default MTU is 1500 bytes. For Gigabit Ethernet a 9000 byte frame size possible (Jumbo Frames).

Disadvantages of IPv4 fragmentation are:

1. Bandwidth efficiency with larger packets is not realized in the part of the path with larger bandwidths since once a packet is fragmented it can only be reassembled at the endpoint.
2. For IPv4, IP header compression cannot be used. This is not the case for IPv6.
3. For IPv4, the overhead is large when IP fragmentation is used. Also, fragmentation can be performed at any link along the path. This can result in heavy processing demands on the routers in the network. IPv6 fragmentation is only allowed end to end.

End-to-end fragmentation, whether using IP fragmentation or fragmentation above the IP layer (“application level” fragmentation), can be used to adjust the packet size to the path MTU but is not suitable to solve issues around a slow link. This is because IPv6 allows the MTU to be set to a minimum of 1280 octets which is not small enough for slow link issues.

Since the disadvantages of IP fragmentation are not relevant when performed end-to-end, IP fragmentation would be supported in the UTRAN nodes to adjust the packets to the path’s MTU. It should only be done end-to-end for both IPv4 and IPv6. Also, the network should be designed such that MTU sizes are not so small that the IP headers consume too much bandwidth. This is the same approach taken for the GTP protocol and assumes that the operator has some control over the network.

IP fragmentation would not be used to facilitate delay-sensitive traffic on slow links. Layer 2 mechanisms would be used for this as indicated in the IPv6 RFC [27.]:

“IPv6 requires that every link in the internet have an MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6”.

~~6.9.1~~6.3.1.3 Fragmentation to facilitate delay sensitive traffic

In order to facilitate delay sensitive real time traffic, large packets can be segmented and the segments can be mixed with the higher priority traffic. This is only relevant for slow speed links where any delays can effect the performance of the applications.

IP fragmentation does not automatically address this problem since IP fragmentation only fragments based on the size of packet that a link can handle. This packet size may not be small enough to allow the efficient use of the link when delay sensitive traffic is present. It could be possible for IPv4 networks to set the MTU of the link to a smaller size than necessary to facilitate delay sensitive traffic. However, this can effect the efficiency of the higher speed links along the path . IP fragmentation is always end to end for IPv6.

~~6.9.1~~6.3.1.4 Application level fragmentation

Application fragmentation can help with avoiding IP fragmentation but does not automatically solve the problem for efficiency over slow links. MTU discovery can be used to determine the size of packet required to avoid IP fragmentation but it does not provide the necessary information required to know what packet sizes should be used for efficiency over slow links. It is possible that this size could be configured based on knowledge of the slow links but this affects the processing and routing efficiency efficiency over higher speed parts of the transport network

~~6.9.1~~6.3.1.5 Layer 2 fragmentation solution

In general, it’s best to take care of slow link problems only over the slow link and not over the entire path. One alternative is to handle segmentation as a lower layer issue. As an example, for PPP, the fragmentation capabilities in multilink PPP [20.] can be used for this purpose. With multiclass extensions, multiple flows can be identified within a PPP stream. The IPv6 specification says that for links that cannot convey a 1280 octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.

Layer 2 fragmentation provides flexibility because it doesn’t need to be end-to-end. It can be multi-hop using tunneling in which case it is more flexible than application level and IP fragmentation.

~~6.9.1~~6.3.2 Sequence information

If fragmentation is provided between IP and RNL, then a sequence number is required in order to reassemble the fragments.

Many of the Radio Network frame protocol specifications say that the transport layer must deliver frames in order. However, it is part of the IP UTRAN investigation to determine if this is actually a valid requirement.

If it is shown that a sequence number is required then this functionality could be provided between the frame protocols and the IP transport layer (i.e. UDP).

~~6.9.1~~6.3.3 Error detection

AAL2/ATM has the following error detection capabilities:

1. ATM provides no error detection capability for the payload, but only for the ATM header.
2. AAL2 provides error protection for the header using the HEC.

IP has the following error detection capabilities:

1. The link layer can protect the payload. Examples are the HDLC and the AAL5 checksums.
2. UDP has an optional checksum for IPv4 that is mandatory in IPv6.

Therefore, for AAL2/ATM no error checking is performed on the payload. For IP, error detection capabilities are provided at the link and transport layer. Whether additional error checking is required above the UDP layer is FFS.

~~6.9.1~~6.3.4 Flow Classification in IP Networks

Once these QoS classes have be defined and the respective priorities or requirements set, it shall be possible for UTRAN traffic to be recognised as pertaining to each of the individual classes, so that transport nodes can deliver appropriate QoS. Therefore nodes implementing Transport function are not only responsible for differentiating service among a set of IP packets but also to classify those IP packets to be able to deliver the respective QoS.[2]

Classification can basically be realised according to specific layer information, such as header field values or context information. One can distinguish between Radio Network Layer and Transport Network Layer based classification.

~~6.9.1~~6.3.4.1 Classification based on RNL information

For instance, SRNC knows about relative and absolute QoS requirements for RABs and can base its transport differentiation on RNL information based classification. It is an implementation issue only how this can be done, but it is very easy to realise thanks to additional information in layer to layer primitives.

In DRNC and Node B, such a classification can be envisaged if relevant RNL information is available. However QoS requirements as extensive as RAB parameters may not be available in those nodes.

RNL information is assumed to be unreachable in intermediate transport nodes that are UTRAN agnostic. In those nodes, classification can only be done with standard or classical IP methods.

~~6.9.1~~6.3.4.2 Classification based on TNL information

Various QoS models and solutions exist for IP networks, with specific advantages and best uses. However they have common features that they all need to realise, like flow classification. Instead of listing all QoS solutions, this section limits to information commonly used to classify IP flows to provide QoS:

- IP TOS (Type of Service) field can be used to classify among some traffic classes. This field is used in core Diffserv routers to deliver Per Hop Behaviour (so called Behaviour Aggregate Classifier).
- L3/L4 fields: IP header and Transport Protocol (UDP, TCP, and SCTP…) contain additional fields that can be used to classify among IP packets. Most commonly used fields are IP addresses, Transport Protocol ports and Protocol Identifier of IP header. Those classifiers are called Multi-Field Classifiers.
- MPLS label can also be used to distinguish among separate FEC (Forwarding Equivalence Class), even if they share a common destination.
- MPLS EXP (Experimental) bits are also proposed to be used to provide flow classification on a granularity similar and compatible with Diffserv model.

Input interface can also be used when classifying packets.

[2] Differentiation has a larger meaning than *DiffServ* acceptation. Even in *IntServ* model, IP packets are differentiated according to flow filtering, i.e. they receive different services according to established reservations.

~~6.9.1~~6.3.5 Classification Configuration

Classifications presented in 6.3.4.2 are relevant in the Transport Network Layer only. Nevertheless, they shall be defined according to UTRAN QoS requirements and to RAB classes, since those requirements are known by RNL.

Such a mapping can be done:

- At Transport bearer selection, when deciding transport bearer end point addressing that can later be used to classify the flows (e.g. IP, UDP addresses directly or mapped on MPLS label).
- At UTRAN flow source (Node B, RNC) on a packet per packet basis, by assigning the relevant TOS field, EXP field or by encapsulating in the relevant MPLS label.

Both methods offer different characteristics that are detailed hereafter.

~~6.9.1~~6.3.5.1 Transport bearer based classification

Transport Bearer based classification can be very fine but impose intermediate node to be aware of part of or all end point addressing. This is needed to create filters based on this information in intermediate nodes.

This knowledge of transport bearer addressing by intermediate transport nodes can be:

- Signalled for each individual transport bearer, but it would need non-scalable and complex signalling like RSVP.
- Pre-configured with semi static classification filters based on partial transport bearer addressing information, e.g. source UDP port, destination IP address etc. With such an alternative, intermediate transport nodes need not to be signalled at transport bearer establishment of particular filtering for the new bearer. Intermediate nodes can either be configured by O&M or by aggregate RSVP reservations.

Moreover, if the classification is based on destination information only, the source node may be unaware of classification. It does implicit classification ruled by destination node at transport bearer termination selection.

~~6.9.1~~6.3.5.2 Packet per packet classification

If QoS is marked in source node by relevant tagging in IP or MPLS headers, filtering in intermediate node is simpler. The classification in intermediate transport nodes does not depend on end node transport addresses and therefore is simpler to configure and manage.

On the other hand, the granularity may be coarser if only TOS or EXP bits are available to distinguish between traffic classes.

~~6.9.1~~6.3.6 UTRAN Hop-by-Hop QoS Approach

This approach relies on the QoS differentiation, which is provided by the IP backbone. This means the UTRAN internal flows (e.g. RAB traffic, NBAP signalling, ...) have to be mapped to the IP network. This mapping is not obvious because of the specific properties of UTRAN traffic. Due to the fact that the RLC/MAC layer are on RNC side, even the best effort RAB QoS class becomes time constraint traffic in UTRAN, but with more relaxed delay requirements than the conversational RAB QoS class. The delay requirements themselves are dependent from the MAC strategy in the RNC, which is manufacturer dependent.

QoS differentiation in the IP backbone could be provided by Diffserv for example. Scheduler algorithms and strategies from the installed routers are used and must be configured to meet the UTRAN requirements.

The last mile between the edge router and a NodeB is assumed to be a bottleneck for all UTRAN traffic flows. The adaptation to the low speed link has to be done by L2 techniques. Advanced functions like QoS differentiation, segmentation and multiplexing are needed in L2. For example, the PPP protocol is a meaningful candidate for this adaptation. It provides with its extensions Multi-Class PPP and PPPmux the required QoS differentiation, segmentation and multiplexing functionality.

However, still some issues need to be solved:

- A mechanism shall be defined to inform the edge router about the needed quality classes towards the NodeB and the parameters used for the differentiation.
- It shall be defined on which edge router functionality the standard design relies on, and what can remain implementation dependent.
- The interworking of PPPmux with MC-PPP should be defined, for instance the availability of a separate PPPmux instance per QoS class shall be clarified.

~~6.9.1~~6.3.7 UTRAN End-to-End QoS Approach

The end-to-end approach provides QoS differentiation for the UTRAN traffic flows inside the UTRAN NEs. User plane protocol proposals like CIP and LIPE rely on this principle. But also the PPPmux based proposal can provide an e2e approach by tunnelling the PPP protocol via L2TP (TCRTP). The queuing and scheduling is performed inside the NEs under control of the UTRAN equipment manufacturer. In the IP backbone only one QoS class is needed for UTRAN traffic, which could be the expedited forwarding (EF) class of Diffserv.

The QoS differentiation is simpler because the quality classes are well known inside the NEs and the complex management function to distribute the QoS parameter in the IP network can be avoided.

However, for the implementation dependent O&M traffic the head of line blocking problem still exists. In case the edge router provides on data link layer only one QoS class, IP fragmentation at the O&M center could be configured to a reasonable IP packet size. If the edge router provides at least two QoS classes (ML-PPP) the best effort O&M traffic could easily be distinguished from the tunnel carrying the other UTRAN traffic.

6.4 Transport network bandwidth utilisation

This study area is related to bandwidth efficiency by e.g. multiplexing/header compression, resource management, and the use of segmentation. Lower speed links, such as E1, or shared higher speed links may require different techniques (e.g. header compression and multiplexing) than dedicated higher speed links.

When evaluating and comparing efficiency of different candidate schemes for efficient bandwidth utilisation, their impacts on the other study areas of this chapter have to be identified and considered.

~~6.9.1~~6.4.1 General issues

~~6.9.1~~6.4.1.1 Multiplexing

Multiplexing provides a means for reducing the impact of the size of the UDP/IP headers in a packet. It is important for gaining better bandwidth efficiency with small packets. Multiplexing can be performed at the application layer or a lower layer. An example of application level multiplexing would be if the length field in the GTP header would be used to delimit GTP tunnels multiplexed within one UDP/IP packet. This is not currently supported in GTP. Application level multiplexing reduces the impact of the IP and UDP headers. However, when header compression is applied, the overhead is already significantly reduced.

Multiplexing within a PPP frame is being addressed currently in the IETF [10.]. Advantages of PPP multiplexing are:

1. Layer 2 multiplexing provides the possibility for routing multiplexed packets using tunneling as does application level multiplexing.
2. Layer 2 multiplexing is not end-to-end so how multiplexing is applied at the source does not need to be based on the worst case link in the path.
3. Packets with different IP addresses can be multiplexed in same PPPmux frame. With application level multiplexing, only packets going to same IP address can be multiplexed.

~~6.9.1~~6.4.1.1.1 Location of multiplexing in transport network

Three architectures are proposed for multiplexing distribution in transport network, as depicted in

Figure 12. They are presented and discussed hereafter.

~~6.2.4.5.1.2~~6.4.1.1.1.1 Scenario 1:

Multiplexing is done end-to-end, i.e. transparently to intermediate transport nodes. This solution has the benefit of simplicity regarding intermediate transport nodes that may be multiplexing agnostic.

Some limitations can be noted for this scenario:

- All information multiplexed in one packet shall follow the same path and shall be serviced with the same QoS, since intermediate transport nodes are multiplexing agnostic.

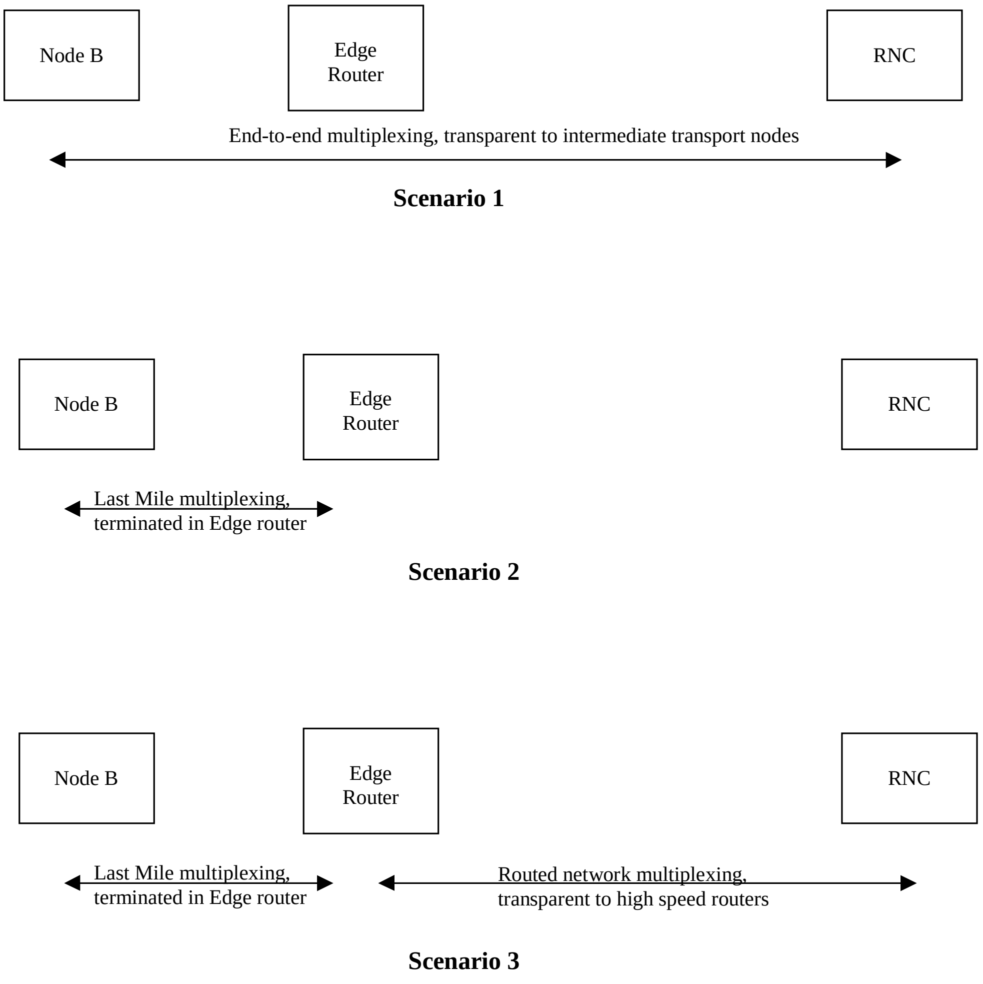
  However it is still possible to handle differentiation in end nodes and to take benefits of several QoS in the transport network: there is only the restriction that all information in one packet cannot be serviced differently, once they have been multiplexed.

  As far as the routing/ path is concerned and considering current RNL architecture, Node B has only one Iub interface towards one C-RNC and therefore it is not a requirement to allow multiplexing of information having different destinations.

- Both aspects of multiplexing as introduced above in 6.4.1.1 cannot be distinguished. Therefore they cannot be optimised separately.

  Nevertheless, since low speed link multiplexing is the most important aspect, it can be the basis for optimisation.

As a conclusion, scenario 1 has some limitations but it can provide simple transport network solutions, since it needs only basic functionality in transport network intermediate nodes


Node B
Edge Router
RNC
End-to-end multiplexing, transparent to intermediate transport nodes
Node B
Edge Router
RNC
Last Mile multiplexing, terminated in Edge router
Node B
Edge Router
RNC
Last Mile multiplexing, terminated in Edge router
Routed network multiplexing, transparent to high speed routers

Scenario 1

Scenario 2

Scenario 3

Figure 12: Scenarios for multiplexing location.

~~6.2.4.5.1.2~~6.4.1.1.1.2 Scenario 2:

Multiplexing is on last mile low speed links only, where bandwidth is a limiting factor and where high-speed interface resource optimisation is not required. It provides functionality on the exact network portions that require efficiency.

Hereafter are the characteristics of this solution:

- This scenario induces some functionality in edge router to terminate the multiplexing.
- Downlink packets arrive in the edge router and shall be multiplexed and differentiated according to some knowledge of QoS. Therefore the edge router shall participate in QoS differentiation and end-to-end differentiation is not sufficient.
- Packets multiplexed together on the uplink/ downlink can be forwarded to/ from different paths with different QoS after the edge router. This brings flexibility, with some complexity in the transport network.

Therefore scenario 2 is more flexible and optimal, with more complex QoS handling in transport network and higher processing power per packet in the edge router. It does not cover the multiplexing on high-speed interfaces for reduction of number of packets per second.

~~6.2.4.5.1.2~~6.4.1.1.1.3 Scenario 3:

Scenario 3 can be considered as an extension of scenario 2 for high speed link multiplexing.

There are indeed two multiplexing "sessions", one between Node B and edge router and another between edge router and RNC. The first one is very similar to the one described in scenario 2. The second one is presumably routed with less stringent bandwidth requirement.

It can be expected that sufficient concentration exist between edge router and RNC to allow several sessions towards several RNC. Therefore the edge router is really doing routing of individual information payloads of both types of multiplexing sessions: it de-multiplexes on one side what it receives and re-multiplexes on the output interface.

~~6.9.1~~6.4.1.2 Resource Management

The solution for resource management should be scalable in complexity. It should also allow traffic other than UMTS traffic without seriously degrade the quality of service of the UMTS traffic. Some operators will require IP connectivity for other applications using the same network as the UTRAN. The use of VPNs can be investigated in order to facilitate the sharing of network resources. Resource management setup time should be minimized such that it meets the requirements but does not add too much delay for the application connection setup.

For the low-speed links, delay needs to be well controlled for soft handover and other time critical operations. Also, since these interfaces are part of the network where resources are more expensive, it's particularly important to utilize the bandwidth in an efficient way. In addition, where node synchronization messages are used, they must have small delay in order to be effective. For these reasons the use of on-demand resource allocation should be given particular consideration.

Static routing or dynamic routing using a routing protocol could be used. Static routing allows easier control over delays but puts heavier requirements on configuring the network. Dynamic routing protocols add complexity but increase the possibilities for automatic configuration.

The following possible functions relating to resource management should be considered.

- Admission control: Enforces a limited load within a traffic class in order to limit the delay caused by buffering in network routers.
- Policing: Once traffic has been admitted in a network based on certain traffic characteristics, it may be policed to ensure that it does not violate the conditions of its admission.
- Reservation of resources: How should resources be reserved in the transport network?

Allocation of resources can be static or dynamic. It can also be performed by one or a combination of several methods, for example:

- Over-provisioning: This method is static and there is no need for admission control. However, it does not take advantage of transport bandwidth efficiency gains that IP can provide.
- Allocation of aggregates of flows (a trunk). This can be dynamic but changes of bandwidth allocation are made more slowly than per flow allocation.
- Allocation per flow: Allocation of resources is made on a per call basis.
- The admission control function can be centralized or distributed:
- With server based admission control, resource requests are made to a server. A centralized or partly distributed server architecture can be used.
- Distributed admission control uses signalling (e.g. RSVP). The admission control function is distributed in the routers and is performed hop-by-hop. RSVP could have scalability problems for large networks if it is used per flow.

~~6.9.1~~6.4.2 Solution Comparison data

Preliminary simulation results for LIPE and PPPMux indicate that in general, comparison of capacity performance of the different multiplexing protocols alone is inconclusive. Other criteria must be used in order to select one protocol over another.

6.5 User plane transport signalling

The use of IP based protocols for the user plane mandates compatible signalling in the control plane. The signalling must accommodate the appropriate mechanisms to specify, establish, and manage IP streams as opposed to virtual circuits/connections. Signalling for IP bearer exchanges transport bearer identifiers, (e.g. IP addresses and UDP port numbers) for each end of the bearer stream. If there is a need for user plane connections, it should be investigated how connections between UMTS nodes should be handled. It should be investigated whether an ALCAP protocol is required.

~~6.9.1~~6.5.1 Solution without ALCAP

Unlike Iu-cs, Iu-ps does not require an TNL signalling protocol to establish/maintain/release user plane Transport Bearers.

The transport bearer termination points, at CN and UTRAN sides, are identified by Information Elements carried by RANAP messages [3.]:

- Transport Layer Address IE: This information element is an IP address to be used for the user plane transport. It generally corresponds to the IP address of the board that processes GTP-u for the RAB to be established.
- Iu Transport Association IE: This information element is the GTP Tunnel Endpoint Identifier.

These fields are coded as bit strings or octet strings. They are transparent to RANAP i.e. to Radio Network Layer (RNL), and are only seen by the Transport Network Layer (TNL).

The reason for not using ALCAP in the PS domain is linked to the connectionless aspect of IP layer.
ALCAP protocol is needed for the case there is a TNL switch between two RNL nodes, since RNL protocol (RANAP on Iu, RNSAP on Iur, NBAP on Iub) does not terminate in the TNL switch (e.g. AAL2 switch). This is shown in Figure 13.
In the case of IP networks, destination IP address is sufficient to route an IP packet to the TNL termination point.

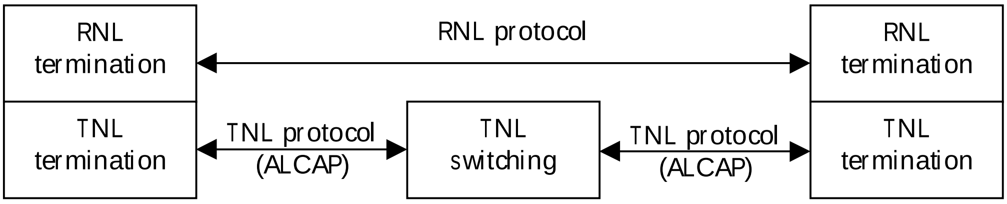


Figure 13: RNL and TNL terminations

When IP is used as transport in the UTRAN, it is therefore possible to avoid the use of a TNL protocol (i.e. ALCAP) on Iur and Iub while keeping the independence between RNL and TNL. Avoiding the use of a TNL protocol results in benefits with regards to e.g. connection set-up delays.

Similarly to Iu-ps, it is proposed to exchange Transport Bearer termination point identifiers via the RNL signalling protocols over Iur and Iub (i.e. via RNSAP and NBAP).
Transport Bearer termination points can always be defined by:

- The IP address of the termination point
- The transport bearer identifier within this IP address
- Transport Bearer Characteristics.

The first two items correspond respectively to Transport Layer Address IE, Iu(x) Transport Association IE used in RANAP messages. The last item is added to carry information which is specific to the Transport Bearer and which is not interpreted by the Radio Network layer.
The contents of those fields should be coded as bit strings or octet strings in order to comply with the RNL/TNL independence: these fields are transferred to the TNL without being interpreted by the RNL.

A simple solution consists of introducing two IEs in appropriate RNSAP and NBAP messages to identify the user plane transport bearer termination points:

- Transport Layer Address IE: This information element is an IP address to be used for the user plane transport.
- Iur/Iub Transport Association IE: This information element is the identifier of the Transport Bearer at the IP address termination point.
- Transport Bearer Characteristics IE: This information element contains information specific to the Transport Bearer.

These IEs shall be transferred transparently by the RNL to the TNL.

Related RNSAP messages are e.g. RL Setup Request, RL Setup Response, RL Addition Setup, RL Addition Response.
Related NBAP messages are e.g. RL Setup Request, RL Setup Response, RL Addition Setup, RL Addition Response, Common Transport Channel Setup Request, Common Transport Channel Setup Response.

Note: Special attention shall be given to the fact that any unnecessary parameter dependence on the TNL type shall be avoided.

~~6.9.1~~6.5.2 LIPE solution

When LIPE is being used for Iub/Iur User Plane traffic, there are two alternatives for user plane transport signaling. Alternative I requires no changes in the existing RNSAP and NBAP procedures but a lightweight ALCAP-like procedure is required. Alternative II introduces a new information element to Radio Link Setup Messages in RNSAP and NBAP but ALCAP is not required.

~~6.9.1~~6.5.2.1.1 Alternative I Solution:

There are two steps involved in creating a communication channel between two LIPE peers. The first step is to set up a LIPE tunnel. Once a tunnel has been set up, connections for different streams may be multiplexed into this tunnel. Typical scenarios for a LIPE tunnel are illustrated in Figure 14. In the case of point to point link, we assume that IP layer connectivity has been established using mechanisms such as PPP, ATM-AAL5 etc.

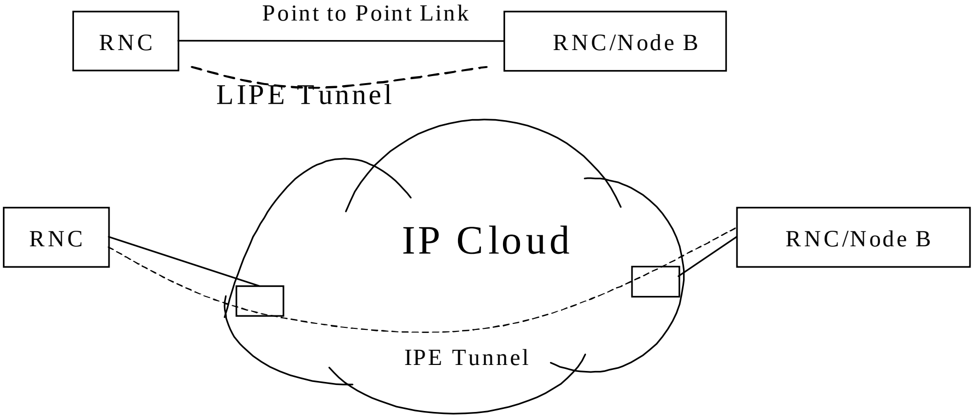


Figure 14: Typical LIPE tunnels in a 3GPP network.

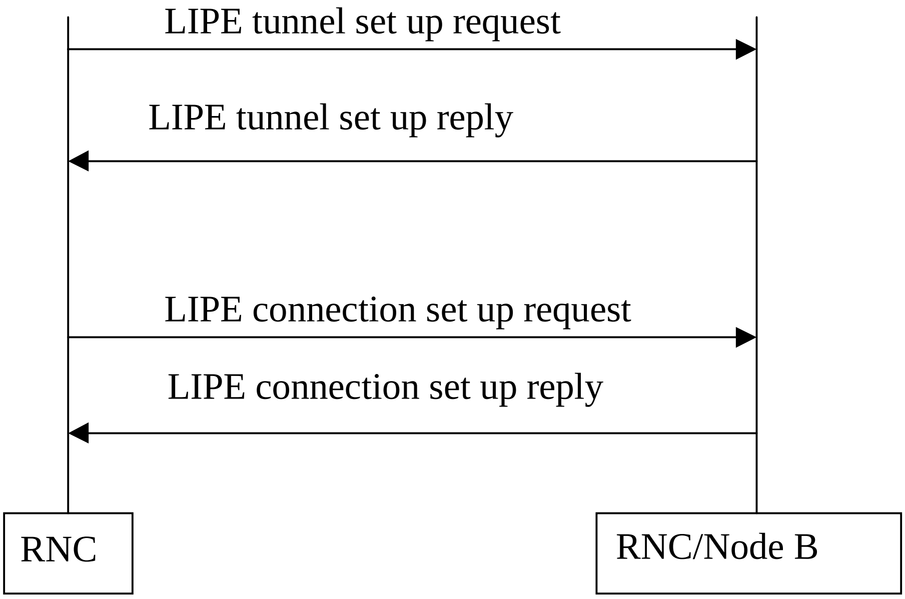

LIPE tunnel set up request
LIPE tunnel set up reply
LIPE connection set up request
LIPE connection set up reply
RNC
RNC/Node B

Figure 15: Tunnel/Connection set up procedure.

~~6.9.1~~6.5.2.1.2 LIPE Signaling Channel

A specified UDP destination port is used for the exchange of LIPE signaling messages The format of the LIPE signaling message is given in Figure 16.

IP (20)	UDP (8)	TYPE (4)	LENGTH (4)	Control Message Payload (20)

Figure 16: LIPE Signalling Channel Message format

~~6.9.1~~6.5.2.1.3 Tunnel Setup Procedure

The actual format of the tunnel setup control message payload is shown in [22.].
The tunnel set up request message payload should consist of the following

1) UDP destination port number for the LIPE tunnel for the reverse LIPE tunnel.

Protocols such as RSVP may be used for reservation of bandwidth resources across the path between LIPE peers for QoS guarantees. This issue is not addressed in this contribution.

A successful tunnel set up reply message should consist of

1) UDP destination port number at the destination node for the forward LIPE tunnel.

A tunnel setup failure condition is triggered by a tunnel set up reply message or time out. Retransmissions of LIPE tunnel set up messages for failed tunnel set up instances should be supported.

~~6.9.1~~6.5.2.1.4 Connection Set up Procedure

Once the tunnel set up procedure has been completed, connections for several RAB's can be set up on the tunnel. A control message type is defined for connection setup request. The actual format of the connection setup request control message payload is shown in [22.]. Connection request for a LIPE connection for a RAB carries:

1) RABID
2) Flow ID (FID)

A control message type is defined for connection setup reply. The actual format of the connection setup reply control message payload is as shown in [22.]. A successful connection set up reply message carries

1) Error Code
2) RABID
3) FID for the reverse path.

A connection setup failure condition is triggered by a connection set up reply message or time out. Retransmission of LIPE connection set up messages for failed connection set up instances should be supported.

~~6.9.1~~6.5.2.1.5 Tunnel tear down

A control message type must be defined for tunnel tear down. The actual format of the tunnel tear down control message payload is as shown in [22.]. Tunnel tear down may be initiated by either peer. The tunnel tear down message should contain.

1) UDP destination port for the forward tunnel (w.r.t to the peer initiating tunnel tear down).

A tunnel should not be torn down without tearing down all connections through the tunnel.

~~6.9.1~~6.5.2.1.6 Connection tear down

A control message type must be defined for connection tear down. Connection tear down request should carry.

1) FID

~~6.9.1~~6.5.2.2 Alternative II Solution:

For the Iur interface, the procedures setting up transport bearers should be modified to include an information element for conveying the flow identifier information in the Request message. Correspondingly, the DRNC should return a flow identifier information for the reverse direction in the Response message.

Similarly, for the Iub interface, the NBAP, the procedures setting up transport bearers should be modified to include an information element for conveying the flow identifier information in the Request message. Correspondingly, the Node B should return a flow identifier information for the reverse direction in the Response message.

When Alternative II solution is being used to establish flow identifiers, ALCAP is not required.

6.6 Layer 1 and layer 2 independence

This study area is related to the capability to allow multiple layer 1 and layer 2 technologies.

The role of Layer 2 and Layer 1 in the QoS and/or in the transport resource efficiency needs to be considered when specifying the requirements towards L2/L1.

Requirements on L2/L1 (e.g. in sequence delivery) should be documented in the UTRAN specifications to ensure that appropriate technologies can be more easily selected.

~~6.9.1~~6.6.1 Options for L2 specification

~~6.9.1~~6.6.1.1 General

The used L2 techniques may vary across the different interfaces and links. Especially, if slow links are used at Iub interfaces, specific features from the L2 protocol are required. Besides the multiplexing functionality, ML/MC-PPP [20.], [21.] may be required for QoS differentiation. It provides several queues, segmentation and scheduling functionality. Header compression is an other important feature which may be required to improve the efficiency.

A common case in the IP transport architecture is that the UTRAN NEs are connected to an IP router which is then responsible for the L2 termination. Supported L2 techniques have to be negotiated with the IP network provider to build an efficient TNL. It is then up to the operator what layer 2 protocols are used in the transport network.

However, also the use of point-to-point links between UTRAN NEs is a reasonable scenario. Here, no intermediate router will terminate the L2, both NEs have to implement the same L2 protocol. In a multi-vendor scenario this case may cause problems.

~~6.9.1~~6.6.1.2 L2 not standardised

Not standardising any L2 will provide the most freedom for the operators to build their transport network. A variant of this approach could be to standardise some requirements for the selection of L2 to ensure that the expected functions for UTRAN TNL are provided. However, because the usage of these functions in L2 is essential to provide an efficient TNL service, they will be implemented anyway even if not required in the standard. The only issue which remains here is the multi-vendor scenario.

~~6.9.1~~6.6.1.3 L2 standardised

Fully standardising one L2 to the exclusion of allowing others would solve the multi-vendor issue for point-to-point links. But, standardising one exclusive L2 protocol that must be used in the UTRAN NEs would restrict the flexibility for the operators. A solution which solves the multi-vendor issue, but still offers the full flexibility would be the preferred approach for the L2 standardisation for IP transport in UTRAN.

Requiring the implementation of one or a limited set of L2 protocols, but still allow the use of any L2 protocol in the UTRAN NEs would be a good solution for the standard.

The L2 protocol specified in the standard to be implemented in the UTRAN NEs should be the PPP protocol [11.] with its extensions PPPmux [10.] and ML/MC-PPP [20.], [21.] and header compression. During the work in RAN3 for IP transport it has been shown that the PPPmux approach fulfils the requirements and provides good performance.

The layer 2 framing protocol below ppp is FFS.

6.7 Radio Network Signalling bearer

This study area is related to the transport of Radio Network Signalling over an IP network.

~~6.9.1~~6.7.1 Iub RNL signalling bearer

~~6.9.1~~6.7.1.1 SCTP characteristics

SCTP/IP [24.] can provide the following:

- Acknowledged error-free non-duplicated transfer of user data.
- Data fragmentation to conform to discovered path MTU size.
- Sequenced delivery of user messages within multiple streams, with an option for order-of-arrival delivery of individual user messages.
- Optional bundling of multiple user messages into a single SCTP packet.
- Network-level fault tolerance through supporting of multi-homing at either or both ends of an association.
- Congestion avoidance behaviour.
- Resistance to flooding and masquerade attacks.

~~6.9.1~~6.7.1.2 Proposal 1

In an IP network, transport protocols like TCP or UDP are used to transport messages. UDP is unreliable. TCP has weaknesses regarding signalling transport e.g. it is a byte-oriented protocol instead of a message-oriented protocol (see [24.]). SCTP, the new protocol that is being developed in IETF for the purpose of signalling transport in an IP network, is a suitable alternative. Furthermore, SCTP has already been introduced on Iur and Iu-PS interfaces in R99 specifications. (See [4.] and [6.] Therefore, it is proposed to adopt SCTP on Iub as well.
The proposed protocol stack in RNC and Node-B for the IP option is as follows:

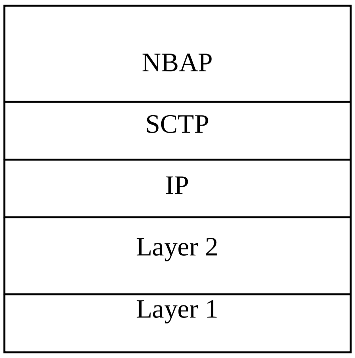


Figure 17: Iub Signalling bearer protocol stack without Adaptation Layer

~~6.9.1~~6.7.1.3 Proposal 2

For an SCTP-based solution for the Iub signalling bearer, an SCTP adaptation module would be used between NBAP and the SCTP protocol.

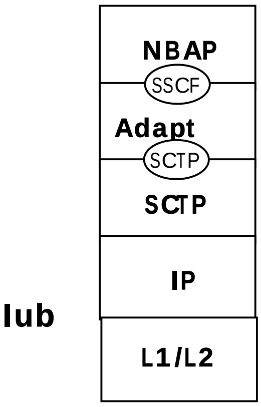


Figure 18: Iub Signalling bearer protocol stack with Adaptation Layer

~~6.9.1~~6.7.1.4 Use of SCTP

A SCTP connection between two endpoints is called an association. One SCTP association can be considered as a logical aggregation of streams. A stream is a unidirectional logical channel between 2 endpoints. In order to achieve bi-directional communications, two streams are necessary, one in each direction. Each user message (i.e. a message originated from the SCTP user application) handled by SCTP has to specify the stream it is attached to, a stream identifier allows to identify each stream inside the association. Therefore, each SCTP stream can be considered as an independent flow of user messages from one SCTP node to another. The stream independence has the advantage of avoiding blocking between streams.

Between CRNC and Node B, one or several SCTP associations might exist. Node-B selects a SCTP association at creation of an UE context. It would not be very efficient to consider each association as a signalling bearer because all requirements of NBAP signalling transport can be fulfilled by one SCTP stream. Since it can be considered one SCTP association is an aggregation of NBAP signalling bearers, it is proposed that each NBAP signalling bearer be mapped on a pair of SCTP streams (one in downlink and one in uplink). The choice of stream identifiers being done by the user application, the simplest solution is to choose the same stream identifier for the two streams.

Although two streams per association (one in each direction) is enough for the transfer of NBAP messages, this proposition adds more flexibility as it allows each association to support several flows of NBAP messages and it has the advantage to avoid blocking between signalling bearers.

[7.] describes the Node-B logical model as it is seen from the CRNC. It defines one Node B Control Port and Communication Control Ports within each Node-B. A communication control port corresponds to one signalling bearer and each signalling bearer between Node-B and CRNC can at most correspond to one communication control port. At creation of an UE context, Node-B selects a communication control port whose identity is communicated to CRNC. According to the previous discussion, each communication control port will correspond to one SCTP association and two SCTP streams in opposite directions of the same association. And similarly for the Node-B control port.
It is expected NBAP specifications will not be impacted by this change. The IE “Communication Control Port Id” still identifies the signalling bearer i.e. one SCTP stream number inside one SCTP association between the Node-B and the controlling RNC.

~~6.9.1~~6.7.2 RNSAP Signalling

The SUA delivery mechanism provides the following functionality:

- Support for transfer of SS7 SCCP-User Part messages (e.g., RNSAP).
- Support for SCCP connectionless service.
- Support for SCCP connection oriented service.
- Support for the seamless operation of SCCP-User protocol peers.
- Support for the management of SCTP transport associations between a SG and one or more IP-based signalling nodes).
- Support for distributed IP-based signalling nodes.
- Support for the asynchronous reporting of status changes to management.

Given these capabilities, SCCP (and the associated adaptation protocol, M3UA) may be unnecessary and it should be considered that they may be eliminated in order to provide a simpler and more efficient signalling transport that may be carried via SUA/SCTP/IP over ATM AAL5 or other Layer 2 protocols, such as HDLC-PPP, etc.

~~6.9.1~~6.7.3 RANAP Signalling

In order to minimise the changes on UTRAN Radio Network Layer and thus to reduce the number of different variants of any application signalling protocol, the SCTP shall be used together with the suitable Adaptation Module. This is according to the signalling transport framework architecture of the SigTran Working Group of IETF, RFC 2719 [24.]. The following figure illustrates the application of Adaptation Module in the Transport Network Layer of Iu interface.

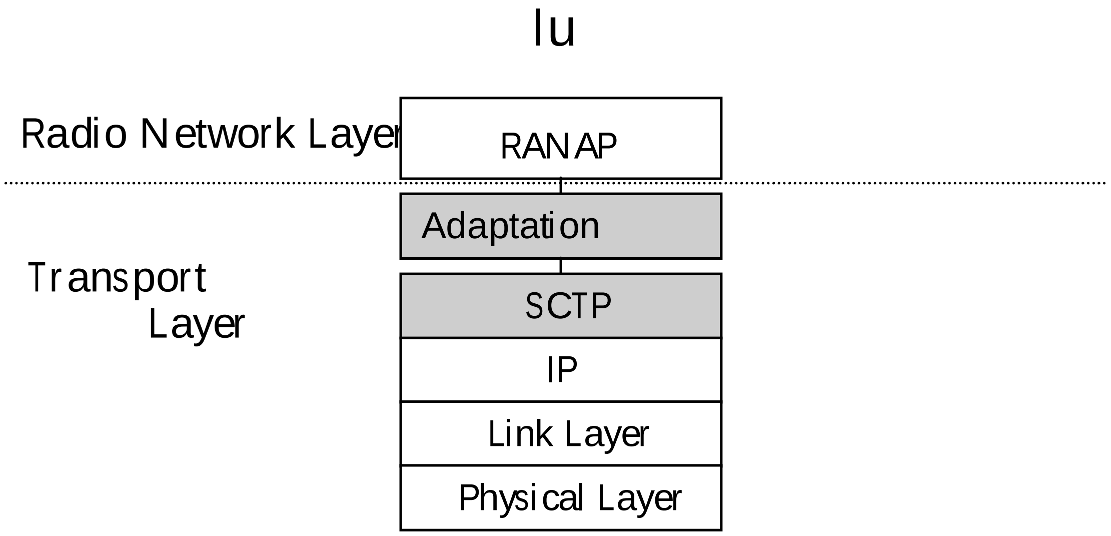


Figure 19: RNL Signalling bearers on Iu interface, the principle.

6.8 Addressing

This study area is related to all addressing issues with regards to the introduction of IP Transport. For example, the advantages of using IPv6 should be investigated. Also, addressing issues relating to inter-working with AAL2/ATM nodes should be considered.

IPv6 has a 16 byte address field compared to 4 byte address field for IPv4. It is well known that the IPv4 public address space is running out, especially outside the U.S.

~~6.9.1~~6.8.1 General addressing requirements

- IP addressing in UTRAN shall be logical and should not have any dependency on network element or interface type.
- In case of Ipv4, to ensure efficient usage of IPv4 addresses and routing efficiency, IP based RAN shall adopt classless IP addressing scheme, using Variable Length Subnet Masks (VLSM).
- IP addressing in UTRAN scheme must support hierarchical routing network design and work well with the chosen routing protocol to provide best route convergence time in order to avoid network instability.
- Where applicable, IP addressing in UTRAN must budget for multi-homing of network elements.
- IP addressing in UTRAN must be scalable and take network element/interface growth and network expansion into consideration.
- RAN IP Addressing scheme must be flexible and be suitable for different RAN sizes and topologies.
- IP addressing in UTRAN must allocate addresses efficiently.

In an IP based UTRAN it is necessary that every UTRAN Node gets at least one IP address. Even in an UTRAN with ATM transport UTRAN Nodes will require IP addresses, e.g. for O&M functions. In fact there will be the situation that the most UTRAN nodes will have several IP addresses. Because of this reasons it is necessary to ensure that sufficient IP addresses are available. Especially when an operator decides to use public IP addresses for some UTRAN nodes, the availability of sufficient number of IP addresses must be studied with respect to the bearer addressing scheme.

If there is a private, isolated UTRAN network, then its possible that the IPv4 address space would be sufficient. However, if the UTRAN traffic is routed through a public network or a broader private network, then the IPv4 address space may not be sufficient. Using private addresses may require the use of a Network Address Translation (NAT) function when the UTRAN traffic must traverse a network using public addresses in order to translate public addresses to private when entering the private network. Private IPv4 addressing is a commonly used solution for extending the IPv4 address space.

However, the use of NATs causes problems in the network. Some of these are:

1. It breaks the End-to-End Paradigm for Security when using IPSec.

UTRAN protocols use external signalling to exchange transport address and connection identifier information. An Application Level Gateway might be needed to take care of ensuring that the correct addresses are used for a session. When intermediate Application Level Gateways are used the performance is hurt and the delay is increased.

It adds costly manipulation on all packets.

It is a single Point of Failure.

It increases management and system configuration complexity.

~~6.9.1~~6.8.2 Bearer addressing solutions

~~6.9.1~~6.8.2.1 Destination IP addresses and destination UDP ports as connection identifiers

Destination IP addresses and destination UDP ports are used for connection identification based on the following assumptions:

- UDP ports provide approximately 65,000 connection identifiers. It is acceptable to require the addition of an IP address to support additional 65,000 connections. Adding IP addresses is not a concern, particularly if IPv6 is used in IP UTRAN networks.
- Using dynamic UDP ports means that a large range of UDP ports must be allowed through a firewall for the radio network application IP host. This can compromise the internal network if the host also supports other applications that use dynamic UDP ports.
- The use of VPNs can be used to isolate the UDP ports used as connection identifiers from a firewall and can remove the need for a firewall in some cases.
- Network Address Translators (NATs) can also cause problems when dynamic UDP ports are used since they change the address and possibly the UDP ports of packets. Only IPv6 could be used in the IP UTRAN network so that NATs can be avoided or VPNs should be used such that NATs will not effect the IP address and UDP port used for the application.

6.9 IP transport and routing architecture aspects

6.9.1 Flexibility of IP architectures

Wide deployment and cost effectiveness of IP infrastructure are major reasons for introducing IP as a transport option in UTRAN. Therefore the chosen architecture must take best benefit of IP technologies and infrastructure.

Infrastructure transporting IP packets encompass a large variety of equipment like routers and switches, implementing a wide range of functions (routing, switching, route discovery, tunnelling, load sharing, QoS handling etc). The flexibility that can be used to combine those equipment and functions are a major advantage of IP.

It implies that several different architectures can be built with IP, which can adapt to various topologies and link layer technologies. This flexibility brings both adaptability and competitiveness.

That flexibility has to be considered, when defining higher layers for IP transport. No optimisation should be made according to a limited set of topologies or link layer technologies that could later restrict the competitive advantage of IP.

~~6.9.1~~6.9.2 Hosts and routers

Basically, the IP Transport Network is a set of nodes and links connecting Network Elements implementing UTRAN functions (Node B, RNC, and Management Platform). That network is responsible for transporting user, control plane, data and O&M data between the Network Elements implementing UTRAN functions with some requirements (addressing, security, Quality of Service…).

Several networks can fulfil these requirements. It relies on vendors, operators and third party service providers to determine best implementations for the transport network.

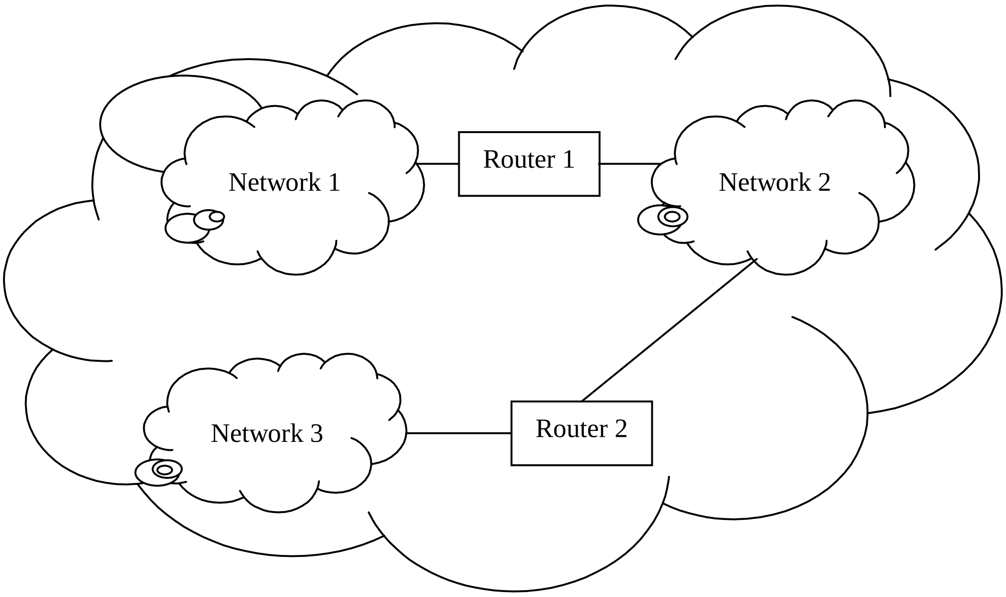
In an IP Transport Network, one can distinguish between end nodes (hosts) and intermediate nodes responsible for forwarding IP packets.

Since standardisation of IP transport option is intended to be layer 2 independent, in this study area, IP Transport architecture is limited to nodes implementing an IP layer.

Nodes implementing an IP layer are either hosts, or routers. According to [8.], the forwarding capability is the only feature distinguishing routers from hosts.

IP Hosting is a necessary function for a network element supporting of the UTRAN functions (Node B, RNC) but these network elements may also include transport network functions. Like AAL2 switching for ATM transport, IP forwarding and routing is not part of UTRAN functions. Routers connect networks of IP hosts to build internets. Hosts are not allowed to route packets they did not originate.

Figure 20: Routers interconnecting IP networks.



Routers forwarding IP packets in the transport network may have the following characteristics:

□ They can process user plane and control plane data at any layer lower or equal to IP.

□ They may process higher layer information for Transport Network O&M or configuration purpose.

Other IP features may encompass tunnelling mechanisms (e.g. GRE, MPLS, L2TP, IPSec) or mechanisms requiring storage of state information for every flow (e.g. RSVP). Such features, if too much specific or complex, should not be required to be standard function of the transport network.

In IP architecture, a host sees only routers directly accessible (without intermediate router). In most cases (no multi-homing), there is only one such router, named First Router in the Architecture. A node acting as a router may be a First Router for other Node Bs.
If the First Router is part of the IP network of routers, it is typically named Edge Router.

In the special case when two UTRAN NEs are directly connected with a point-to-point link, taking no benefit of IP infrastructure, no intermediate router exists between both UTRAN NEs. However there are still benefits for IP (e.g. no QAAL2). This case constitutes one very specific topology solution.

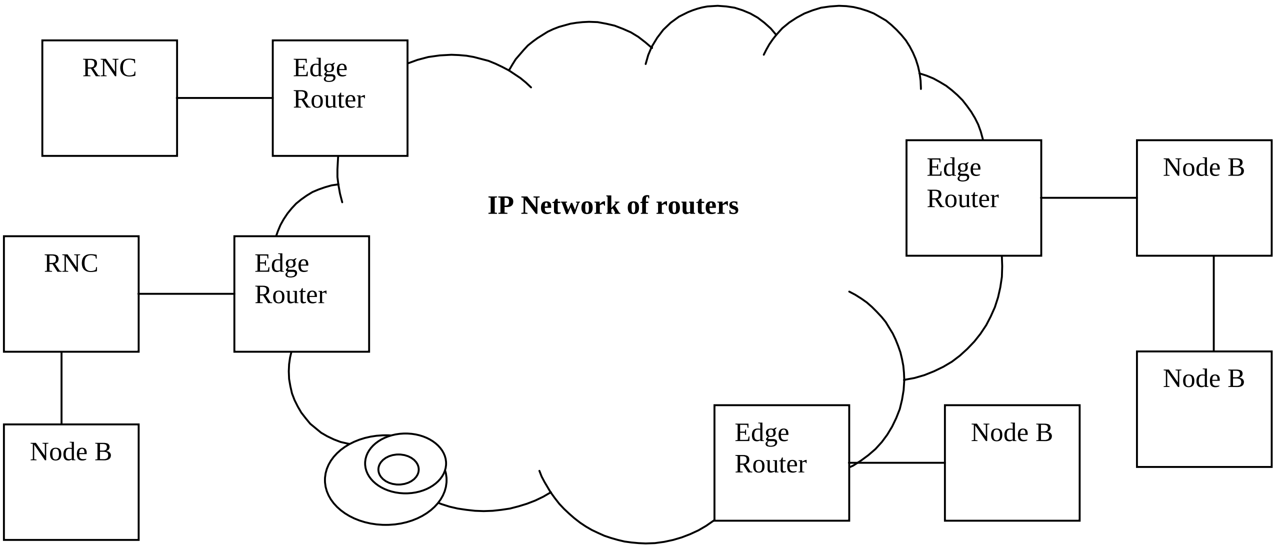


Figure 21: Example Architecture for IP Transport Network

The physical medium between one Node B and the first router is expected to be often bandwidth limited.

~~6.9.1~~6.9.3 IPv6 aspects

The UTRAN can be a very large network, with potentially thousands of end system hosts connected to a large routed network. If public IPv4 addresses are used in this network to begin with, the work is substantial to later reconfigure this network to IPv6, when the IPv4 address space is running out, or when the operator desires to move to using the IPv6 protocol in all of his networks.
If the network is a newly built closed intranet in the first release, it is quite easy to use IPv6 from the start, since interworking with IPv4 nodes will not be needed in that case.

~~6.9.1~~6.9.3.1 Improved Performance

There is potential for improved performance when IPv6 is used. This is due to the following:

1. There are fewer header fields and optional headers compared to IPv4 (from 12 to 8) and the checksum in the IP header has been removed.
2. IPv6 header fields are better aligned. This also facilitates implementation in hardware.
3. Header compression can reduce the header size better than IPv4 under certain conditions.

Network performance is improved due to the hierarchical address architecture.

~~6.9.1~~6.9.3.2 Autoconfiguration

Address Management is provided using Auto-configuration. This provides the following benefits:

1. Lower administrative cost
2. Easier renumbering
3. Easier Address Management

There are two address management schemes defined:

1. Stateful autoconfiguration using DHCPv6. This is also used with IPv4. Hosts obtain interface addresses and/or configuration information and parameters from a server.
2. Stateless autoconfiguration: Stateless autoconfiguration requires no manual configuration of host and no configuration of servers.

Stateless and stateful autoconfiguration can complement each other. The stateless approach is suitable in the case where the exact addresses a host use is not a great concern. The stateful approach is suitable when tighter control over exact address assignments is required.

~~6.9.1~~6.9.3.3 IPv6 to IPv4 interworking

A wide range of techniques have been identified and implemented for IPv6/IPv4 interworking. They basically fall into three categories: tunneling techniques, translation techniques, and dual stack techniques.

- Tunnels can be used for routing packets between two IPv6 hosts via an IPv4 network by adding an IPv4 header to the IPv6 packet.
- Translators are used for IPv6 to IPv4 interworking by translating the headers.
- Dual stack techniques mean that IPv4 and IPv6 co-exist in the same host.

It is likely that if an operator starts with an IPv4 UTRAN they will not change to IPv6 all at once by upgrading all IP UTRAN nodes to IPv6 at the same time. New nodes that are IPv6 capable will be added as the network grows. These IPv6 nodes must then interwork with the existing IPv4 UTRAN nodes and utilize the IPv4/IPv6 interworking techniques developed by the IETF. Particularly on the Iur, where full connectivity is required, interworking between IPv4 nodes and IPv6 nodes could require many more IPv4 addresses than the operator has left available.

Interworking techniques have disadvantages such that it is best to avoid using them if it is possible. Summaries of the main interworking techniques are provided in the following sections.

~~6.9.1~~6.9.3.3.1 Network Address/Port Translators-Protocol Translators (NAPT-PT)

The use of NATs for interworking between IPv4 hosts and IPv6 hosts has similar problems as using NATs with private IPv4 addresses for extending the IPv4 address space.

In the UTRAN, bearer control (exchange of IP address/UDP port) will be performed using signalling such that IP addresses are included in the payload of signalling messages. The bearer control messages tell a UTRAN host what destination address to use to send data to the peer UTRAN host. An IPv4 host will not be able to use an IPv6 address received from an IPv6 peer host. There must be an Application Level Gateway (ALG) that intercepts the bearer control message and changes the transport parameters to the appropriate IP version. This must be done in coordination with the NAT so that the addresses in the traffic packets are changed according to the address put in the bearer control message.

ALGs and NATs are undesirable. They add complexity and degrade performance. This technique also requires that there be a pool of IPv4 addresses available that the NAT can use to translate IPv6 addresses. In addition, the NAPT-PT provides a single point of failure since all inbound and outbound traffic pertaining to a session must traverse the same NAPT-PT router. This increases costs since the reliability must be high.

The advantage of NAPT-PTs over other interworking techniques is that it allows more efficient use of IPv4 addresses. This is because one IPv4 address can be used for multiple IPv6 hosts by mapping IPv6 hosts to different UDP ports for the same IPv4 address. Other interworking techniques require an IPv4 address be mapped to an IPv6 host. One key disadvantage of NAPT-PTs is the need for ALGs.

~~6.9.1~~6.9.3.3.2 Stateless IP/ICMP Translation Algorithm (SIIT)

SIIT provides a method for interworking that doesn't require ALGs. However, it does require that an IPv6 host must be dynamically assigned a temporary IPv4 address that is used for the time of the session. The IPv6 host provides the IPv4 peer with the temporary IPv4 address using UTRAN bearer control. The IPv4 host uses this address for traffic packets. When the packets reach the SIIT router, the temporary IPv4 address is mapped to the IPv6 host address. The packet is then tunnelled from the SIIT router to the IPv6 host.

The IPv4 host provides the IPv6 host with an IPv4 address using UTRAN bearer control. For traffic, the IPv6 host maps this IPv4 address to an IPv6 address, which causes the packet to be routed to a SIIT router. The SIIT router will translate the mapped address back to the IPv4 address and forward it to the IPv4 host.

The SIIT technique allows multiple SIIT routers in a network so it does not cause a single point of failure like with the NAPT-PT technique.

This technique requires that the operator have a pool of IPv4 addresses available. It also requires that the traffic is routed through a SIIT router and the IP headers are translated which can have an impact on performance.

When an IPv4 address is assigned to an IPv6 node, it's necessary for the SIIT routers to be provided the address mapping between the assigned IPv4 address and the IPv6 address. This requires a protocol from the AIIH server assigning the IPv4 address to the SIIT router. AIIH stands for "Assignment of IPv4 Addresses to IPv6 Hosts" and is a DHCPv6 server with extensions.

~~6.9.1~~6.9.3.3.3 Dual stack

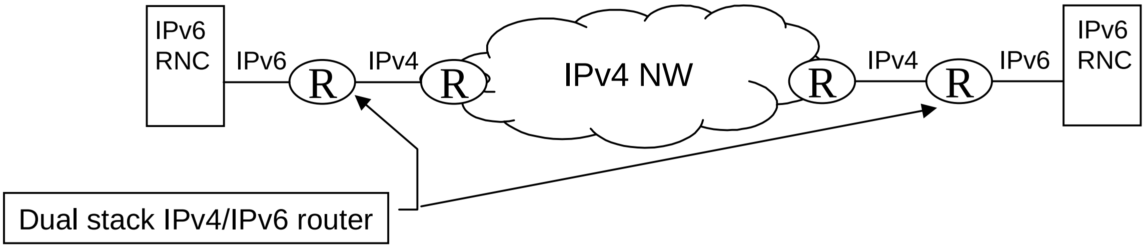
It's also possible for new nodes to deploy a dual stack when migrating to IPv6. The IPv4 stack can be used toward an existing IPv4 node and the IPv6 stack can be used toward IPv6 nodes.

Dual stack hosts also require that the operator have a pool of IPv4 addresses still available in order to assign one to the host when it must communicate with an IPv4 host. This can be a problem if the operator is running out of IPv4 addresses. Dynamic IPv4 address assignment also requires the use of a DHCPv6 server.

It's also necessary to keep track of which UTRAN hosts use IPv4 and which use IPv6 in order to know which type of address information to provide in the bearer control signalling.

~~6.9.1~~6.9.3.4 Tunneling

Where there is only an IPv4 network available, IPv6 UTRAN traffic can be transported over the network using tunnelling. As shown in the following figure, this requires that only the first-hop routers be IPv6 capable. Techniques have been developed in the IETF to determine the appropriate tunnel endpoints.



The use of tunnelling will be common in the IP UTRAN anyway for various reasons including:

1. Multiplexing of small packets into larger packets using PPPMUX and tunnelling with L2TP.
2. Virtual Private Networking for security and quality of service control.

Therefore, requiring tunnelling for transporting IPv6 packets over an IPv4 network is not a drawback.

~~6.9.1~~6.9.3.5 Summary

There is a good case for using only IPv6 for IP UTRAN hosts:

1. There are advantages to deploying IPv6.
2. The UTRAN is a closed IP network in that UTRAN applications only communicate with each other, not to applications in other networks such as the Internet and so could be a good place to deploy IPv6.
3. There is a strong advantage to avoid IPv4/IPv6 transition techniques for UTRAN hosts since they add complexity and impact performance. They also require that an operator have a pool of IPv4 addresses available.
4. The disadvantage of using IPv6 is that, where only an IPv4 network exists, the IPv6 traffic must be tunnelled over it. However, tunnelling will commonly be used for other purposes anyway in the UTRAN transport network.

It is true that other applications in a UTRAN node besides the UTRAN applications may need transition mechanisms between IPv6 and IPv4. An example of this would be an OAM application. The following scenarios are possible:

1. A client could be upgraded to IPv6 and must interwork with an existing IPv4 server in the operator's network. These applications are not as sensitive to performance considerations as the UTRAN applications so the interworking mechanisms are not a problem.
2. The servers could be upgraded to IPv6 along with the clients.
3. The clients could be run on hosts different than those of the UTRAN applications and continue to use IPv4 to avoid the need for interworking.

Also, the release '99 Iu interface already supports IPv4. For this interface, a dual stack should be required though it should be recommended that the Iu interface be upgraded to IPv6 when the IP UTRAN is deployed.

Inter-working between IPv4 and IPv6 is possible and will have to be mastered by operators, like ISPs, and vendors. However, when this inter-working can be avoided, it simplifies the overall IP network management and configuration.

One case avoiding any interworking is to deploy new IP networks with IPv6 only, when new equipment has to be installed to build it. Reasons why the standard shall allow using IPv4 equipments, when they are available are:

- Since IP technology is a good solution to mix several applications on the same common infrastructure, re-use of existing IP networks shall be possible.
- No specific feature of IPv6 is required by the RNL. No addressing shortage is expected when a private network is used for UTRAN,
- Allowing IPv4 makes IP Transport in UTRAN deployment independent from any IPv6 deployment.

6.10 Backward compatibility with R99/Coexistence with ATM nodes

It should be investigated how to inter-work the user plane between IP and AAL2/ATM interfaces including inter-working with a node that supports only AAL2/ATM interfaces, and how to interwork the control plane between IP and ATM interfaces.

~~6.9.1~~6.10.1 General

An IP UTRAN node should not be required to support AAL2/ATM UTRAN interfaces in order to interoperate with AAL2/ATM UTRAN nodes. The solution for interoperating between a UTRAN node with only IP interfaces and a release '99 and later AAL2/ATM UTRAN node should be performed only in the transport network layer (TNL) in order to maintain transport independence for the Radio Network Layer. The separation of RNL and TNL is an architectural principle in [1]. This means that the UTRAN RNL applications must not be affected nor should any interworking be required in the UTRAN RNL control plane or user plane when interworking between different transport technologies.

As shown in Figure 22 there are principally 3 cases (3-5) where interworking between IP and ATM nodes on Iub and Iur is necessary.

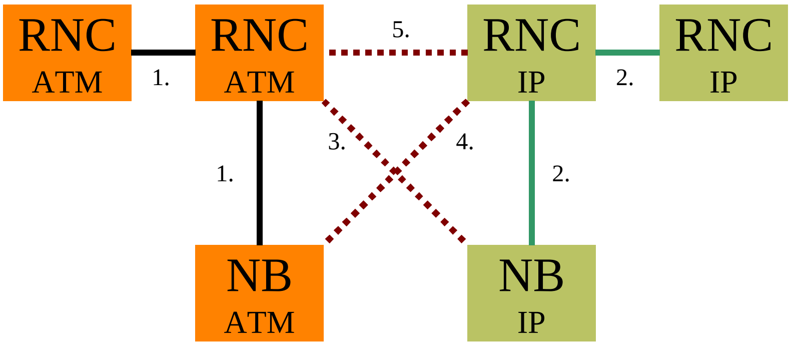


Figure 22. Interworking cases

The cases of interconnecting can be summarized as follows:
1. Iub/Iur - All ATM
2. Iub/Iur - All IP
3. Iub - ATM RNC with IP Node B
4. Iub - IP RNC with ATM Node B
4. Iur - IP RNC with ATM RNC

When an operator is migrating from ATM to IP, for example, a newly deployed UTRAN node should be allowed to support only IP interfaces and still be able to interwork with ATM UTRAN nodes. It should not be required to support AAL2/ATM UTRAN interfaces in order to interoperate with AAL2/ATM UTRAN nodes. This is the case for the following reasons:

1. Otherwise, all Release 4 RNCs having connectivity with both ATM NEs and IP NEs terminating RNL protocols would need to support both types of interfaces .
2. There may be manufacturers that want to supply only UTRAN nodes with one transport technology (such as IP-only nodes) but interwork with existing ATM nodes terminating RNL protocols in the operators network.
3. When an operator is migrating from ATM to IP, the newly deployed nodes would need to also support ATM interfaces to interwork with the legacy ATM nodes terminating RNL protocols. This means that the ATM network is being extended, which defeats the original purpose.

~~6.9.1~~6.10.2 Interworking Options

A design goal for the IP transport option within Rel.4 is to minimize the effects on the RNL ([1], sec. 5.2). The fact that an R99 node can be connected without having been upgraded to Rel.4 must be taken into account.

In the following three potential interworking options (dual stack operation, and TNL IWU) should be considered:

~~6.9.1~~6.10.2.1 Dual Stack operation within Rel.4 RNCs

Within the dual stack option a Rel.4 RNC must provide both stacks. Generally, it is assumed that only RNCs should provide both types of interfaces, so that Node Bs are either IP or ATM nodes. Nevertheless, for interworking case 3, where an IP based Node B is connected with a R99 RNC, also an interworking on Iub would be necessary. Within a pure IP or ATM environment the RNC must only provide one type of interface.

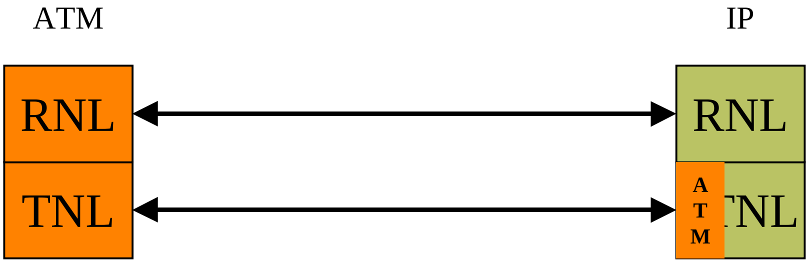


Figure 23. Dual Stack operation within Rel.4 RNCs

A Rel.4 IP node that needs to communicate with a pure ATM node (R99 or later) requires the complete ATM/AAL2 protocol stack. Beneficial of such an dual stack solution is, that it does not require a TNL control protocol on IP side. On Iub this solution would be quit sufficient, but on Iur there may be certain cases where a simple IWF or dual stack operation are not sufficient and an interworking unit (IWU) will be needed. (If interworking case 3 and 4 should be supported, also on Iub an IWU would be needed.)

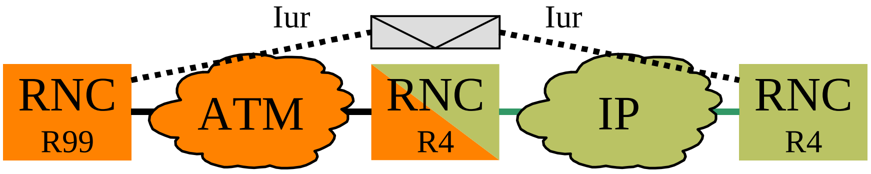


Figure 24. Full Meshed Iur

In the network, that is shown in Figure 24, are some RNCs pure IP based, some RNCs are pure ATM based and some RNCs are dual stacked. Assuming a network configuration where a pure IP based RNS borders on a pure ATM based RNS, the Iur interface between both RNSs must be supported.

A dual stacked RNC with an IWF in the middle would be able to communicate on both networks but would not be able to combine both parts of the network. In that case either an interworking unit is needed or a configuration as shown in Figure 24 is not possible and every RNC needs to support both interface types (IP and ATM).

~~6.9.1~~6.10.2.2 Transport Network Layer IWU

Also an TNL IWU can either be placed somewhere between the connecting nodes or can be integrated within one node.

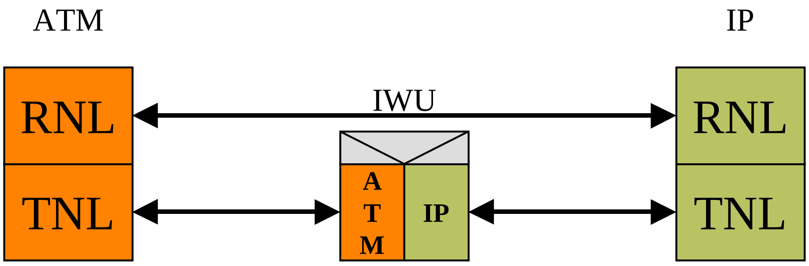


Figure 25. Transport Network Layer IWU

On transport network layer the IWU must support the translation between ATM and IP transport formats and QoS requirements. It must hold all states of active connections.

Although it is conceivable that a pure IP TNL could work without a TNL control protocol a simple TNL IWU would probably require a TNL control protocol. At least this depends on the agreed addressing scheme for the IP transport.

~~6.9.1~~6.10.2.2.1 Issue on TNL IWU control protocol

The following two figures show an example of a radio link setup request on Iur between an R99 and Rel.4 IP RNC. The first example, where the SRNC is a R99 and the DRNC is a Rel.4 IP RNC, avoids the usage of an TNL control protocol due to an appropriate choice of the binding ID and transport layer address within the RNSAP messages. In the second example, where the SRNC is a Rel.4 IP and the DRNC is a R99 RNC, the usage of a TNL control protocol is unavoidable.

Figure 26 and Figure 27 show the relevant information exchange on RNSAP and the involved primitives and messages of the AAL2 signalling protocol regarding [2] for each example.

In the first example the R99 SRNC requests a radio link setup. The Rel.4 DRNC RNL requests from its TNL resources for the new connection and receives an appropriate transport layer address and a binding ID. For example, the BID would be the UDP port, where the TNL is waiting for the new connection, and the transport layer address (TLA) would be a the code point (CP) that terminates at the IWU and identifies the DRNC. Therefore the Rel.4 TNL must have the knowledge that it is communicating with an ATM node. It provides an CP instead of an IP address and encodes the necessary information in a way that allows the IWU to establish the IP path later on. Within the radio link setup response message the UDP port number can be transported within the binding ID. Both information's, TLA and BID, are transmitted via ALCAP to the IWU. The IWU maps code points to IP addresses and extracts the port number out of served user generated reference (SUGR). The mapping between code points and IP addresses must be configured by O&M within the IWU and within the TNL of the IP node. The IWU is than able to establish a UDP connection and to complete the ALCAP connection setup. Some ATM specific information's like the link characteristics get either lost or translated into an IP equivalent IE.
Failure behaviour is FFS.

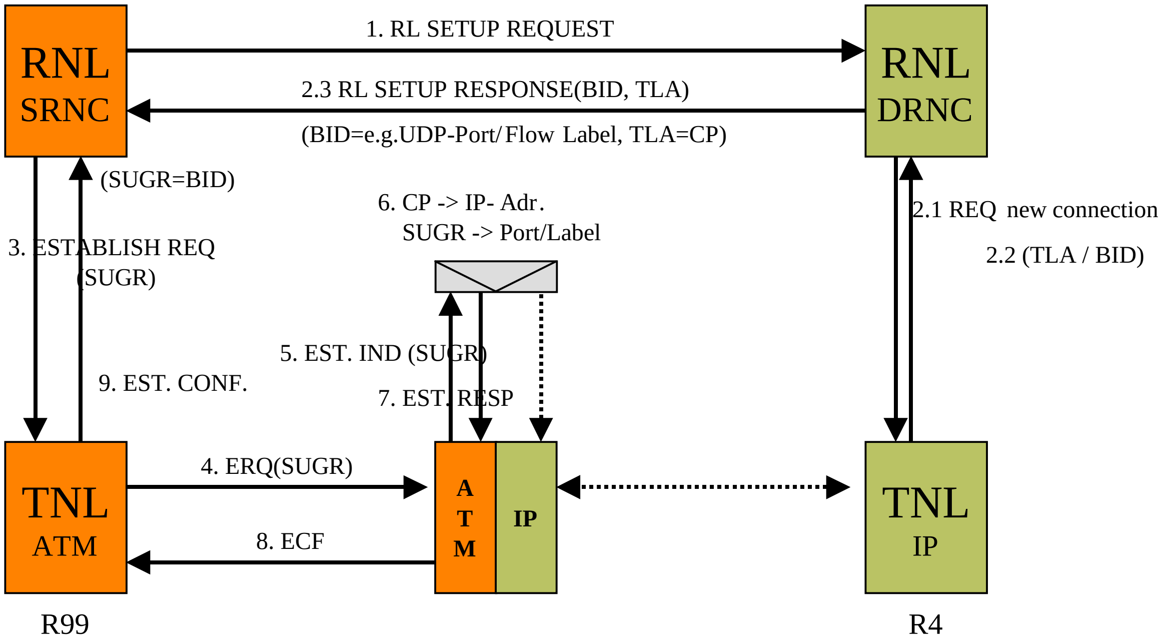


Figure 26. Example 1: RNSAP: DCH RL Setup, SRNC = R99; DRNC = Rel.4

Note: in this case the IWU must always send data to the DRNC before the DRNC can transmit data towards the SRNC because the DRNC does not know to which IP address/UDP port to send data before receiving this first data.

In the case where the Rel.4 IP RNC requests a radio link setup from the R99 RNC, the R99 RNC is not aware of the fact that it is communicating with an IP node. Beside, it must choose the binding ID completely free (e.g. without the knowledge what ports are free on the IWU or the IP RNC). The Rel.4 SRNC can map the TLA to an appropriate IP address but it can not map the binding ID to an appropriate UDP port number. Trying to map the binding ID to the port numbers results either in assigning a large number of IP addresses to both, the IP RNC and the IWU, or restricting the binding ID space within the R99 RNCs. Even if a trade off between numbers of needed IP addresses and restrictions of the binding ID space could be found, information like the link characteristics that can't be generated within the IWU itself must be transmitted somehow to the IWU. For that purpose a TNL control protocol also on the IP side of the connection is necessary.

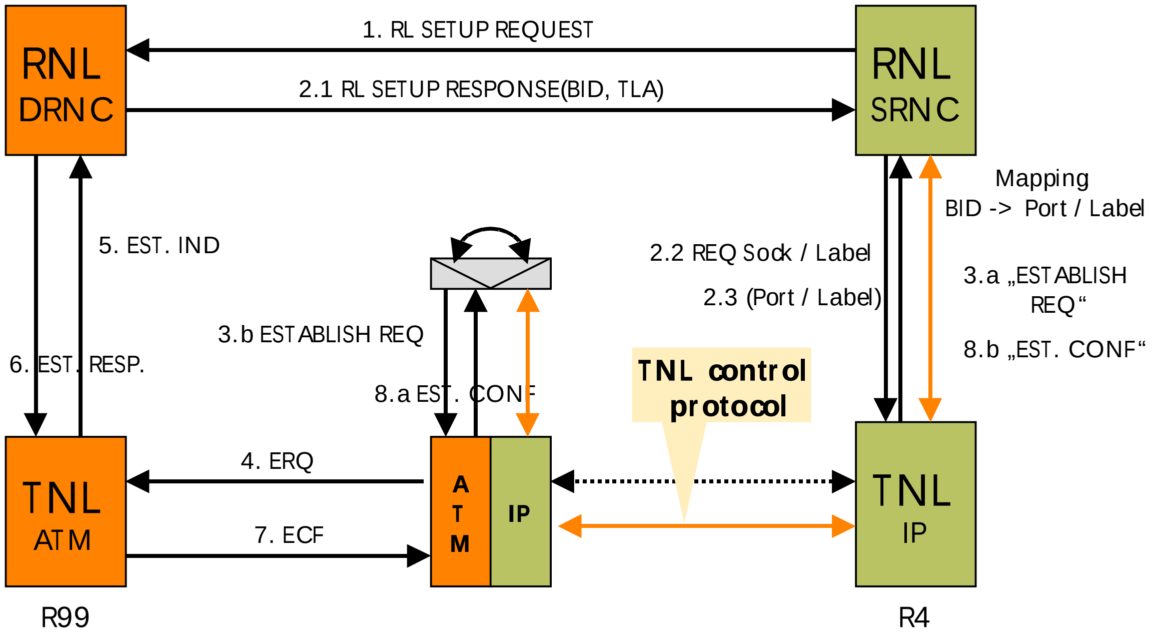


Figure 27. Example 2: RNSAP: DCH RL Setup, DRNC = R99; SRNC = Rel.4

~~6.9.1~~6.10.3 Conclusion

- It must be clarified if an interworking on Iub (interworking case 3 and 4) should be supported or if an dual stack operation is sufficient for the Iub interface.
- For the Iur interface an IWU is needed, which is either integrated within an UTRAN node or a independent box.
- An IWU that works only on TNL requires a TNL control protocol that must be specified within the standard.

~~6.9.1~~6.10.4 UTRAN Architecture considerations

The following figures show the Iur interface where an IP UTRAN node is introduced. They are shown as interworking examples for the purpose of this discussion. In Figure 1, a release ‘99 SRNC is shown with an Iur interface to an IP DRNC. In Figure 2, an IP SRNC is shown with an Iur interface to a release ‘99 DRNC.

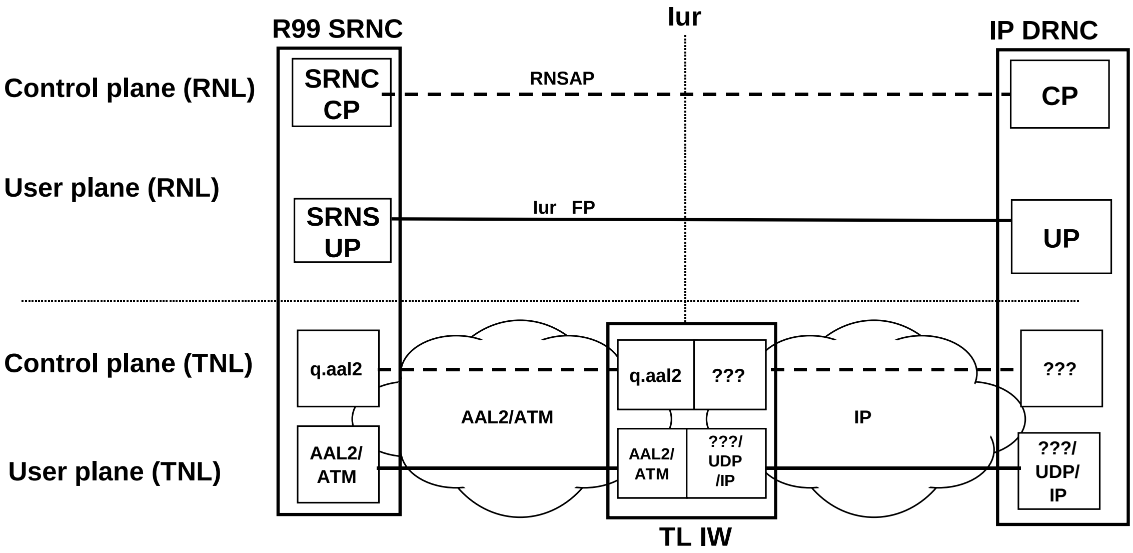


Figure 28: Transport network layer interworking with release ’99 SRNC

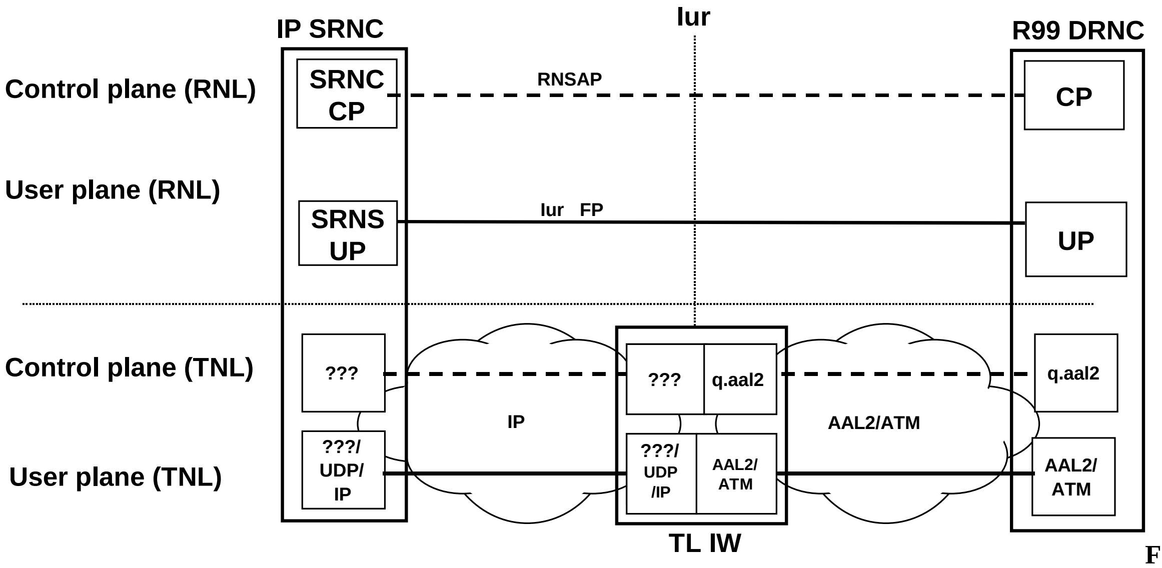


Figure 29:

Transport network layer interworking with IP SRNC

These figures show the separation between the RNL control plane, the RNL user plane, the TNL control plane, and the TNL user plane. The IP protocols and the need for a TNL control plane protocol in the IP domain are yet to be determined so they are shown with question marks.

The following statements concerning interworking can be made based on the discussion and examples above:

1. IP and AAL2/ATM UTRAN nodes use different address and flow identification types. The appropriate types must be provided to the appropriate nodes when establishing a transport bearer.
2. A release ’99 SRNC will initiate q.aal2 connection signalling and expect a response when establishing a transport bearer.
3. A release ’99 DRNC will expect to receive q.aal2 connection signalling when a transport bearer is being established by the SRNC.
4. A transport network interworking function is required in the transport network. This function could be implemented in a third UMTS node with both IP and AAL2/ATM interfaces, for example.

~~6.9.1~~6.10.5 ATM/IP Interworking solution proposal.

~~6.9.1~~6.10.5.1 Bearer control proposal

For exchanging transport layer information between IP UTRAN nodes, the RNL signalling should be used (RANAP, RNSAP, NBAP) without a Transport Network Control Protocol.

For establishing transport connections between an IP UTRAN node and an ATM UTRAN node, a Transport Network Layer interworking function should be used in the transport network. This function would be implemented in a third node (such as an RNC) that has both ATM and IP interfaces.

In order to interwork with the q.aal2 signalling used by the AAL2/ATM node, an IP ALCAP will be used. It is proposed to use Session Initiation Protocol (SIP) signalling with Session Description Protocol (SDP) parameters. SDP supports both IP and ATM parameters. SIP is proposed since it is an IETF signalling protocol and is used to carry SDP.

Since a node must know what type of interface to communicate with, a Network Type parameter should be added to the RNL signalling. The following table shows how the Network Type parameter is used.

R'99	R4 IP	R4 ATM	Action
SRNC	DRNC		R4 DRNC knows the SRNC is R'99 because of missing transport parameters in RL setup req. R4 IP RNC does interworking steps.
DRNC	SRNC		SRNC sends IP transport parameters that R'99 DRNC will ignore. SRNC must know that it is receiving ATM parameters. Absence of network type in response will indicate that it is R'99. R4 IP RNC does interworking steps.
SRNC		DRNC	R4 DRNC knows SRNC is R'99 because of missing transport parameters in RL setup req.
DRNC		SRNC	SRNC sends ATM network type parameter that R'99 DRNC will ignore. SRNC must know that it is receiving ATM parameters from DRNC. Absence of network type will indicate that it is R'99.
	SRNC	DRNC	SRNC sends IP transport parameters. SRNC must know that it is receiving ATM parameters. It can know this from the network type parameter in DRNC response. SRNC then performs interworking steps.
	DRNC	SRNC	SRNC sends ATM network type. R4 DRNC knows its ATM from the network type and performs interworking steps.

~~6.9.1~~6.10.5.2 Bearer control between IP and ATM nodes signalling examples

The following figures provide signalling diagrams that show how the interworking can be achieved with this proposal. The Iur is shown as an example. UDP ports are shown for connection identifiers as an example.

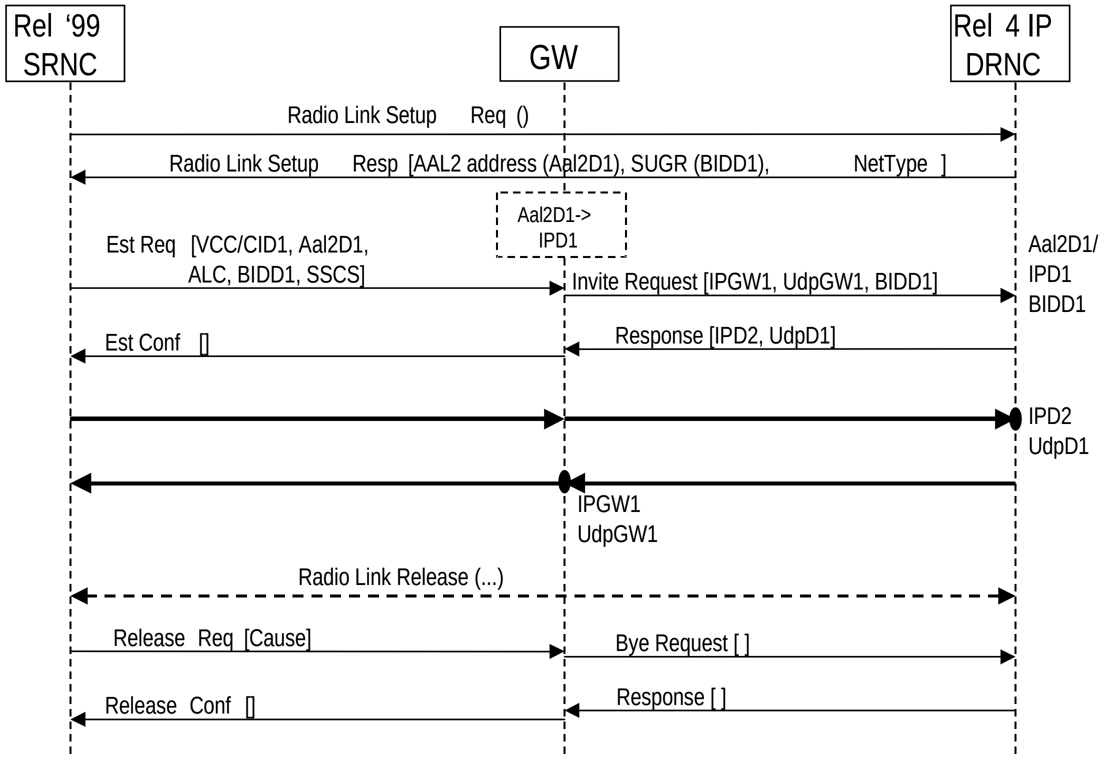


Figure 30: Interworking between an AA2/ATM SRNC and an IP DRNC

Notes:

1. The rel ’99 SRNC sends radio link setup. There is an SCTP Signalling Gateway for interworking the SCTP/IP signalling to ATM signalling.

The IP DRNC node responds with ATM transport parameters. The IP DRNC must have both ATM and IP addresses assigned to it.

The SRNC uses q.aal2 signalling to establish a connection towards the DRNC based on the address received in the RL Setup Response. The TNL IW node is along the route to the DRNC.

When the TNL IW function receives the q.aal2 set up message it determines that the destination node is an IP node.

The TNL IW function translates the ATM address to the IP address for the DRNC and sends a SIP Invite message to the IP DRNC. The Invite message includes the IP address and UDP port for traffic toward the TNL IW node. Also included is the binding ID so that the DRNC can correlate the transport signalling with the RNL signalling.

The IP DRNC responds to the Invite message. Included in the response message is the IP address and UDP port for traffic towards the IP DRNC.

When the TNL IW node receives the Response message it sends the q.aal2 confirmation message to the ATM SRNC.

To release the connection, the SRNC sends a q.aal2 Release Request.

When the TNL IW function receives the request it sends a SIP Bye Request to the IP DRNC.

The IP DRNC responds to the Bye Request and when the TNL IW function receives it, it sends the q.aal2 Release Confirm.

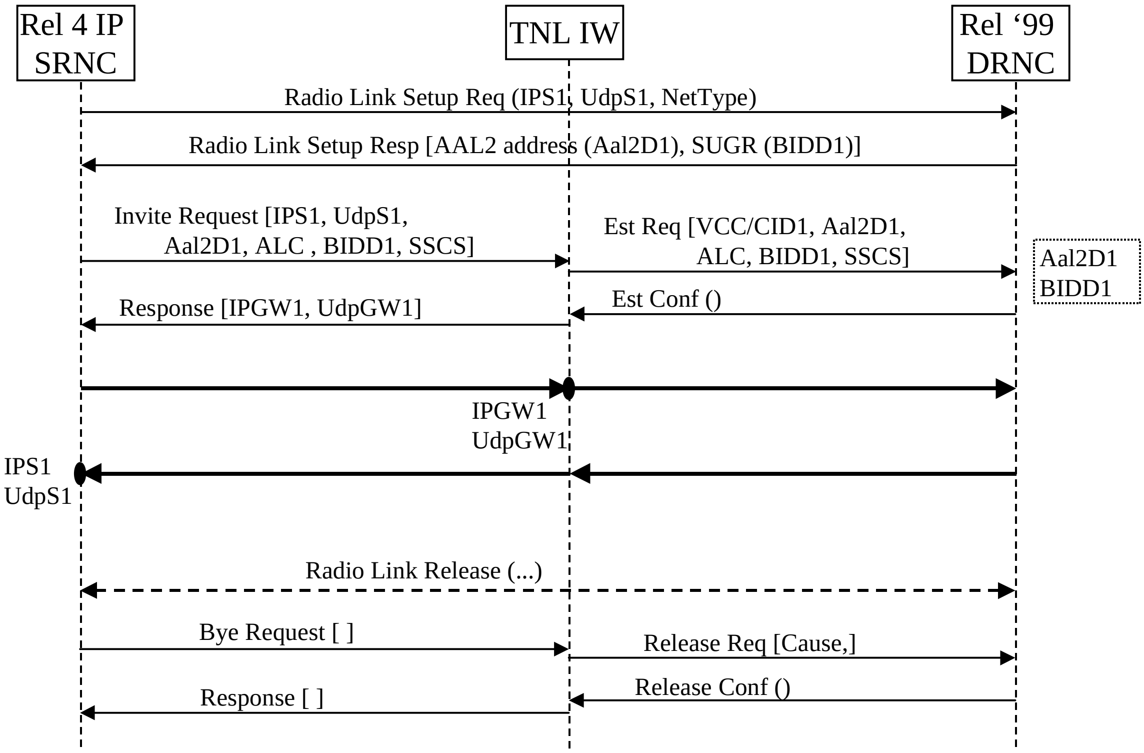

Rel 4 IP SRNC
TNL IW
Rel '99 DRNC
Radio Link Setup Req (IPS1, UdpS1, NetType)
Radio Link Setup Resp [AAL2 address (Aal2D1), SUGR (BIDD1)]
Invite Request [IPS1, UdpS1, Aal2D1, ALC , BIDD1, SSCS]
Est Req [VCC/CID1, Aal2D1, ALC, BIDD1, SSCS]
Aal2D1
BIDD1
Response [IPGW1, UdpGW1]
Est Conf ()
IPGW1
UdpGW1
IPS1
UdpS1
Radio Link Release (...)
Bye Request [ ]
Release Req [Cause,]
Response [ ]
Release Conf ()

Figure 31: Interworking between an AA2/ATM SRNC and an IP DRNC

Notes:

2. The rel 4 SRNC sends radio link setup. An SCTP Signalling Gateway is used for interworking the SCTP/IP signalling and ATM signalling. The Setup message includes IP address, UDP port, and network type that will be ignored by the rel'99 DRNC.

The ATM DRNC node responds with the ATM transport parameters.

The SRNC sends a SIP Invite message to the TNL IW node. It includes the IP address and UDP port to be used for traffic towards itself. It also includes the ATM parameters received from the ATM DRNC so that the TNL IW function can establish an AAL2 connection with the ATM DRNC.

The TNL IW function initiates a q.aal2 establish request based on the parameters received from the SRNC.

The ATM DRNC responds to the q.aal2 establish message

When the TNL IW node receives the establish confirm message is sends a SIP response message to the IP SRNC. The response includes the IP address and UDP port used for traffic towards itself.

To release the connection, the SRNC sends a SIP Bye Request.

When the TNL IW function receives the request it sends a q.aal2 Release Request to the ATM DRNC.

The ATM DRNC responds to the Release Request.

When the TNL IW function receives it, it sends SIP response.

~~6.9.1~~6.10.1. Coexistence between Rel4 and R99 Iur Control Plane using SUA protocol

Section 6.7.2 describes the option of SUA as IP based signalling User Adaptation Layer in Iur Control Plane.

It is clear that SUA provides seamless functions and services as SCCP (from RNSAP point of view), and also, as advantage, SUA is optimised to be used over SCTP/IP, providing e.g. SCCP-to-SCTP/IP address translation. See [25] for further details.

~~6.9.1~~6.10.1.1. Connecting an Rel4 RNC to a R99 RNC

A way to interwork an Rel4 RNC to a R99 RNC is using signalling gateway. (this gateway can be embedded in the same physical equipment as an RNC) Using SUA, the RNSAP SAP is maintained for both TNL options since SCCP and SUA provide the same primitives and services to RNSAP, so no changes to RNSAP are needed to support both TNL options. With SUA, the RNL independence is maintained for Rel4 as in R99.

The signalling gateway would perform the L2/L1 to AAL5/ATM/L1 conversion. In this case SUA does not add any interworking problem, since the signalling gateway performs the domain conversion from SCCP to SUA/SCTP/IP and viceversa. Also, it is noted that the signalling gateway could come from any vendor, since all protocol used in both ends of the SG are standardized.

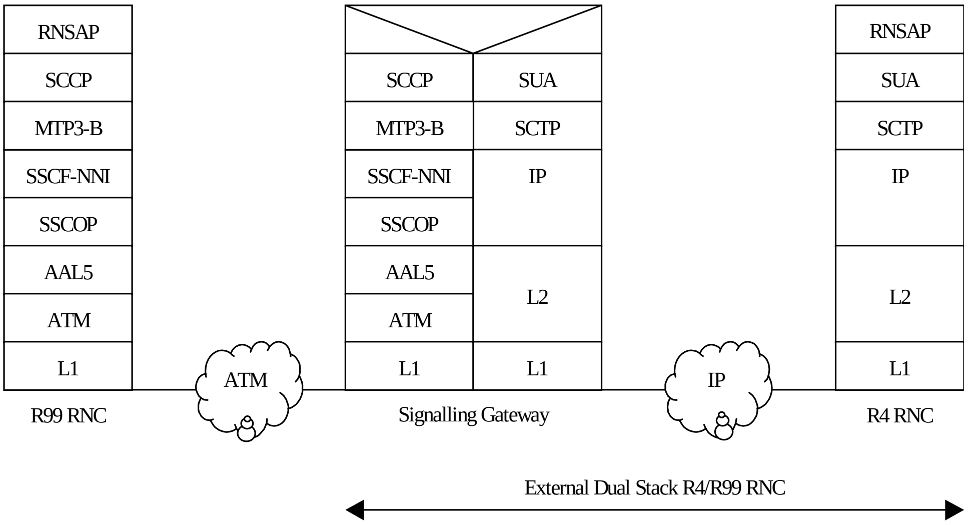


Figure 32: Interworking between External dual stack RNC and a Rel4/R99 RNC.

This option permits the providers and operators to handle the different interworking scenarios in an efficient way, e.g. several Rel4 RNCs sharing the same signalling gateway to a R99 only RNC through an IP network, or RNCs with embedded signalling gateways connected to R99 only RNCs and using both IP and ATM networks, while maintaining the RNSAP protocol as in R99.

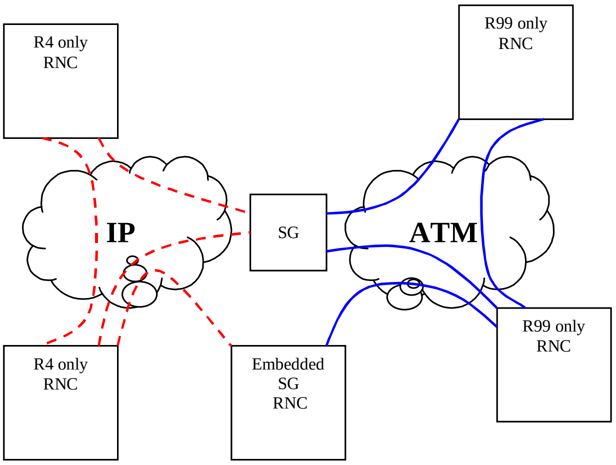


Figure 33: Possible interworking scenarios using SUA.

Summarising the conclusions:

- SUA maintains the same primitives and services as SCCP and is optimised to be used over SCTP/IP, e.g. including the SCCP-to-SCTP/IP address translation.
- There are no interworking problems between Rel4 and R99 Iur Control Plane when SUA protocol is used below RNSAP for Rel4 stack
- With SUA, the Signalling gateway approach also gives flexibility to both providers and operators to implement the interworking between the R99 and Rel4 releases, depending on transport network characteristics and equipment availability, while maintaining open interfaces (Rel4 and R99 Iur) in both R99 and Rel4 RNCs.

6.11 Synchronisation

Node synchronisation requirements for an IP based UTRAN nodes should be investigated including minimising delay variation and clock frequency differences between an application source and sink.

6.12 Security

This study area is related to security aspects.

~~6.9.1~~6.12.1 Security Threats

[43.] classifies between threats associated to the air interface, to the UE or to other part of the network. For the other part of the network, the identified threats are the following:

- Unauthorised access to data: traffic eavesdropping, receiver masquerading, unauthorised access to stored data, traffic flow analysis.
- Threats to integrity: manipulation of stored data, traffic or network elements by masquerading or any other way.
- Denial of service: physical or protocol intervention, abuse of emergency services.
- Repudiation: of charge, of traffic origin or delivery.
- Unauthorised access to services: by masquerading or misusing privileges or services.

~~6.9.1~~6.12.2 Security Operation in IP networks

~~6.9.1~~6.12.2.1 IPSec architectures

In IETF, security is a whole area of work, in which one group focuses especially on security architecture and IPSec protocol suite[44.], [45.]. IPSec is a protocol providing authentication and integrity protection in two different architectures:

- End-to-end security provisioning between hosts: this solution puts the complexity into the hosts;
- Gateway to gateway: IPSec is terminated in intermediate nodes (routers) that protect the data in a sub-part of the network that may be insecure.

When the security is provided from host to host, two modes are possible:

- Transport mode, in which integrity and authentication cover only transport protocol (above IP) and higher protocols.
- Tunnel mode, in which the IP header is also protected. That mode needs a second IP header to be present to allow routing.

The tunnel mode is the only possible solution for gateway to gateway architecture.
Both modes cause additional overhead per IP packet.

IPSec is a separate protocol in IPv4 but is fully integrated in IPv6. However its use is optional in IPv6. It is possible to provide security to IPv6 hosts without using IPSec in the hosts, for instance with gateway to gateway tunnel mode.

IPSec architecture assumes the existence of a Key management system. That system can be manually administered or controlled by IETF protocols like, ISAKMP [45.].

~~6.9.1~~6.12.2.2 SCTP Security features

SCTP (Stream Control Transmission Protocol) has been designed to transport signalling and control data on top of IP. It delivers a reliable transport service, like TCP. But it brings also some additional features.
It incorporates a cookie exchange mechanism at association establishment. That procedure was explicitly designed to prevent unauthorised connections to be set up at transport level.

~~6.9.1~~6.12.2.3 Firewalls and other systems

Beyond standard protocols and architecture defined by IETF, constructors have proposed their own security features in boxes often called “Firewalls”. They most often implement standard security solutions but they also incorporate additional functions.

This kind of equipment is mainly dependent on the State of the Art of any kind of security experts. The decision to use it is out of the scope of UMTS standardisation.

6.13 Iu-cs/Iu-ps harmonisation

This study area is related to the possibility of removing the Iu-cs/Iu-ps distinction in the user plane and in the control plane.

~~6.9.1~~6.13.1 GTP-U for Iu user plane

~~6.9.1~~6.13.1.1 Iu PS

With IP transport for UTRAN, GTP-U will be used on the IuPS interface as in release ´99. However, when real-time applications are considered and IP header compression is used, the GTP-U header is relatively large. There are currently 2 possible headers that can be used for GTP-U. One consists of 8 octets, the other 12 octets. In addition, there is the GTP’ protocol that is used for GPRS charging that uses a smaller header than GTP (6 octets). Flags in the header indicate which header is being used.

IP header compression allows the IP/UDP headers to be compressed to 2 – 5 octets. If a sequence number is needed with GTP-U, the header size is 12 octets. For example, for a 40-octet payload, the GTP-U overhead alone can be over 20% of the packet size (IP/UDP/GTP/payload) when a sequence number is included.

For real-time applications much of the GTP-U header is not needed. A header definition for GTP-U should be defined that is optimized for real-time applications.

~~6.9.1~~6.13.1.2 Iu CS

GTP-U could be used for the IuCS interface over IP transport for the following reasons:

- The requirements for the real-time IuPS applications and the real-time IuCS applications are the same.It results in the same protocols being used for both IuCS and IuPS (harmonization). GTP-U will be used for the IuPS interface so it will already be available for the IuCS in the Media Gateway.
- It is under the control of 3GPP. Any desired modifications for optimization can be handled by 3GPP.

An alternative to GTP-U is to use RTP:

According to RFC 1889, RTP is designed to satisfy the needs of multi-participant multimedia conferences. It therefore provides more functionality than is required and has a large overhead of 12 octets.

The RTP header can be compressed but the decompressor needs to be updated for every packet so it adds processing load over IP/UDP compression alone.

The advantage of RTP is that it is an IETF protocol. However, this protocol will be terminated where the framing protocol is terminated at the UTRAN interface endpoint. It is therefore not important that it is an IETF protocol.

~~6.9.1~~6.13.1.3 Simplification of GTP header for real-time applications

The release ‘99 GTP header is shown below.

	Bits							
Octets	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
1	Version			PT	(*)	E	S	PN
2	Message Type							
3	Length (1^{st} Octet)							
4	Length (2^{nd} Octet)							
5	Tunnel Endpoint Identifier (1^{st} Octet)							
6	Tunnel Endpoint Identifier (2^{nd} Octet)							
7	Tunnel Endpoint Identifier (3^{rd} Octet)							
8	Tunnel Endpoint Identifier (4^{th} Octet)							
9	Sequence Number (1^{st} Octet)$^{1) 4)}$							
10	Sequence Number (2^{nd} Octet)$^{1) 4)}$							
11	N-PDU Number$^{2) 4)}$							
12	Next Extension Header Type$^{3) 4)}$							

Notes: The GTP-U header is 8 octets unless one or more of the E, S, or PN bits are set, then it is 12 octets. The (*) bit is unused.

- **Version field**: This field is used to determine the version of the GTP protocol.
- **Protocol Type (PT):** This bit is used as a protocol discriminator between GTP (when PT is ‘1’) and GTP’ (when PT is ‘0’). GTP’ is described in GSM 12.15.

- **Extension Header flag (E)**: This flag indicates the presence of the Next Extension Header field when it is set to '1'. When it is set to' 0', the Next Extension Header field either is not present or, if present, must not be interpreted.
- **Sequence number flag (S)**: This flag indicates the presence of the Sequence Number field when it is set to '1'. When it is set to '0', the Sequence Number field either is not present or, if present, must not be interpreted.
- **N-PDU Number flag (PN)**: This flag indicates the presence of the N-PDU Number field when it is set to '1'. When it is set to '0', the N-PDU Number field either is not present, or, if present, must not be interpreted.

The following simplifications to the GTP-U header can be made in order to reduce overhead for real-time applications:

1. The length field can be removed since GTP level multiplexing is not done and the length information is available from lower layers.
2. A one-octet message type field is larger than required for GTP-U and is based on GTP-C requirements. There are only a few GTP-U messages. However, for this discussion, it is assumed that GTP-U signalling messages are always sent with a full header. All GTP-U messages use a TEID value of 0.

   The following table shows the messages used by GTP-U:

GTP-U Message	TEID
Echo Request	0
Echo Response	0
Error Indication	0
Supported Extension Headers Notification	0

   The following text defining the use of the TEID field in the GTP-U header is from the GTP specification, 29.060.

   - *TEID: Contains the Tunnel Endpoint Identifier for the tunnel to which this T-PDU belongs. The TEID shall be used by the receiving entity to find the PDP context, except for the following cases:*
   - *The Echo Request/Response, Supported Extension Headers notification and the Version Not Supported messages, where the Tunnel Endpoint Identifier shall be set to all zeroes.*
   - *The Error Indication message where the Tunnel Endpoint Identifier shall be set to all zeros.*
3. The sequence number in GTP is larger than required for real-time applications.
4. The N-PDU number will never be needed since it is used only for non real-time applications to guarantee that packets are not lost or duplicated during the Routing Area Update procedure and SRNS Relocation.
5. GTP includes a 4 octet Tunnel Endpoint Identifier to identify a flow. This is a larger than required. It is proposed to use a 2 octet TEID.
6. Header Extensions do not need to be supported for real-time applications. If extensions are needed for an application, the full GTP header should be used.

~~6.9.1~~6.13.1.4 Proposed GTP-U header for real-time applications

It is assumed that the TEID/IP address is used to identify a flow (RAB/PDP context). GTP-U signalling messages will use the full GTP header.

The following table shows a proposed GTP-lite header for real-time applications:

Octet	8	7	6	5	4	3	2	1
1	Version			PT0	PT1	*	S	*
2	TEID (1st Octet)							
3	TEID (2nd Octet)							
4	sequence number							

- **Protocol Type 0, 1 (PT0, PT1)**: These flags indicate how the header and protocol should be interpreted as shown in the following table.

PT1	PT0	Meaning
0	0	GTP'
0	1	GTP-full header
1	0	GTP-lite header

1	1	Undefined

- **Sequence number flag (S)**: This flag indicates the presence of the Sequence Number field when it is set to ‘1’. When it is set to ‘0’, the Sequence Number field is not present.

Figure 34: GTP-lite header

~~1~~7 Agreements and associated agreed contributions

This section documents agreements that have been reached and makes reference to contributions agreed in RAN-WG3 with respect to this study item. This section is split according to the above mentioned Study Areas.

7.1 External standardisation

7.2 QoS differentiation

The user plane protocol stack standardised for IP transport shall not preclude any of the following two network configurations:

- QoS differentiation provided by the IP network on a hop-by-hop basis, and
- QoS differentiation provided on an edge-to-edge basis.

The standard shall not preclude any of the following alternatives within the transport network:

- Flow per flow or aggregate classification,
- Classification based on packet per packet information or on flow addressing information,
- Classification made on information provided by the transport bearer initiator

The needed information for quality of service differentiation between several UTRAN flows shall be available at the IP layer used for RNL flow adressing. The UTRAN NEs shallprovide this QoS information to this IP layer.

7.3 Transport network bandwidth utilisation

~~6.9.1~~7.3.1 Multiplexing

No additional multiplexing layer/functionnality shall be specified between UDP/IP and the UTRAN Frame Protocols since adequate solutions exist below IP achieving the UTRAN requirements.
All multiplexing scenarios, introduced in section 6.4.1.1.1 Figure 12, bring specific benefits and shall be supported for IP Transport in UTRAN.

7.4 User plane transport signalling

7.5 Layer 1 and layer 2 independence

The use of one exclusive L2 protocol shall not be standardised for IP transport. One or a limited set of L2 protocols shall be specified and required. The use of any L2 protocol fullfillingthe UTRAN requirement towards layer one and two, shall not be precluded by the standard. The PPP protocol [11.] shall be supported by each UTRAN NE for IP transport. UTRAN NEs having interfaces connected via slow bandwidth links like E1/T1/J1 shall also support Header Compression and the PPP extensions PPPmux[3] [10.] and ML/MC-PPP [20.], [21.]

7.6 Radio Network Signalling bearer

7.7 Addressing

7.8 Transport architecture and routing aspects

IP Hosting is a necessary function for a network element supporting of the UTRAN functions (Node B, RNC).
UTRAN NEs shall have at least one IP address, onto one or several IP subnets.
No restriction is imposed, regarding routing domains and autonomous systems.

[3] The mandate for PPPmux is currently a Working Assumption

7.9 Backward compatibility with R99/Coexistence with ATM nodes

7.10 Synchronisation

7.11 Security

7.12 Iu-cs/Iu-ps harmonisation

7.13 Iur/Iub User plane protocol stacks

7.14 Iu-cs/Iu-ps user plane protocol stacks

7.15 IP version issues

~~1~~8 Specification Impact and associated Change Requests

This section is intended to list the affected specifications and the related agreed Change Requests. It also lists the possible new specifications that may be needed for the completion of the Work Task.

8.1 Specification 1

~~6.9.1~~8.1.1 Impacts

This section is intended to make reference to contributions and agreements that affect the specification.

~~6.9.1~~8.1.2 List of Change Requests

This section lists the agreed Change Requests related to the specification.

8.2 Specification 2

~~6.9.1~~8.2.1 Impacts

~~6.9.1~~8.2.2 List of Change Requests

~~1~~9 Project Plan

9.1 Schedule

Date	Meeting	[expected] Input	[expected]Output
September 27-29, 2000	RAN3 IP Ad Hoc #1	- Requirements, - Transport Network Architecture and Routing, - Bandwidth Utilisation, - RNL flow identification, - Iur/Iub User Plane Stack Definition	- Agreements on the Requirements.
October 16 -20, 2000	RAN3#16	- Iur/Iub User plane transport signalling, - Radio Network signalling, - Addressing for control plane, - QoS Differentiation,	- Agreements on Transport Network Architecture. - Agreements on addressing for control plane, - Agreements on Transport signalling and Radio Network signalling. -

November 6-8, 2000	RAN3 IP Ad Hoc#2	- Iur/Iub User Plane further details and comparison - IP/ATM networks compatibility, - Iu User Plane stack. - L1/2 independence,	- Agreements on the Iur/Iub/Iu user plane stacks, and RNL flow identification. - Agreements on IP/ATM networks compatibility principles.
November 20 - 24, 2000	RAN3#17	- Iur/Iub/Iu User Plane further details, - Iucs/Iups harmonisation, - Security, - Synchronisation, - CRs on RANAP/RNSAP/NBAP/ALCAP.	- Informative version of TR 25.933 for RAN#10
15 - 19 January 2001	RAN3#18	According to previous agreements: - CRs on Iur/Iub/Iu user plane, - CRs on Iucs/Iups harmonisation, - CRs on IP/ATM networks compatibility, - CRs on Security, synchronisation, L1/L2 independence, - Other CRs	- CRs agreed in principle.
26 February - 02 March 2001	RAN3#19	- Updated CRs.	For submission to RAN#11: - Final TR version - All CR's completed. - ASN.1 for xxxAP completed.

9.2 Work Task Status

	Planned Date	Milestone	Status
1.	September 2000 (IP Adhoc #1)	Requirements definition (5)	Almost complete
2.	September 2000 (IP Adhoc #1)	Transport Architecture and routing aspects (6.8)	Work in progress, partly agreed
3.	October 2000, (RAN3#16)	Radio Network Signalling Bearer (6.6)	Contribution available, not discussed
4.	November 2000, (IP Adhoc #2)	Transport network bandwidth utilisation (6.3)	Work in progress
5.	November 2000, (IP Adhoc #2)	User plane transport signalling (6.4)	Contribution available, not discussed
6.	November 2000, (IP Adhoc #2)	QoS Differentiation (6.2)	Work in progress
7.	November 2000, (IP Adhoc #2)	Addressing (6.7)	Work in progress
8.	November 2000, (IP Adhoc #2)	Backward compatibility with R99/ Coexistence with ATM nodes (6.9)	Work in progress
9.	November 2000, (IP Adhoc #2)	Layer 1 and Layer 2 independence (6.5)	Work in progress
10.	November 2000, (RAN3#17)	Synchronisation (6.10)	Not started
11.	November 2000, (RAN3#17)	Iu-cs/Iu-ps harmonisation (6.12)	Not started
12.	November 2000, (RAN3#17)	Security (6.11)	Not started
13.	November 2000, (RAN3#17)	External Standardisation (ref 1, 6.1)	Work in progress

~~1~~10 Open Issues

~~1~~11 History

Document history		
V0.0.1	2000-05	First proposal
V 0.1.0	2000-06	Version agreed at RAN3#13 (Hawaii).
V0.1.1	2000-07	Version including changes agreed at RAN3#14 (Helsinki) in: - R3-001706 (partially) - R3-001712 (partially)
V0.2.0	2000-08	Version agreed at RAN3#15 (Berlin).
V0.2.1	2000-09	Editor's proposal
V0.2.2	2000-10	Version including: - new sections 6.2, 7.13, 7.14 - text agreed at RAN3 IP-Transport AdHoc#1 in Swindon from Tdocs 2428, 2398, 2427, 2410, 2401, 2412,2426, 2402, 2421, 2405, 2411, 2414, 2400. - Editorial modifications in 5.2 and 6.4.
V0.3.0	2000-10	Version agreed at RAN3#16 (Windsor). Additional decision: The simulation model should be included in the next draft V0.3.1.
V0.3.1	2000-10	Editor's proposal: addition of a new Annex A for the description of the Simulation Model, with the agreements taken at RAN3 IP AdHoc#1 in Swindon, UK.
V0.3.2	2000-11	Editor's proposal: Version including changes agreed at RAN3 IP Adhoc#2 (Paris).
V0.4.0	2000-11	Version agreed at RAN3#17 (Chicago).
V0.4.1	2001-01	Inclusion of text agreed at RAN3#18, coming from tdocs 120, 121, 160, 161, 180, 182, 191, 241.
V0.4.2	2001-02	Inclusion of text agreed at IP-Transport Adhoc#3 meeting, coming from tdocs 10402, 10421,10425, 10412, 10416, 10420, 10414, 10409, 10423, 10406, 10407, 10425
V0.5.0	2001-03	Draft version including agreements of RAN3#19 first day.
V1.0.0	2001-03	Version agreed at RAN3#19 to be submitted to RAN#11.
Rapporteur for 3GPP RAN TR 25.933 is:		
Nicolas Drevon, Alcatel nicolas.drevon@alcatel.fr		
This document is written in Microsoft Word version 97 SR-2.		

Annex A: Simulation Model

A.1 Introduction

The simulation model is intended to give criteria to compare different IP based Iub User Plane protocol stacks. ATM/AAL2 will be used as a baseline case for comparison.

A.2 Simulation scenarios

Four different traffic mixes are defined for the simulation runs:

- 100% voice,
- 100% data,
- 80% voice & 20% data, with 5 voice users per data user
- 20% voice & 80% data, with 3 data users per voice user

Data rates are 64, 144 and 384 Kbps.

Throughput will be specified as a percentage of used bandwidth at source level, not including TNL protocol overheads (but TNL protocol overhead is included in simulation).

NBAP and O&M traffic will not be included in simulations.

A.3 Simulation model framework

The general simulator model can be split in four parts which are nearly independent from each other.

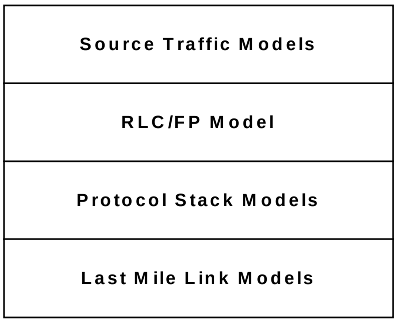


Figure 35: General Simulator Model

This modular concept allows an efficient reuse of simulator modules for the investigation of different proposed protocol stacks and provides transparency for comparison.

A.4 Source Traffic Models

A.4.1 Speech source model

For simulation, speech sources are based on AMR codecs with only the 12.2 kbps mode. Each AMR 12.2 kbps source is modelled with an ON/OFF model for DTX, having the following statistics:

- Voice Call Duration Distribution: Exponential, mean: 120 sec
- Duration of On-state Distribution: Exponential, mean: 3 sec.
- Duration of Off-state Distribution: Exponential, mean: 3 sec.

A.4.2 Data source model

Two equivalent data source models can be used:

A.4.2.1 Data Source Model 1

Each data user is modelled as a WWW application, consisting of a sequence of file downloads. Each file download is modelled as a sequence of packet arrivals, having the following statistics.

Data model: Each web browsing download has Pareto distributed file size with a parameter α = 1.1, mean 12,000 bytes, minimal file size 1858 bytes, maximal file size 5,000,000 bytes. The p. d. f. (probability density function) is

$$f(x) = \begin{cases} \dfrac{\alpha \cdot k^{\alpha}}{x^{\alpha+1}}, & k \le x \le m \\ \dfrac{k^{\alpha}}{m^{\alpha}}, & x > m \end{cases}$$, where *α=1.1*, *k=1858*, and *m=5,000,000*

Chop the file into IP packets with size of 1500 bytes (and one less than 1500 bytes if size is not a multiple of 1500). Inter-arrival time of IP packets is exponentially distributed with mean of 8.3 ms. This yields about 1445.78 Kbps IP packet arrival rate (larger than 64, 144, 384 Kbps data transmission rates). Therefore, the inter-arrival time has no significant impact on simulation results.

Reading time is defined by the time that the last bit of a file leaves from the RNC (G. data queue on the Figure 1) to the time that the first bit of the next file arrives to the "C. RLC data buffer". The distribution of reading time is exponential with mean 12 sec.

A.4.2.2 Data source model 2

Interactive data traffic is mainly generated by WWW serving. As for the background traffic, the number of active users will be assumed to be constant. The parameters are listed in table 1.

Class	Parameter	Values	Remark
Transmission	bit rate [kbit/sec]	64, 144, 384	
Packet Call	# of packets per call distribution	Geometric	
	# of packets per call mean	25	
	packet inter arrival time distribution	Exponential	packet inter arrival time within a packet call
	packet inter arrival time mean	0.0083 sec	
	inter packet call time distribution	Exponential	reading time between to consecutive packet calls
	reading time mean	12 sec	
Packet	packet size mean	480 bytes	Pareto PDF: $\frac{\alpha k^{\alpha}}{x^{\alpha+1}}$ If X is a Pareto distributed random variable then packet sizes are computed as P=min(X,m). Parameters are not independend.
	packet size distribution	limited Pareto with α=1.1, k=81.5, m=66666	

Table 1: Interactive data traffic

A.5 RLC/FP model

<u>1. Voice Traffic</u>

The RLC layer is transparent for voice traffic. Therefore, no overhead and no functionality is required in the simulation model for voice traffic in the RLC layer.

In the frame protocol, flows are composed to streams, which results in additional overhead as summarised in Table 2. The frame protocol PDU has a header of 2 Bytes and a trailer of 2 Bytes which results in a general 32 bit overhead per PDU. Each flow in the PDU has an overhead of 8 bits for the TFI, according to ref. [8.]. In the frame protocol, each flow will be padded to 8 bit boundaries which results in additional overhead.

Class	Parameter	Value/Size	remark
Stream	overhead per stream packet (CRC + CFN)	32 bit	overhead added per stream packet, regardless of its contents
Flow	overhead per flow (TFI)	8 bit	overhead added once per flow in each stream packet

Table 1: Parameters for Stream Overhead

The following example explains the FP PDU generated for the 12.2 kbit/s AMR mode in ON state.

- □ Header CRC, CFN 2 bytes
- □ 4 flows (DCH0-3) for class A, class B, class C and signalling
  - □ 4 x 8 bit TFI 4 bytes
  - □ 81 bit class A + padding 11 bytes
  - □ 103 bit class B + padding 13 bytes
  - □ 60 bit class C + padding 8 bytes
  - □ signalling 0 or 10 bytes
- □ Payload CRC 2 bytes

Signalling is assumed every 300 ms.

<u>2. Packet data Traffic</u>

The RLC/FP splits the input packets into segments and also aggregates segments to new packets. While the input queue is not empty one or more new packets are created per TTI. There size is chosen from a connection specific set of possible packet sizes. Depending on the signalled TFS, multiple small packets or one large packet are used to satisfy the transmission demand. If required, padding packets are used as input to extend the new packets to the smallest possible allowed size.

Class	Parameter		Value/Size	remark
Scheduler	inter packet time		TTI of the connection	
Packet Control	packet overhead		16 bit	Length Indicator
Segment Control	segment size set		{0, 320} bits	
	segment overhead		16 bit	
Transport Format	Peak data rate		64 kbps	
			144 kbps	
			384 kbps	
	RLC Buffer size		256 kByte	
	TTI		40 ms	20 ms optional
	TF set size	64 kbps	{0,1,2,3,4,6,8} x 336 bits	TF set for 20 ms see TSGR1#14(00)0844
		144 kbps	{0,1,2,4,8,16,18} x 336 bits	
		384 kbps	{0,1,2,4,8,12,16,20,24,32,40,48} x 336 bits	

Table 2: Packet data traffic RLC/FP model parameters

A.6 Protocol Stack Models

A.6.1 Overview

By investigating the protocol stacks for IP transport e.g. PPPmux or CIP one can find that the modules needed for implementation are:

- □ Header compression (FFS)
- □ Packetizer
- □ Queues
- □ And the scheduler providing the prioritisation for the voice traffic

In the different protocol stacks these functions are provided by different layers. For the performance study these functionality can be modelled equally for all protocol stacks. The performance depends only on:

- □ Header overhead per stream which can not be shared
- □ Header overhead per container to be sent over the link
- □ The position of the packetizer
- □ The position of the queues and scheduler

The overhead can be introduced by parameters. The positions for the packetizer and the queues with the scheduler depend on the chosen implementation of the protocol stack. The implementations can be optimised per protocol stack depending on the QoS strategy. Two possible structures are shown in Figure 36 and Figure 37. The structure implemented in the simulator model shall be given together with the simulation results.

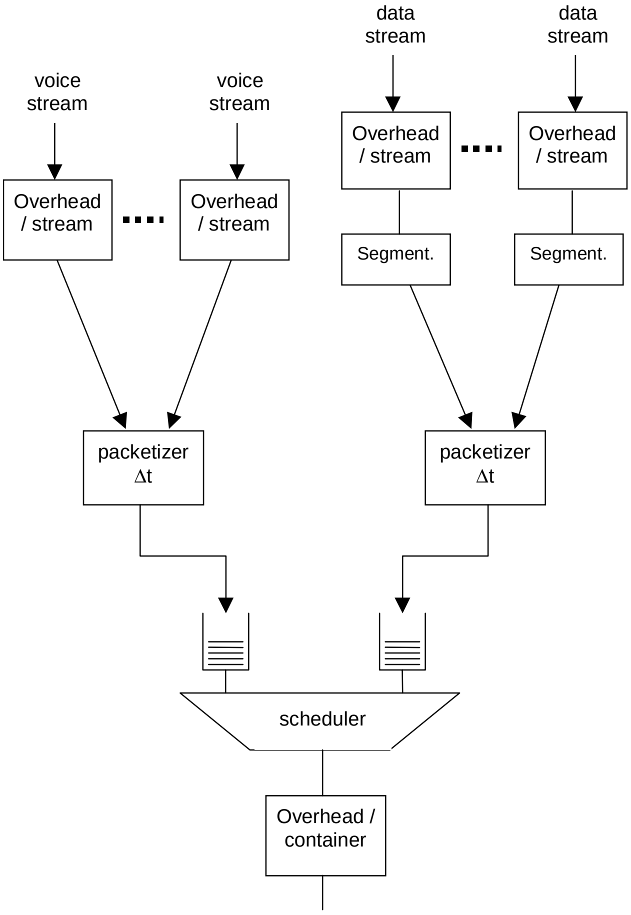


Figure 36: Implementation Structure, Variant 1

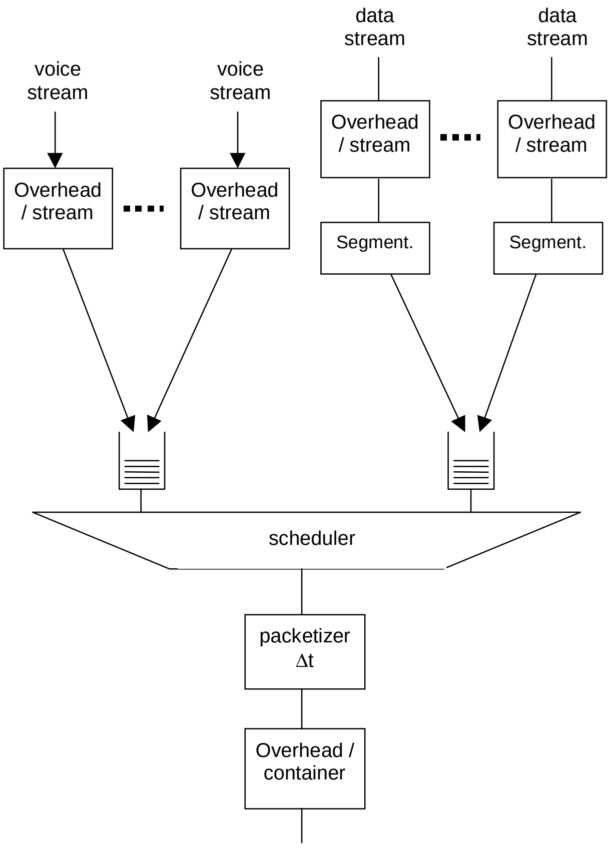


Figure 37: Implementation Structure, Variant 2

A.6.2 Module Functions

1. Header Compression (FFS)

[Editor's note: contributions are invited]

2. Packetizer

The packetizer composes the input packets to containers up to a maximum size or up to a maximum time. This process introduces additional delay to the streams.

Class	Parameter	Example Value	remark
Container Control	time out	0.003 sec	maximum delay time
	max container size	2400 bit	maximum container size

Table 3: Packetizer Parameters

3. Queues

Due to the limited bandwidth of the Last Mile Link Model queues must be provided. This process introduces additional delay to the streams.

Class	Parameter	Example Value	remark
Queue Control	Strategy	FIFO	
	max. size	infinite	no packet loss

Table 4: Queue Parameters

4. Segment Function

The segment function splits the input packets to segments down to a fixed size. The related overhead shall be introduced on a per stream or per container basis depending on the implementation. This process introduces no delay to the streams.

Class	Parameter	Value	remark
Segment Control	Segment size	tbd	

Table 5: Segment function Parameters

5. Scheduler

The scheduler is a functional entity which provides prioritised service for two input queues. In our model one voice queue and one data queue are assumed. The voice queue shall be serviced until empty, at which time the data queue shall be serviced until the voice queue has become non-empty or the data queue is also empty. Voice packets cannot preempt data packets.

A.6.3 Examples

In the following table examples are given how the Protocol Stack Model could be used for protocols already introduced in above sections.

Protocol	Structure	Overhead/stream		Overhead/container	
Protocol 2	Variant 2	CUDP PPPlen	3 byte 1 byte	PPPID PPPmux HDLC	1 byte 1 byte 3 byte
Protocol 1	Variant 1	CIP	3 byte	CUDP PPP HDLC	4 byte 1 byte 3 byte

Table 6: Examples

A.7 Last Mile Link Models

A point-to-point connection between the Edge-Router and the NodeB is considered as Last Mile Link. It shall be modelled as infinite server providing a fixed service rate.

Class	Parameter		Value	remark
Link Model	n*E1	n=1	1.92 Mbps	
		n=2		
		n=3		

Table 7: Link Parameters

A single E1 link is assumed.

A.7 Performance criteria

The most important performance criteria are delay and link utilisation. The delay figures contain the packetisation delay, the queuing delay and the transmission delay per individual stream. Confidence intervals shall be calculated based on the results of several independent simulation runs. Empirical studies have shown that about 10 simulation runs are the optimum to minimise computation time by still giving good statistical confidence. The duration of one simulation run depends on the required confidence interval size. It is not possible to make an accurate forecast about the required simulation time to achieve good statistical confidence. Therefore, the simulation time must be increased if the results are not meaningful. It is important for the reporting of simulation results that confidence intervals are included.

Statistic	Confidence Level	Remarks
99.9-percentile voice delay	0.95	
link utilisation		Confidence level not important, can be calculated analytically
99.9-percentile transmission delay	0.95	
99.9-percentile packetisation delay	0.95	

Table 8: Performance criteria