WCDMA RAN W12 Troubleshooting

STUDENT BOOK
LZT1380619 R2A

ericsson.com

DISCLAIMER

This book is a training document and contains simplifications. Therefore, it must not be considered as a specification of the system.

The contents of this document are subject to revision without notice due to ongoing progress in methodology, design and manufacturing.

Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

This document is not intended to replace the technical documentation that was shipped with your system. Always refer to that technical documentation during operation and maintenance.

This document was produced by Ericsson.

- The book is to be used for training purposes only and it is strictly prohibited to copy, reproduce, disclose or distribute it in any manner without the express written consent from Ericsson.

This Student Book, LZT1380619, R2A supports course number LZU1088650.

Table of Contents

Intentionally Blank

1 WCDMA RAN System

Objectives

After this chapter the participants will be able to:

1 Understand the system concepts, redundancy and configurations in CPP

1.1 Use COLI/AMOS commands to understand Fault Tolerant Core (FTC) concept and Reliable Program Uniter concept

1.2 Understand how link redundancies work in both ATM and IP based transport interfaces

1.3 Understand the concept of moveable Connection End Point (Mv CEP)

1.4 Understand the File system in a CPP based node

1.5 Be able to perform an emergency recovery of a CPP based node from a backup placed outside of the node

1.6 Describe the Subscriber Capacity RNC and RBS with New Hardware and Software.

1.7 Be able to interpret Managed Object attributes to explain how interfaces are configured from a CPP based node using Element Manager and AMOS

Figure 1-1: Objectives of Chapter 1

Intentionally Blank

1 WCDMA RAN Network Architecture

WCDMA RAN contains several Radio Network Subsystems (RNS), each of which providing all the WCDMA RAN Services over a specific geographic area. An RNS contains one Radio Network Controller (RNC) and several RNC-RBS subsystems, besides the link to the core network.

The WCDMA RAN node types are RNC, Radio Base Station (RBS) and RAN Aggregators (RXI), all based on the Ericsson Connectivity Packet Platform (CPP). They contain hardware and software resources needed to implement WCDMA RAN services.

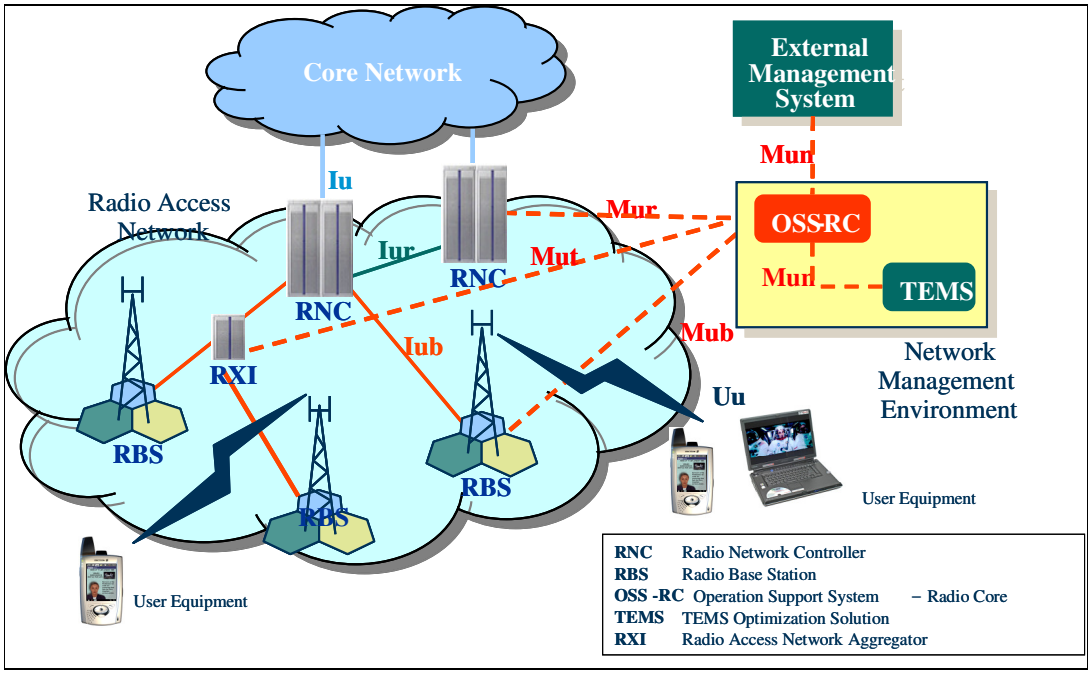


Fig2ure 1-2: WCDMA RAN Interfaces

- Uu: Radio interface between UTRAN and the User Equipment (UE)
- Iub: Interface between the RNC and Node B. Control layer information is relayed using NBAP (Node B Application Protocol) and Q2630.2 (Q.AAL2); user data is relayed using logical channels
- Iur: This is the logical interface between two RNCs. While logically representing a point-to-point link between RNCs, the physical implementation need not be a point-to-point link. This interface

would be used for the relay of intra-RNC handover. The control signaling is relayed using RNSAP.

- Iu: The Iu is the interface between the Core Network and WCDMA RAN, more precisely between the RNC node and the different domains of the Core Network. The terms Iu-cs and Iu-ps are used to indicate associations to the circuit switched network and the packet switched network, respectively.
  The term Iu-bc is used to indicate association to broadcast of unacknowledged messages.
  - the Circuit Switched domain, Iu-cs to MSC server (RANAP) and MGw (Iu User Plane)
  - the Packet Switched domain, Iu-ps to the SGSN
  - the Broadcast domain, Iu-bc from the Cell Broadcast Center (CBC)
- Mur: The Mur interface is the Operation and Maintenance interface between the RNC and OSS RC.
- Mub: The Mub interface is the Operation and Maintenance interface between the RBS and OSS RC.
- Mut: The Mut interface is the Operation and Maintenance interface between the RANAG and OSS RC.

1.1 Protocols Stacks on Interfaces

1.2 Iu Interface

The Iu interface capabilities include the following:

- Procedures to establish, maintain, and release Radio Access Bearers
- Procedures to perform Inter-Radio Access Technology handover
- Procedures for transferring of Non Access Stratum (NAS) signaling messages between UE and Core Network
- Location services by transferring requests from the Core Network to the WCDMA RAN and location information from the WCDMA RAN to the Core Network
- Simultaneous access to multiple Core Network domains for a single UE
- Mechanisms for resource reservation for packet data streams
- Broadcast of short text messages to defined geographical areas known as cell broadcast service areas.

The structure and protocol stack for Iu-cs and Iu-ps carried over ATM are illustrated in the following figures.

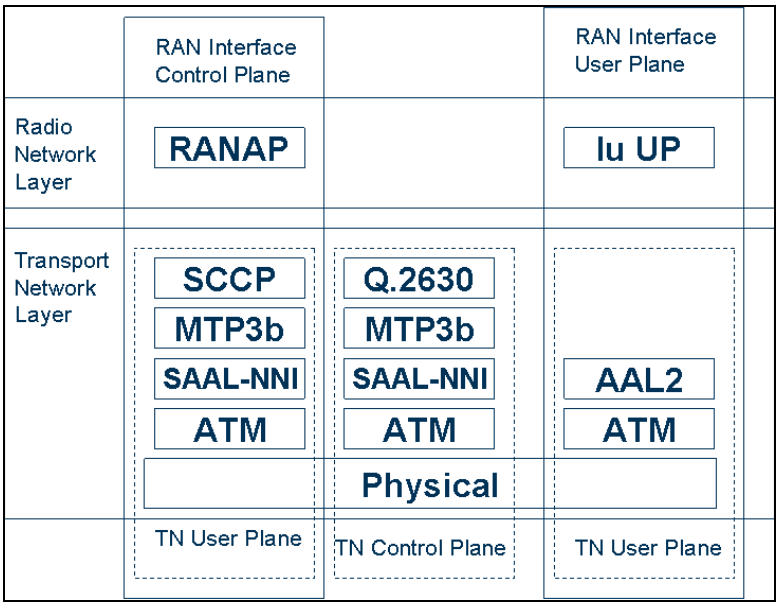


Figure 1-3: Protocol Stacks for Iu-cs over ATM

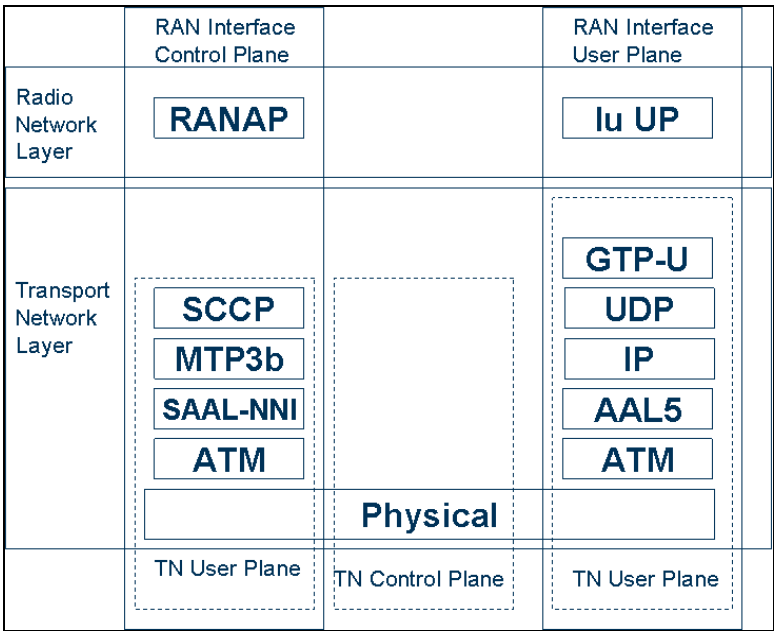


Figure 1-4: Protocol Stacks for Iu-ps over ATM

The structure and protocol stack for Iu-cs (signaling and user plane) carried over IP are illustrated in the following figure.

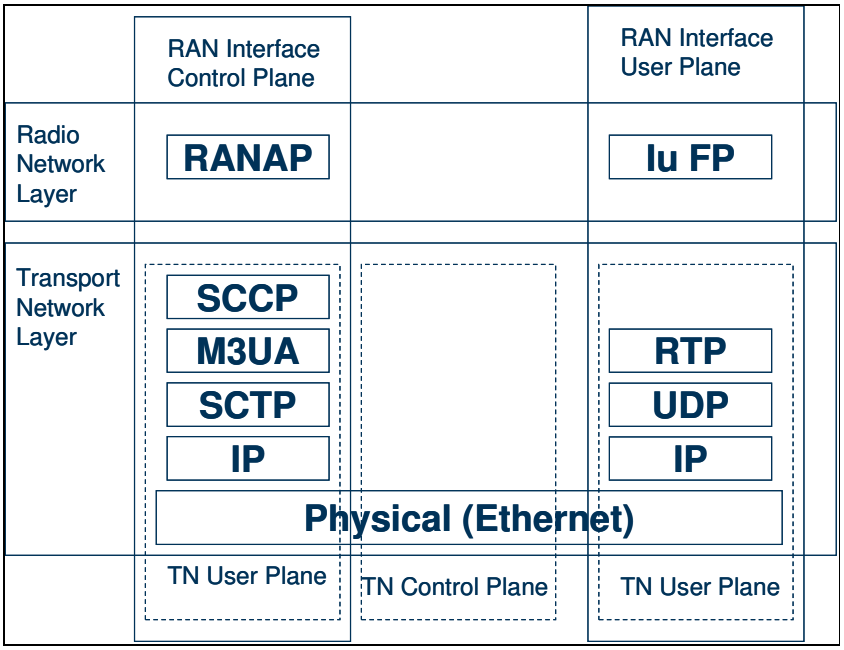


Figure 1-5: Iu-cs over IP over protocol stacks

Note: Q.2630 over IP for Iu-cs is also possible.

1.2.1 Control Plane:

The radio network protocol for control plane is called Radio Access Network Application Protocol (RANAP) and is used towards both the MSC node and the SGSN node. One RNC is connected to up to two Core Network parts per UE connection, using RANAP signaling, carried on SCCP for ATM and SCTP for IP. No connection when the RNC acts as a DRNC, one or two when the RNC acts as an SRNC (circuit and/or packet switched part).

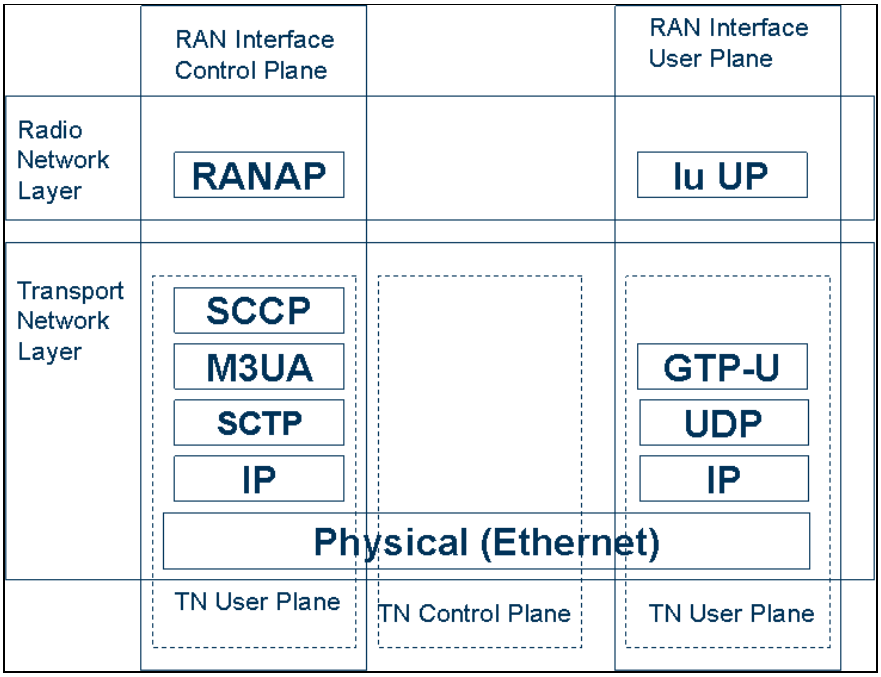


Figure 1-6: Iu-ps protocol stacks over IP

User Plane:

Two main types of User plane bearers are provided over the Iu:

- Towards the circuit switched Core Network for voice or Circuit Switched Data using AAL2 connections or using Real Time Protocol (RTP) over IP
- Towards the packet Core Network for IP traffic, using packet tunneling over Iu by the GTP-U protocol, carried by UDP/IP. This is carried over AAL5 when ATM is used.

One UE connection may be involved in both types of Radio Access Bearers. The Iu interface supports a user plane that is either based on a common channel (AAL5 or IP) or a dedicated communication channel (AAL2 connection). The common communication channel is used for packet data services. All circuit switched services use dedicated channels.

A common channel is used by several user connections, that is, they share the same connection. For a dedicated channel, a unique AAL2 connection is used for each user connection. The GTP-U/IP protocol is used for multiplexing the user information at a common channel.

The protocols for User Plane differ between the two Core Network domains. For the Packet Switched domain GTP-U and an IP/ATM or IP based transport network is used. For the Circuit Switched domain, Iu User Plane Protocol is used on AAL2 bearers or over an IP network.

1.3 Iur Interface

The Iur interface is a WCDMA internal interface for the communication between two RNC nodes (and between two RNSs). It is an open and standardized interface.

The interface contains a control plane for radio signaling and AAL2 connection establishment and a user plane supporting guaranteed QoS on ATM or IP. Note that all ATM user plane traffic is carried on the same type of AAL2 connection, packet data, voice, and Unrestricted Digital Information (UDI).

The figure below illustrates the protocol stacks for Iur interface with signaling carried over ATM.

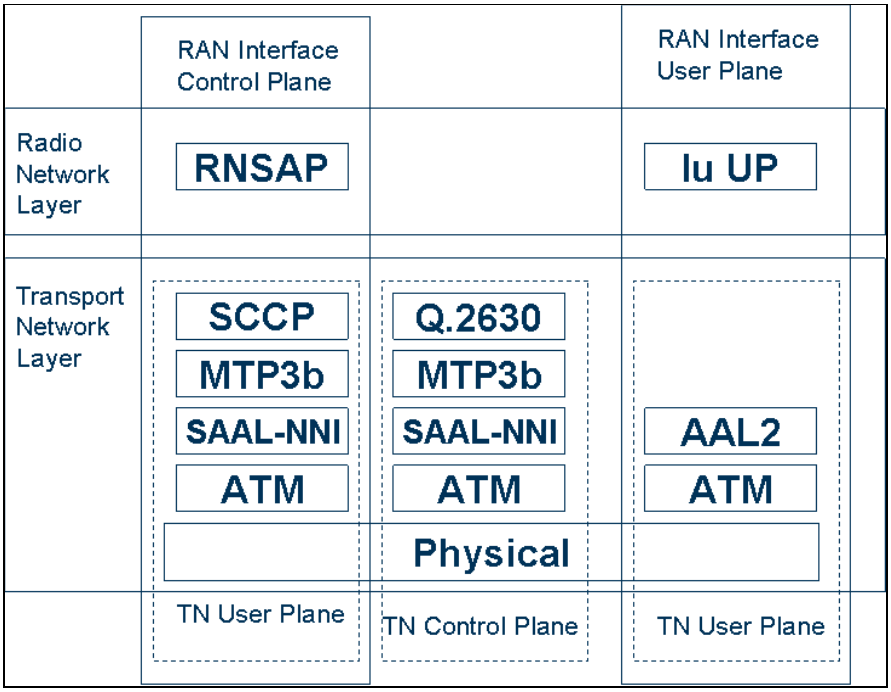


Figure 1-7: Protocol Stacks of Iur over ATM

Similarly, the figure below illustrates the protocol stacks for Iur interface with signaling and user plane carried over IP.

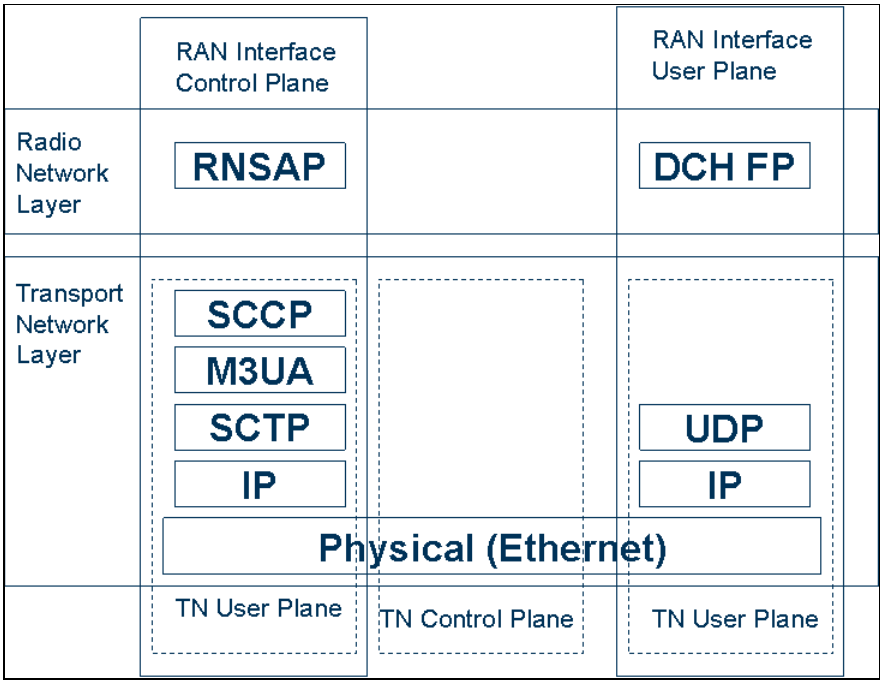


Figure 1-8: Iur (signaling and user plane) over IP protocol stack

Note: If the Iur User Plane is carried over AAL2/ATM, then the Q.2630 between the two RNCs can be carried over either ATM or over IP. The same is true for Iu-cs interface.

1.3.1 Control Plane:

SCCP is used to support signaling messages between two RNCs. The Radio Network Subsystem Application Part (RNSAP) is a user function of the SCCP connection. The RNSAP uses one signaling connection for each DRNC and UE while one UE can have one or more active radio links for the transfer of layer 3 messages. The SCCP signaling connection is requested by the SRNC when the SRNC needs dedicated resources from the DRNC. The control plane includes functionality for establishing AAL2 connections using Q.2630 and Q.2150.1

1.3.2 User Plane:

The DCH FP carries DCH data streams between the SRNC and the RBS through the DRNC

1.3.3 Synchronization Plane:

The following two types of synchronization are used over the Iur interface:

- Frame transport timing for DCH data streams.
- Downlink Synchronization Control Frames are used in a synchronization procedure to achieve or restore the synchronization of the DCH data stream in downlink direction.

1.4 Iub Interface

The information transferred over the Iub is categorized as follows:

- Radio application-related signaling

  The Iub interface allows the RNC and the RBS, for example, to negotiate about radio resources, to add and delete cells controlled by the RBS to support communication of a dedicated connection.

- Radio frames

  The Iub interface provides the means for transport of uplink and downlink radio frames between the RNC and the RBS.

- Quality estimation of uplink radio frames and synchronization data

  The macro-diversity combining function of the RNC uses RBS quality estimations of the uplink radio frames. Accurate time synchronization between the soft handover branches is also required, which includes frame synchronization and node synchronization as a base.

Figure below illustrates the protocol stacks for Iub interface carried over ATM.

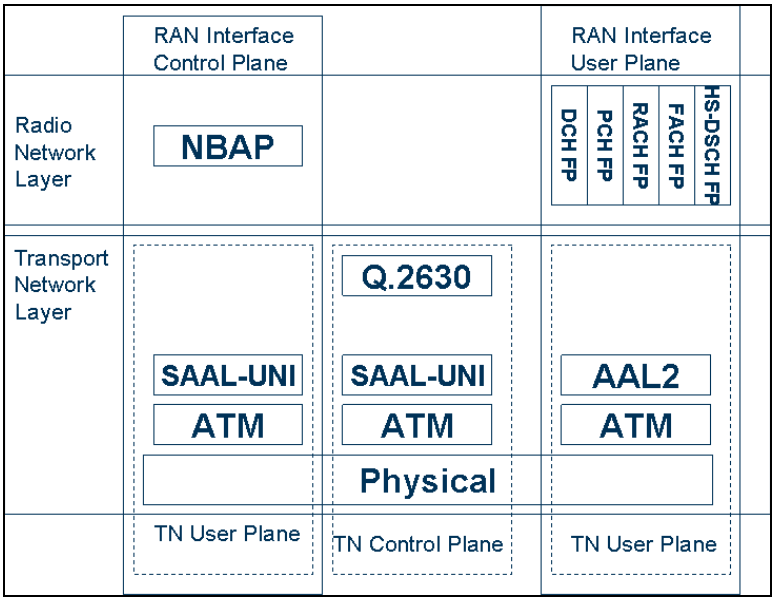


Figure 1-9: Protocol stacks for Iub over ATM

Figure below illustrates the protocol stacks for Iub interface carried over IP.

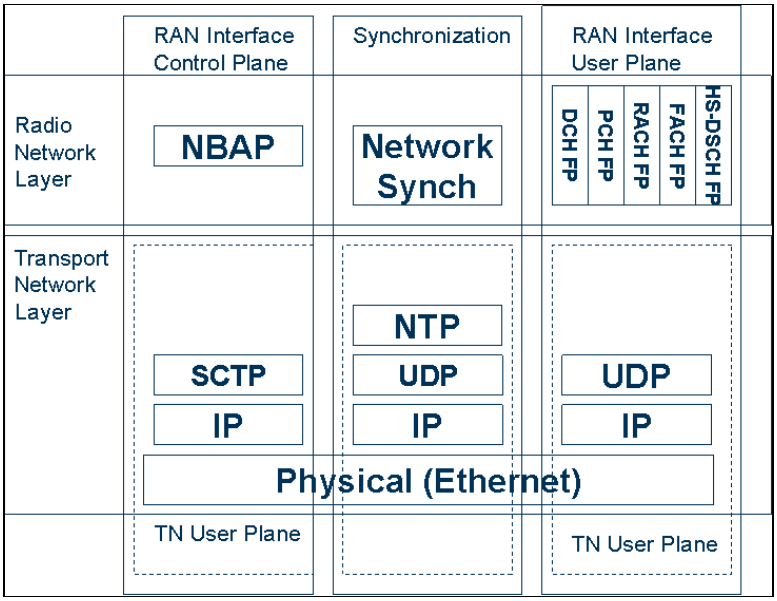


Figure 1-10: Iub over IP protocol stack

1.5 Control Plane:

The Radio Network protocol for the control plane is called Node B Application Protocol (NBAP) and transports Radio Network control plane messages between the RBS and the Controlling RNC (CRNC). The NBAP is carried on a signaling bearer based on SAAL-UNI on top of ATM. The control plane includes functionality for establishing AAL2 connections using Q.2630 and Q.2150.1. When using IP transport, NBAP is carried over SCTP.

1.6 User Plane:

The RNC and the RBS support a user plane over the Iub interface that is based on transport channels. The Radio Network User plane uses several frame protocols: FACH FP, RACH FP, PCH FP, DCH FP and HS-DSCH FP.

The user plane carries data and control frames that are transferred between the RBS and the CRNC, containing radio interface (Uu) user data and user associated control data.

1.7 Synchronization Plane:

Frame synchronization applies for the Iub interface. The Frame Synchronization function provides timing and supervision of traffic frames between the RNC and RBSs, in both the uplink and downlink directions. This is achieved by managing frame offset values that can be set different for downlink during operation. The frame offset values can be predefined in the system, but can also be refined; voice frame offsets are normally lower than for packet data, to be able to fulfill the delay requirements. The Node Synchronization function is a base for the Frame synchronization function, which is also a procedure applied for the Iub. The Node Synchronization functionality results in the knowledge of timing relationship between the CRNC and each RBS.

1.7.1 Protocols explanation

There are different protocols to build both Transport Network User Plane protocol stack and Transport Network Control Plane protocol stack on the Iu, Iur, and Iub interfaces, as illustrated in the figures above. Here is a brief explanation of each protocol.

1.7.1.1 SCCP Protocol

The signaling Connection and Control Part (SCCP) is a protocol that provides additional functions to the Message Transfer Part level 3 (MTP3b and M3UA) and provides connectionless and connection-oriented network services, as well as address translation capabilities. The SCCP protocol provides the functionality to transport signaling messages between SCCP users and also through signaling Transfer Points (STPs) and/or Signaling Gateways (SGs).

The SCCP is used to carry RANAP messages in the Iu interface and RNSAP messages in the Iur interface. The messages associated with a specific UE use SCCP connection-oriented mode; all other messages use SCCP connectionless mode.

The SCCP uses for addressing, a Destination Point Code (DPC) and Subsystem Number (SSN), that allow direct routing by the SCCP and MTP3b or M3UA.

The SSN is used by a terminating node (for example, the RNC) to identify different applications, for example, RANAP or RNSAP.

In connectionless mode, all information required to route the data to its destination is contained in each data packet and is analyzed in each node. Thus, no logical connection is established between the two end nodes.

The connectionless mode is typically used to transfer small amounts of real-time-critical information between two users.

The connection-oriented service is a way of exchanging signaling information between two network service users by establishing a logical connection between them. In connection-oriented mode, the data transfer messages only contain local reference numbers to identify the connection and they follow the previously established path.

The implementation of this protocol is compliant with the ITU-T, ANSI and Japanese standards, respectively: ITU-T rec. Q.711 - Q.714, Q716, Q752 (07/96); ANSI rec. T.111 - T.112 (1996); TTC rec. JT-Q.711, Q.712 and Q.714 (04/97), TTC rec. JT-Q.714 (04/99), TTC rec. JT-Q.771 - JT-Q.774 (1997)

1.7.1.2 SCTP Protocol

The Stream Controlled Transmission Protocol (SCTP) protocol layer provides a connection-oriented protocol between SCTP end nodes by the means of SCTP associations. An SCTP endpoint is defined by the SCTP transport address which consists of one or more IP address and an SCTP port. An SCTP association provides a reliable connection between two SCTP end nodes.

SCTP provides the following:

- Acknowledged error-free non-duplicated transfer of data
- Detection of data corruption, loss of data and duplication of data
- Selective retransmission mechanism to correct loss or corruption of data

SCTP provides a general-purpose transport protocol for message-oriented applications such as signaling operating on top of a connectionless packet service such as IP. It was designed by the IETF SIGTRAN working group, which defines the SCTP standard in document RFC2960.

1.7.1.3 MTP3b Protocol

The Message Transfer Part level 3 (MTP3b) protocol is responsible for transmission of signal units, with information provided by user parts, from one signaling Point (SP) to another. In WCDMA RAN Q.2630 and SCCP protocols are identified as users of MTP3b.

The MTP3b allows transportation of signaling messages through SAAL-NNI links and extends the functionality offered by the SAAL-NNI to provide network layer functionality. MTP3b functions are responsible for distribution, discrimination, and routing of messages.

To provide reliable transport capability for the transfer of signaling messages between signaling points (SPs), the MTP keeps track of the events occurring in the signaling network. The MTP is able to perform the following functions:

- Ensure that incoming messages are either distributed to a user part in its own node or routed and forwarded to the next signaling point or signaling transfer point.
- Ensure that outgoing messages are directed to the appropriate signaling link.
- Control and manage all functions that are needed on each signaling link to provide reliable transfer.

The implementation of this protocol is compliant with the following ITU-T, ANSI, Japanese and ETSI standards, respectively: ITU-T rec. Q.2140, Q.2210 (07/96); ANSI T1.111-1996 ANSI T1.113-1992 ANSI T1.115-1990; TTC rec. JT-Q2210 (version 1, 04/96); ETSI ETS 200 008-1, (01/97) ETSI EN 301 004-1, (02/98)

1.7.1.4 M3UA Protocol

The MTP Level 3 User Adaptation (M3UA) Protocol layer provides the connectionless service to its user as MTP3b Protocol provides. Internally M3UA provides and uses a connection-oriented service to support MTP3 signaling over IP. In WCDMA RAN Q.2630 and SCCP protocols are identified as users of M3UA.

1.7.1.5 SAAL-NNI

The signaling ATM Adaptation Layer-Network Node Interface (SAAL-NNI) offers link layer functionality ensuring that the two end points of a signaling link can exchange signaling messages reliably. It incorporates error checking, flow control, and sequence checking. SAAL-NNI is used as signaling link in both Iu and Iur interfaces.

SAAL-NNI includes Service Specific Connection Oriented Protocol (SSCOP), Service Specific Coordination Function-Network Node Interface (SSCF-NNI), and AAL5 layers.

- SSCF-NNI provides the special requirements at layer 2 for matching to the network-network interface.
- SSCOP provides functions such as: transfer of user data with sequence integrity, error correction by selective retransmission, flow and connection control, error reporting to layer management, and connection maintenance in the prolonged absence of data transfer.

SAAL-NNI is used to support the SS7 signaling protocol where SAAL-NNI links are used by the MTP3b network layer.

The following services are provided by the SAAL-NNI:

- Assured data transfer over point-to-point SAAL-NNI connections.

  Message delimitation, and alignment, error detection, and error correction are part of the assured data transfer service of this SAAL. The SAAL-NNI service relieves the user from loss, insertion, and corruption of data that may occur. In some cases, due to unrecoverable errors in the ATM adaptation layer, duplication or loss of service data units (SDUs) may occur.
- Ability to retrieve SDUs already delivered to the SAAL-NNI.
- One signaling link error monitoring function.

  It is employed when a link is in the proving state of the initial alignment procedure.
- Re-establishment of assured data transfer mode when a protocol error has been detected by the SAAL layer (recovery).

The implementation of this interface is compliant with the following ITU-T and ANSI protocol standards: ITU-T rec. Q2100, ANSI T1.652 (1996); SSCOP ITU-T rec. Q2110, ANSI T1.637; SSCF-NNI ITU-T rec. Q2140, ANSI T1.645.

1.7.1.6 SAAL-UNI

Signaling ATM Adaptation Layer-User Network Interface (SAAL-UNI) is required rather than SAAL-NNI in the Iub interface because the RNC and RBS are not peer entities and because the RNC only needs NBAP connections with the RBSs that are directly under its control.

The SAAL-UNI offers link layer functionality ensuring that the two end points of a signaling link can exchange signaling messages reliably. It incorporates capabilities, such as error checking, flow control, and sequence checking. The use of SAAL-UNI allows an efficient transport of the NBAP in the Iub interface as well as to support Q.2630 transport in Iub.

SAAL-UNI comprises Service-Specific Connection-Oriented Protocol (SSCOP), Service-Specific Coordination Function-User Network Interface (SSCF-UNI) and AAL5 layers. The following services are provided by the SAAL-UNI:

- Unassured data transfer over point-to-point SAAL-UNI connections. The SAAL-UNI unassured data transfer service does not relieve the user from loss or insertion of data which may occur.
- Assured data transfer over point-to-point SAAL-UNI connections. The SAAL-UNI service relieves the user from loss, insertion, and corruption of data which may occur. In some cases, due to unrecoverable errors in the ATM adaptation layer, duplication or loss of service data units (SDUs) may occur.
- Establishment and release of assured data transfer mode on an established SAAL-UNI connection.
- Reestablishment of assured data transfer mode on request from the user (re-synchronization).
- Reestablishment of assured data transfer mode when a protocol error has been detected by the SAAL layer (recovery).

The implementation of this interface is compliant with the following ITU-T and ANSI protocol standards: ITU-T rec. Q.2100, ANSI T1.652 (1996); SSCOP ITU-T rec. Q.2110, ANSI T1.637; SSCF-UNI ITU-T rec. Q.2130, ANSI T1.638.

1.7.1.7 Q.2630 Protocol

This AAL type 2 signaling protocol supports the dynamic establishment and release of individual AAL type 2 point-to-point connections. Q.2630 relies on the MTP3b or M3UA (in Iu and Iur interfaces) or SAAL-UNI (in Iub interface) for transport of messages between the relevant nodes.

Q.2630 Capability set 1 has the following characteristics:

- On-demand establishment, maintenance, and release of end-to-end AAL2 connections over AAL2 network comprised AAL2 end-points and AAL2 switching points
- Support hop-by-hop routing
- Ability to control AAL2 connections on more than one underlying ATM VCC (ATM Virtual Channel Connection)

Q.2630, in the implementation Capability Set 2, extends the AAL type 2 signaling protocol Capability Set 1 to support the following additional capability:

- Selection of AAL2 path type

1.7.1.8 Q.2150.1 and Q.2150.2 Protocols

Q.2150.1, in Iur and Iu-cs interfaces, defines the transport of Q.2630 for MTP3b and provides the necessary reliability for the signaling transport.

Similarly, Q.2150.2 defines the transport of Q.2630 directly over SAAL-UNI in the Iub interface.

1.7.1.9 GTP-U

The GPRS Tunneling Protocol User plane (GTP-U) is used between the RNC and the Serving GPRS Support Node (SGSN) in the Core Network.

GTP-U is a tunneling protocol that is used to transport user packet data over the Iu-PS interface. The main function of GTP-U is to provide separation between packets that belong to different user data flows, so there is a GTP-U tunnel for each PDP context in the SGSN or each RAB in the RNC.

A GTP-U tunnel is defined between a pair of tunnel endpoints. A tunnel endpoint is identified by a Tunnel Endpoint Identifier (TEID), which is carried in the GTP-U header. For one tunnel, there is a different TEID for uplink and downlink traffic, see the figure below. These TEIDs (as well as IP addresses for RNC and Core Network) are exchanged over Iu-PS in RANAP signaling when the packet data RAB is established.

A GTP-U path is defined between two GTP layers in different IP addresses, if there is at least one GTP-U tunnel up between those two GTP layers. In other words, a GTP-U path may contain several GTP-U tunnels.

In the ATM case, typically one (but could be more) GTP-U path is carried over UDP/IP path and AAL5. The PVC is set up by the operator, which is it is not established dynamically. It is possible to have several PVCs over the Iu-ps interface. Several GTP-U tunnels can be carried in a single AAL5 PVC (Permanent Virtual Channel).

When IP transport is used, the GTP-U layer is unchanged and is carried over UDP/IP and Ethernet.

A UDP/IP path is a connection less path defined by two end points where an IP address and a UDP port number define an end point.

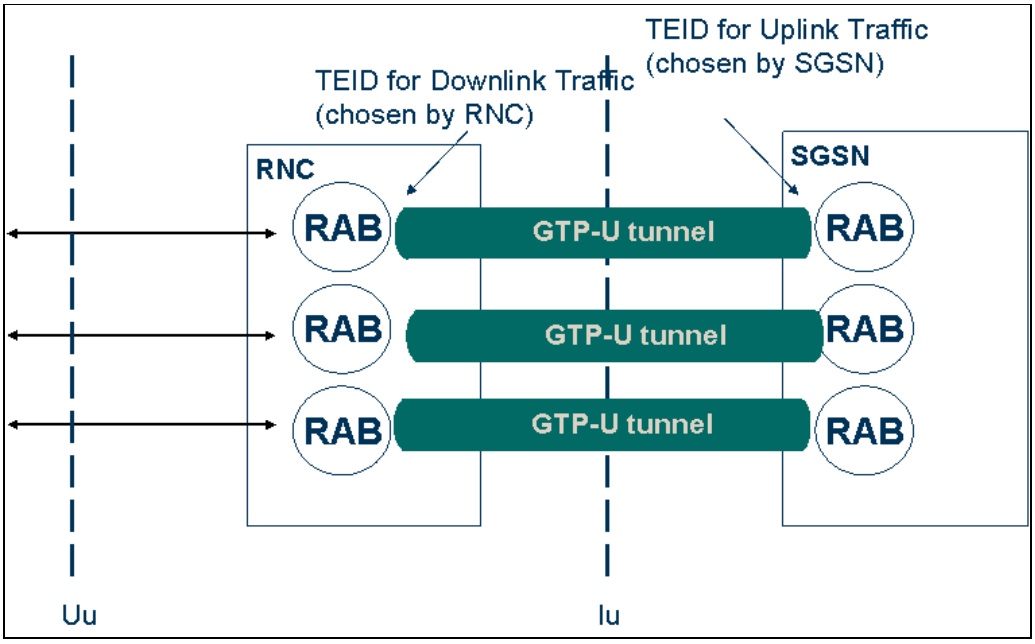


Figure 1-11: GTP-U Tunnels

2 RNC ARCHITECTURE

The RNC, as a radio network controller, handles the Iu interface towards the core network, the Iur interfaces towards the other RNCs for inter-RNS mobility, and the RNC-RBS subsystems via the RNC Modules.

The RNC is based on Connectivity Packet Platform (CPP). The platform consists of a transport system, a distributed real-time telecommunication control system, and a management support system. Cell switching is used for communication between boards and sub racks. Between nodes a range of different communication standards are supported.

For each connection between a UE and the UTRAN, an RNC can act as a Serving RNC (SRNC) or a Drift RNC (DRNC):

A single SRNC controls the radio connection between a UE and the UTRAN. The SRNC terminates the Iu interface for this UE.

A DRNC supports the SRNC with radio resources when the connection between the UTRAN and the UE need to use one or more cells controlled by this RNC.

The RNC owning the Signaling link towards an RBS is sometimes called Controlling RNC (CRNC).

The implementation of these functions is described in details below.

2.1 Layered Architecture

The RNC comprises a number of layers. A layer represents a hierarchical level offering services to the layer(s) above through a well-defined interface. Each layer consists of a number of subsystems.

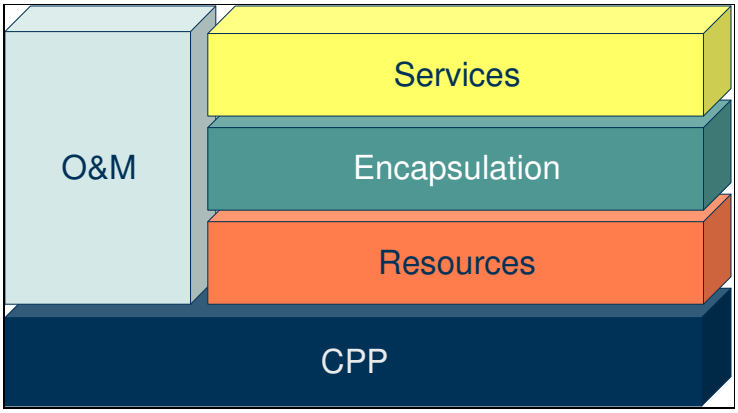


Figure 1-12: RNC Layered Architecture

2.2 CPP platform layer

The CPP platform layer provides basic support to the other layers, for example an operating system, necessary internal communication mechanisms, mechanics and power. The following parts are included: RLIB, TAS, MPE and CPP. It contains both software and hardware.

2.2.1 RNC Component Library (RLIB)

RLIB contains software only. Its main functions are as follows:

Component Library:

Packages used by other RNC subsystems containing common procedures, data classes and constants.

Examples of procedures provided by RLIB are: common restart functionality and MP-SP signaling support.

Application proxy: enables signaling between different OSE processes.

Load Control: secures real time characteristics and avoids restarts due to overload situations.

2.3 Debug Support.

2.3.1 Timing and Synchronization (TAS)

TAS contains software only. It main functions are as follows:

Distribution of timing information:

This information is used for node synchronization by other subsystems in the RNC.

2.3.2 Mechanics, Power and Environment (MPE)

MPE contains both hardware and software and its main functions are as follows:

Building Practice:

- Cabinet and sub-rack mechanics
- Backplane
- Cables
- Fan External Processor (XP)
- Interface Connection Field (ICF), which is used for connecting external cables.

Power:

- Capacitor Unit (CU), used for smoothing out irregularities in the power supply.

2.4 O&M layer

The Operation & Maintenance layer provides the overall O&M functionality for the RNC, including management interface services and access to the O&M support in the Platform layer.

2.5 Resource layer

The resource layer provides user and control plane resources administrated and controlled by the RNC. Examples of such resources are resources for Iu/Iub frame protocol handling, Uu L2 protocol handling. The following subsystems are included: DCS, CCS and PDR. It contains only software, no hardware.

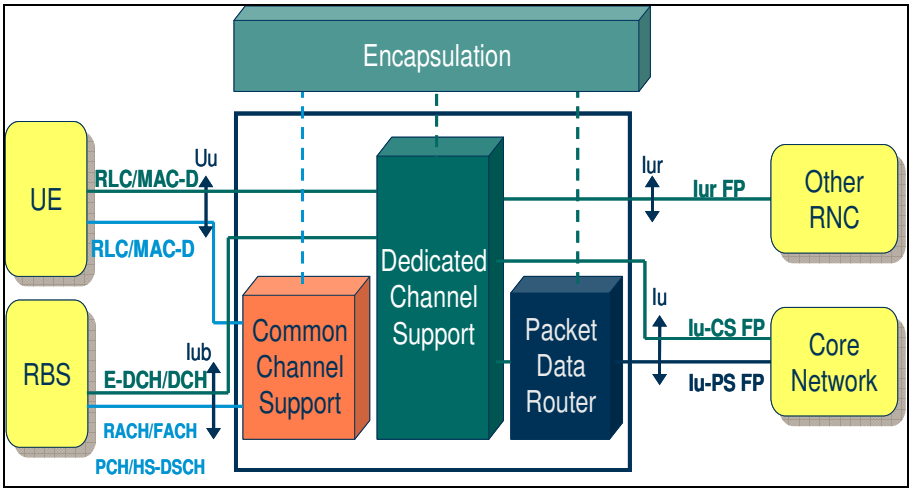


Figure 1-13: RNC Resource Layer

2.5.1 Dedicated Channel Support (DCS)

The main functions of this subsystem are as follows:

2.5.1.1 Iub Frame Transport for dedicated channel:

Transfers data and control frames between the RNC and the RBS over the Iub interface. In DCS, this includes frame handling of the DCH frame protocol.

2.5.1.2 Iu-c Frame Handling:

Transfer's user plane data between the circuit-switched CN and RNC over the Iu-c interface, including frame protocol termination.

2.5.1.3 MAC-C - MAC-D Frame Transport:

Transfers data and control frames between RNC nodes over the Iur interface. In the case of a single RNC node, this Iur protocol is used internally between DCS and CCS SP processors.

2.5.1.4 Uu L1 Termination of Dedicated Channels:

Uu L1Terminates the Uu L1 part handled by the RNC. This mainly includes the diversity handling of several legs in soft handover for dedicated channels.

In the uplink, the function also extracts Uu L1 measurement information that is needed by other RNC functions.

2.5.1.5 Uu L2 Termination:

Specifies the configuration and termination of the RLC and MAC-D protocols within the RNC. The protocols terminate the layer 2 signaling between the UE and the RNC.

RLC mainly handles segmentation, concatenation, buffering and ARQ.

MAC mainly handles scheduling; including choosing the transport format and flow handling between MAC-C and MAC-D.

The MAC-D and RLC protocols also handle ciphering when requested by the UE Security Handling function.

2.5.1.6 Uu L3 Termination:

Specifies the configuration and termination of the RRC protocol within the RNC. The protocol terminates the layer 3 control signaling between a UE and the RNC. Only RRC termination on SP is included in DCS. The rest of the RRC functionality is in UEH.

RNH and CCS handle the termination of the global part of RRC.

2.5.1.7 Frame synchronization:

Provides synchronization of uplink and downlink frames. In the downlink, in order to be transmitted on the air interface at a particular transmission time frames are synchronized.

The Connection Frame Number (CFN) and Transmission Time Instant (TTI) define the time at which the RNC has to send the frame to the RBS so that the frame is transmitted at the correct transmission time.

In the uplink, frames are synchronized towards the CN and also form a base for the macro diversity function.

2.5.1.8 Handover Evaluation:

This handles the handover evaluation for a number of handover functions. This is applicable for UE on dedicated transport channels and will provide the best possible continuous radio environment for the UE and the radio network.

2.5.1.9 Power Control, Dedicated Channels:

The uplink outer-loop includes calculating a quality target, which is the signal to interference ratio that is used by the inner-loop in the RBS.

2.5.1.10 Channel Switching Evaluation:

Monitor the traffic volume and buffer sizes. Based on these values the RNC can suggest changes in used transport channels

2.5.2 Common Channel Support (CCS)

The main functions of this subsystem are as follows:

2.5.2.1 Iub Frame Transport for common channel:

Transfers data and control frames between the RNC and the RBS over the Iub interface. In CCS, this includes frame handling of the FACH, RACH and PCH frame protocol.

2.5.2.2 Iur Frame Transport:

This function transfers data and control frames between RNC nodes over the Iur interface. In the case of a single RNC node, the Iur protocol is used internally between DCS and CCS SP processors.

2.5.2.3 MAC-C - MAC-D Frame Transport:

Transfers data and control frames between RNC nodes over the Iur interface. In the case of a single RNC node, the Iur protocol is used internally between DCS and CCS SP processors.

2.5.2.4 Uu L2 Termination:

Specifies the configuration and termination of the RLC and MAC-C protocols within the RNC. The protocols terminate the layer 2 signaling between the UE and the RNC.

RLC provides data segmentation/sequential sending from RNC to UE.

MAC-C maps the logical channels to transport channels. MAC-C schedules packets from the global part of RRC and QoS queues according to their priority, and selects suitable transport formats for each FACH from set transport format combinations. MAC-C also selects the transport formats for PCH.

2.5.2.5 Uu L3 Termination:

Specifies the configuration and termination of the global part of the RRC protocol within the RNC. The protocol terminates layer 3 control signaling (RRC) between the UE and the RNC.

Only RRC termination on SPB is included in the CCS. The rest of the RRC functions are in RNH.

UEH and DCS handle termination of other parts of RRC, excluding the global part.

2.5.2.6 Paging:

Specifies the configuration and data transport of UE paging within the RNC. Paging functions on SPB is included in CCS. The rest of the paging functions are in RNH.

2.5.2.7 Frame Synchronization:

This function provides synchronization of downlink frames between the RNC and the RBS. The frames are synchronized for transmission on the air interface at a certain transmission time.

The Connection Frame Number (CFN) and Transmission Time Interval (TTI) define the time at which the RNC needs to send the frame to the RBS.

2.5.2.8 Configuration Control:

The configuration data from RNH is sent through the control agent (in the RLIB subsystem on the MP) to the control agent in the CCS subsystem (on the SP) and then further to the SP processing entities (in the CCS).

2.6 Packet Data Router (PDR)

The main functions of this subsystem are as follows:

UDP/IP termination: This protocol is terminated in the RNC and in the packet-switched Core Network.

GTP-U termination: This protocol is terminated in the RNC and in the packet-switched Core Network. It acts as a multiplexing layer for user data packets that belong to different radio access bearers.

LLC/SNAP termination: This protocol is terminated in the RNC and in the packet-switched Core Network. It indicates the protocol that is carried in an LLC/SNAP frame. (Both IP and the Inverse ATM Address Resolution Protocol (InATMARP) are used on top of AAL5).

User data forwarding: In the downlink direction, this function forwards user data packets coming from GTP-U tunnels towards correct RLC connections.

In the uplink direction, this function forwards user data packets coming from RLC connections to the correct GTP-U tunnels to be transported towards the packet-switched Core Network.

2.6.1 Encapsulation layer

The encapsulation layer hides how the resources in the resource layer are implemented. The Device and Resource Handling (DRH) subsystem is included in the Encapsulation layer. This subsystem is used by the Service layer to reserve and release resources implemented in the resource layer.

The encapsulation layer contains only software, no hardware.

2.7 Service layer

The service layer provides the control services offered by the RNC, such as Radio Network Control, functions for paging of UEs, signalling connection handling and Radio Access Bearer service handling. The following subsystems are included: RNH and UEH. It contains only software, no hardware.

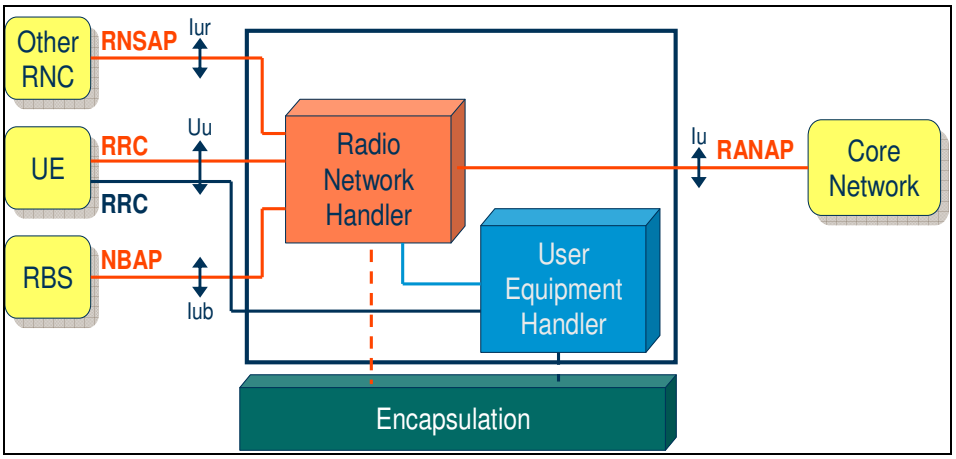


Figure 1-14: RNC Service Layer

2.7.1 Radio Network Handling (RNH)

The main functions of this subsystem are as follows:

Configuration management for logical radio resources:

Enables the user of the Mur interface to configure the radio network areas (URA, RA and LA), cells, common channels and their relations and also to configure system resources (RNC ID, Core Network identities, scrambling code sets and so on).

The user can also configure and establish, lock, unlock and remove signaling bearers through the Mur interface. Locking and unlocking can be performed on logical radio network resources, such as cells and common channels.

Control and mobility functions on common channels:

These include cell update, paging and system information distribution.

Capacity management:

These include power control on common channels, admission control and congestion control.

2.7.1.1 Handling of signaling bearers towards the Core Network:

The following is provided for RBSs and other RNCs to carry control signaling specified by the NBAP, RANAP and RNSAP protocols:

- Supervision, error and redundancy handling message distribution of:

RANAP protocol

NBAP protocol

RRC protocol

- Message termination of:

  common part of the RANAP protocol

  common part of the NBAP protocol

  global part of the RRC protocol

  RNSAP protocol.

Enables traffic functions to allocate and de-allocate RNTI, uplink scrambling codes and downlink channelization codes.

2.7.2 User Equipment Handling (UEH)

The main functions of this subsystem are as follows:

Configuration management:

- Enables the user of the Mur interface to configure, for example, timer values that are to be used in RRC signaling and mapping from RABs to RBSs and vice versa. Provides termination and message distribution of the dedicated part of RRC protocol.
- Termination of the dedicated part of the NBAP and RANAP protocols.
- Keeps track of the RRC State of each UE and the resources allocated to each UE in the RNC.
- Handles the setup and release of a signaling connection from the Core Network to the UEs.
- The signaling connection consists of an RRC connection from the RNC towards the UE and an Iu control plane connection from the RNC towards the Core Network.
- Transparent Message Transfer.
- Handling the setup and release of radio access bearers from the Core Network to the UE.

2.7.2.1 Radio Connection Supervision:

Supervise control and user plane connections between the UE and the UTRAN.

- Handover evaluation.

- Soft/Softer handover execution and handover between RNCs are also supported.
- Inter-Radio Access Technology Handover.
- Inter-Radio Access Technology Cell Change.
- UE Security Handling.
- Channel Switching.
- UE positioning.
- Compressed Mode Control.
- Inter Frequency handover.

2.8 RNC 3820 Hardware Structure

The RNC 3820 product was introduced together with P7.1 software version. This is a high capacity radio network controller, with a node throughput capacity of 2 Gbit/s and a radio network control capacity of 768 RBSs and 2304 cells. All aspects of the node have been enhanced. The main changes include:

- A new high power/high capacity subrack that combines more sophisticated power dissipation, with more connectivity in the back plane (including IP connectivity for certain slots).
- New high capacity processing boards, including the GPB65 and the SPB3.
- New Ethernet switch board – the C-MXB. Has a throughput of 10 Gbps per port and with 12 of the ports connecting to the back plane and 8 in the front panel.
- A switch control board SCB-DF and soon the SCB-TF, that provides two/three power feedings (maximum 1600/2400W), and a communication link to C-MXB.
- A higher capacity Ethernet exchange terminal, ET-IPG, that can connect to the Ethernet in the back plane allowing it to act as an IP/AAL2u gateway for traffic from the C-MXB

2.8.1 RNC 3820 R1.1

The RNC 3820 R1.1 with the new GPB75 board was introduced in W11.0.

The following scenarios for introducing GPB75 while node is in service are:

Added in W11.1:

• GPB75 replacement of SPB

• mMP replacement to GPB75

• GPB75 expansion in empty slots

• c1/c2 replacement to GPB75

2.8.2 RNC 3820 R2

A new RNC node type, RNC 3820 R2, is released which is designed to give a higher node throughput. In order to achieve higher Iub throughput the maximum Number of supported SPBs for RBS 3820 R2 has been increased from 40 to 45.

SPB4 is introduced in RNC 3820 R2. The device and SPM concept will be redefined with the introduction of SPB4. There are three types of devices in the RNC: CC, DC and PDR. For legacy SPBs, only one device LM has been loaded on each SPM. Therefore each SPM has been regarded as a device of a certain device type. On SPB4 each SPM has a dual core with one device running on each.

With SCB-TF a new generation of SCB board is introduced. It is a pre-requisite for introducing the new SPB4 HW due to its higher power requirements.

Compared to the SCB-DF, the SCB-TF board has the following main features:

• Supports up to 2 400 W power feeding capability.

• Supervision of current limiting devices (CLD) which will remove the necessity for cable length restrictions.

• Additional alarm info for the power inputs.

2.9 Main features

RNC 3820 has a number of features that make it differ from its predecessor the RNC 3810.

Inter-Subrack Links: In the RNC3810, the ISL bandwidth can potentially be a limiting factor. This occurs as the maximum throughput per Switch Core Board is limited to 620Mbps, and per link to 310Mbps. This is not much of an issue in the RNC3820, as the node now has Ethernet Subrack Links.

Ethernet Subrack Links: As mentioned earlier are new in the RNC 3820. Traffic destined for the node will enter the node in a preferred subrack. IP traffic is then transported to the correct subrack via these new ESLs, which emanate from the new C-MXB. All IP data remains in IP format until it reaches the correct subrack.

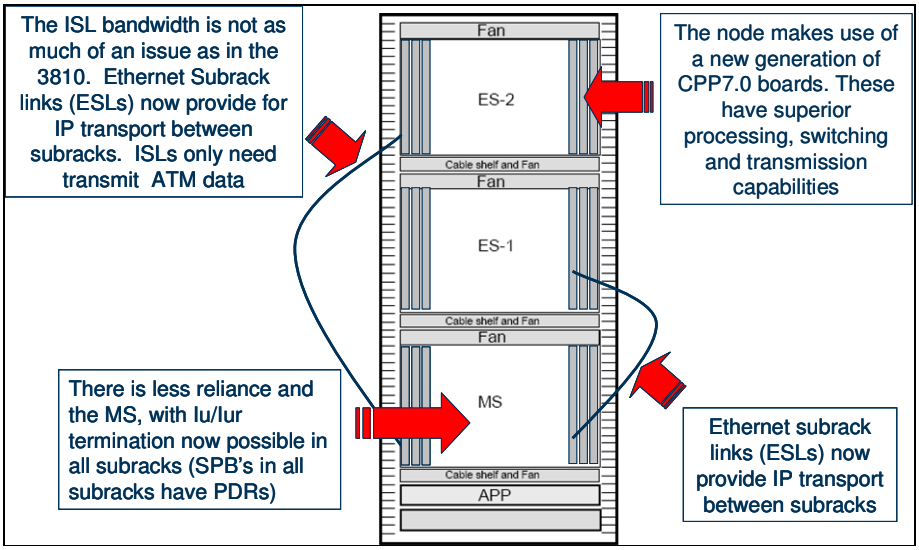


Figure 1-15: RNC 3820

The latest CPP release is used, leading to a more sophisticated set of plug-in-units to carry out the various functions. The processing boards specifically have a much higher processing power – contributing to the higher throughput in the RNC 3820.

MS Reliance: As mentioned above there is less of a reliance to use the Main Subrack, as termination of IuPS traffic is now possible on all subracks. This occurs as the new SPB3 boards which are used allow for PDR processors in all subracks.

2.10 High Capacity Subrack

The High Power Subrack is an enabler for CPP based nodes, such as the RNC3820, to fully utilize the increased capacity of the boards, such as GPB65 and the SPB3, as well as different ET boards.

- New self-contained high powered subrack, developed for use with the RNC 3820 Node. Allows for increased node throughput and subrack capacity
- Superior cooling facilities offset the additional heat generated by the more powerful processors in boards
- Will support 1600/2400W power dissipation, with easy upgrade for SCB-TF (which requires 2400W dissipation)
- IP backplane connectivity to the new C-MXB IP switching board is put in place for 12 slots
- ATM backplane connectivity for ATM boards to the SCB remains in place for the HCS

Figure 1-16: High Capacity Subrack

The subracks are based on the CPP platform, and are designed to handle high power dissipation, with eight pushing fans mounted at the bottom, blowing air into the subrack, allowing the cooling to 1600/2400 Watts. The back plane of the subrack has connectivity for ATM and Ethernet between the boards

The new IP switching C-MXB board will be located in slots 3 and 26 of every subrack (in an active/redundant configuration) and with an Ethernet subrack link between each, for redundancy reasons. The C-MXB board has 12 10Gbps backplane connections, which connect through the new high capacity subrack, from slot 3 and 26 to 12 of the slots in the subrack.

As can be seen below these are slots 4,6,8,10,12,14,15,17,19,21,23 and 25.

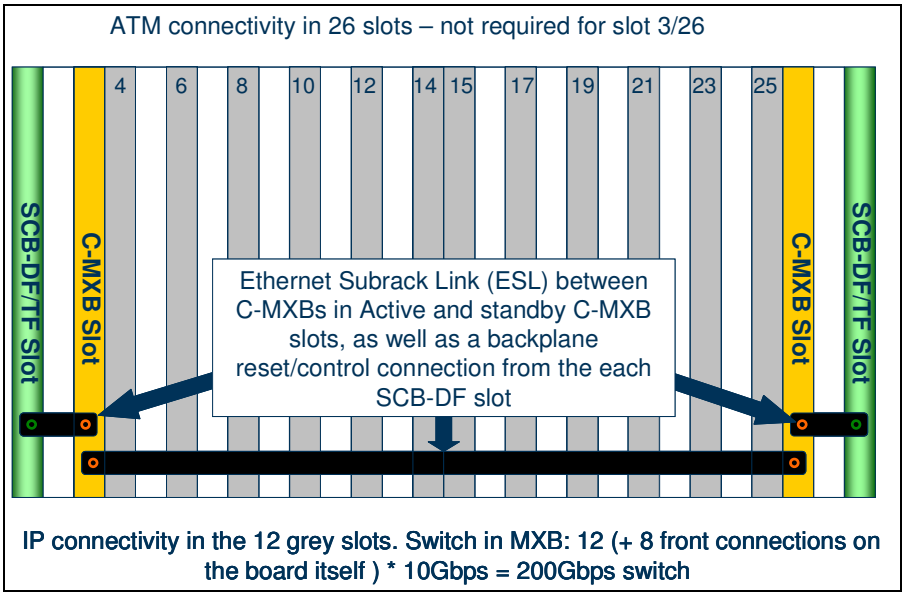


Figure 1-17: Subrack connectivity in the backplane

The implication of this is that we are no longer restricted to using ATM for internal transport. In fact we do continue to use ATM AAL2u for internal transport, as this is what the Processing boards are familiar with. We have the ability, however, to internally switch IP traffic that enters the nodes on the high capacity front ports of the C-MXB (8 of these in all), and send it to the new ET-IPG (IP gateway) board for conversion to ATM.

ATM connectivity remains in place in the subrack as previously, however it is not required in either slot 3 or slot 26, as these will always house a C-MXB. The C-MXB is purely an IP-based board without any ATM-conversion functionality, so will never require an input to the ATM switch.

The C-MXB acts as a 200Gbps switch. It can internally switch up to 120Gbps (12 * 10), but can also carry out external switching for traffic entering the node through its 8 front connections.

2.11 Subrack Configuration for RNC 3820

The subrack configuration in the RNC 3820 is not as stringent and set as in the 3810. In general portions of each subrack may be left empty with dummy boards, and then later utilised to increase the node capacity.

2.11.1 Main Subrack:

The minimum configuration for the main subrack in the RNC 3820 is given below. This is the lowest configuration at which the node can operate, and consists of a combination 17 plug-in-units:

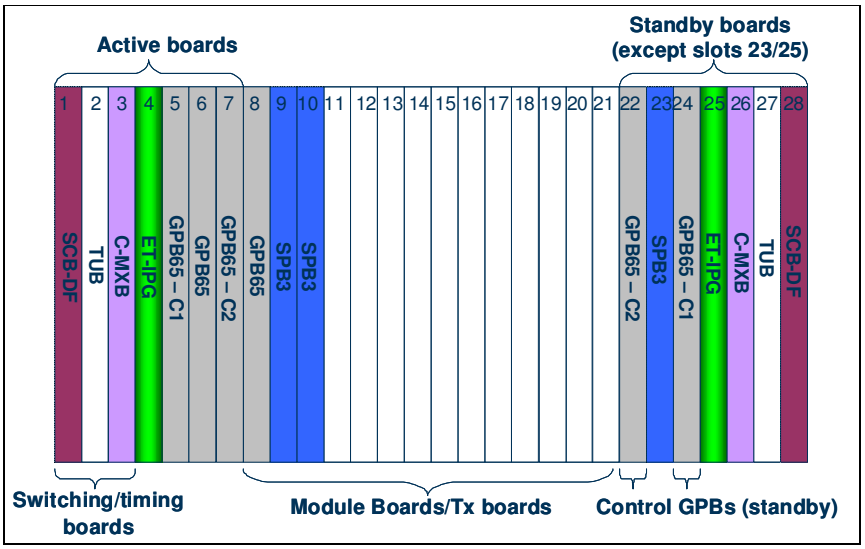


Figure 1-18: Main Subrack minimum configuration

- **SCB-DF/TF** is located as in the RNC 3810 in slots 1/28
- **TUB** is located in slots 2/27, in active/standby configuration
- **C-MXB** is always located in slots 3/26
- **ET-IPG** boards are located in slots 4 and 25, both active.
- **GPB-C1** provides combines the Central and O&M MP functionality
- **GPB-C2** provides the SCCP, RANAP and RNSAP termination.
- **Module Processors** are in place in slots 5 and 8 (GPBs) and slots 9, 10 and 23 (SPBs)

In **RNC 3810**, the number of GPB boards and SPB boards that comprise traffic processing modules is fixed for the main and extension subracks. These boards have fixed positions. The customer has free choice of number of ETs of each type in the free positions.

In **RNC 3820**, the free slots can be used for GPB, SPB and ET boards. The restrictions on number of traffic modules and number of SPB's per traffic module are removed.

2.11.2 Extension Subrack:

The extension subrack minimum configuration is shown below. As can be seen it consists of only ten boards:

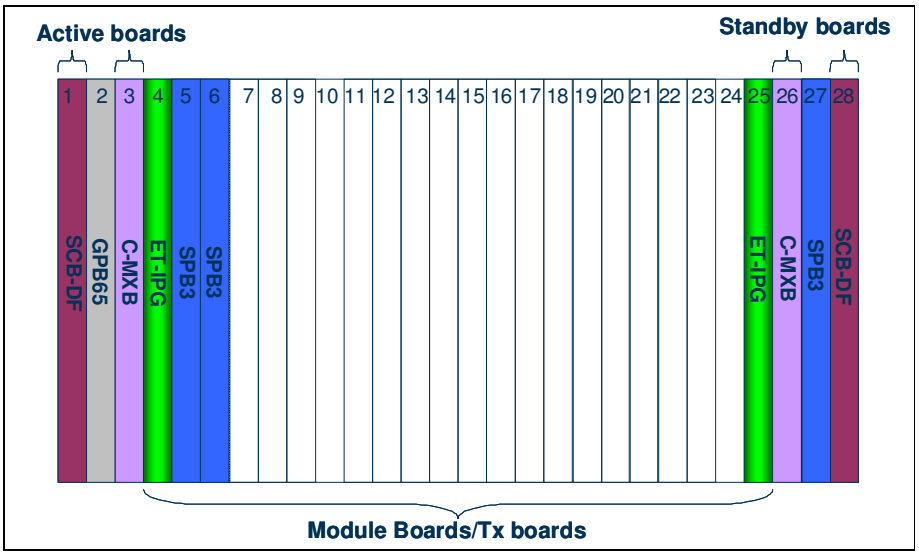


Figure 1-19: Extension Subrack min configuration

Here the SCB-DF/TFs are once again in slot 1/28, and the C-MXBs are once again in slot 3/26, and the ET-IPGs are once again in slots 4/25. There are processor boards in slots 2, 5, 6 and 27. Again the rest of the slots can be filled with processing or ET boards, based on expansion packages (also called C-packs and O-packs).

2.12 Capacity

- When migrating an 3820 from R1 to R1.1 the number of modules will be doubled, if the same number of mMPs are used after the migration. The max number of modules in R1.1 is 64 == 32 mMPs
- The capacity of an GPB75 compared to an GPB65 is at least a factor 1.7. This has been proved in lab by Ericsson.
- The capacity factor will anyhow rely upon the present traffic model which will load the processors in different ways.
- A side effect of an migration might be that the user plane load (DC-load) increases if the GPB previously was a bottleneck in the RNC, since more traffic will be accepted.
- The Dimensioning Description CPI considers the increased capacity from the GPB75 and shall be used when dimensioning an RNC.
- By using the R.1.1 with the possibilities of the flexible configurations, a good balance between user plane and control plane can be achieved.

Figure 20: Capacity, what to consider

2.13 Combining MP functionality:

In the RNC 3810 main subrack there was strict positioning rules used for the control MPs. These MPs are still positioned according to a rule in the RNC3820 main subrack, but there are now fewer of them due to the ability to combine functionalities in the new high capacity GPB65. This is shown below in Figure 2-26, as an example of the differences in slot allocation:

- Removal of the strict slot assignment for module boards, and removal of the requirement to have a certain number of GPB/SPBs per subrack
  - GPB C1 – combining the functionality of the Central MP and the O&M MP on one GPB
  - GPB C2 – combination of SCCP, RANAP and RNSAP signalling termination on one GPB (potentially high load here, but extra processing capability in GPB65 should keep it below 80%)
- Board functionality is combined in MP's to reduce slot usage:
- Higher processing power and more memory on GPB65/GPB75

Figure 1-21: Differences in Slot Allocation

GPB65 has 4GB flash storage, so even with the increased space required usage should not go above 75%.

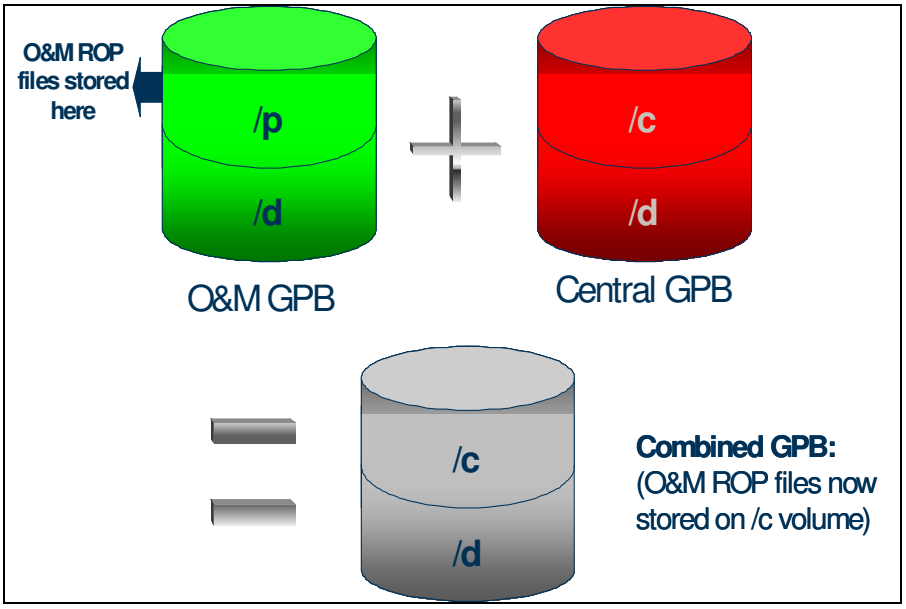


Figure 1-22: Disk Partitioning in GPB-C1

The /pxxyy00 drive is removed for the GPB-C1. Everything that was previously stored on the /pxxyy00 drive (such as the PM ROP files), is now stored on the C drive. The storage path for the CVs and load modules remains the same.

3 CPP History

MicroCPP came in as a part of the CPP7 project. This section briefly describes the evolution of the CPP platform. A lot of concepts (e.g. IP transport, IMA, NCLI, EM Toolbox, Node access security) are reused in MicroCPP.

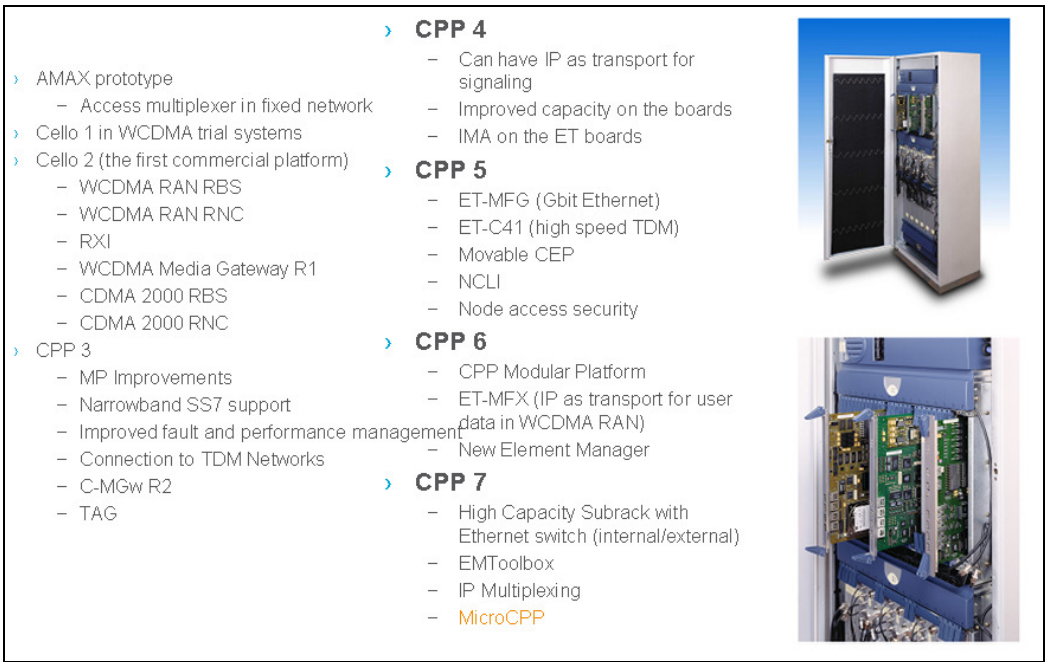


Figure 1-23: CPP History

3.1 Cello 1

The first implementation of a CPP based product was AMAX, an access multiplexer for the fixed network. This was a prototype for test purposes only.

3.2 Cello 2

The design of Cello 2 emphasized WCDMA products, for example the Radio Network Controller (RNC), the RXI, which is a RAN aggregator, and the Radio Base Station (RBS) in the WCDMA Radio Access Network (WRAN). Cello 2 was also the base for the Media Gateway (M-MGW R1) in the core network. However, these products had limited functions and further development was necessary.

3.3 CPP 3

CPP 3 brought more functions to the existing products. Some differences between Cello 2 and CPP 3 are:

- Main Processor (MP) capacity improvements with the GPB3.
- Support for Narrowband Signaling System Nr. 7 (SS7)
- Improved fault management and performance management functions.
- Ability to connect to traditional time-slot based TDM networks.

One product based on CPP 3 was the C-MGw R2. This release of the Media Gateway supported a split architecture where the MSC is only used for control functions and the C-MGw R2 contains functions for user data handling.

3.4 CPP 4

The differences between CPP 3 and 4 were:

- CPP 4 had support for IP as transport for signaling.
- CPP 4 included several objects to further improve the In-service performance on node level.
- There were some new boards in CPP 4. The ET boards were particularly important since they had support for IMA (Inverse Multiplexing for ATM)
- The new boards in CPP 4 provided large price/performance improvements to the application nodes, primarily by higher board capacity (SPB2, GPB4, MSB3), lower board cost (ET-MC1) and by better ability to use inexpensive broadband interfaces (ET-MC41).

3.5 CPP 5

The upgrade from CPP 4 to CPP 5 was mainly about new hardware. New ETB:s were introduced, mainly to improve backbone speed. One of the new boards, ET-MFG, provided IP over Ethernet transport for user data between MGws. Another board, ET-C41, was a high speed interface for TDM traffic. With that board it was possible to connect GSM BSC:s to for example a M-MGw.

Two other features that have been inherited by MicroCPP were released in CPP 5:

- Node Command Line Interface (NCLI). This is a way of accessing and change the configuration in the node through the ordinary Command Line Interface via e.g. a Telnet or Terminal session.
- Security solutions. Authentication and authorization were introduced together with secure transfer protocols for node access.

3.6 CPP 6

Some of the main feature enhancements in CPP 6 are listed below:

- Introduction of Modular Platform. Software and system functions are detachable and can be removed if not needed in a specific node.
- Support for user data transport over IP was introduced in WCDMA RAN. A new ET board (ET-MFX) was introduced as external interface on the Iub link. The board is using Ethernet as the bearer and it has a built in IP switch.
- Plug n' play integration of RBS over an IP/Ethernet link was introduced. This feature later became the "Auto integration of RBS" that is available for both WCDMA and LTE.
- New performance management counters are introduced along with a new "push" mechanism for data collection.

3.7 CPP 7

Some of the main feature enhancements in CPP 7 are listed below:

- New hardware in terms of High Capacity Subrack, SCB-DF, CMXB and ET-IPG. All these boards are used in the RNC3820 product.
- Introduction of MicroCPP which is the base for the RBS6000 product family. MicroCPP is built on new hardware and it has a built-in Ethernet switch that replaces the traditional "ATM" switch in CPP (AMAX switch).
- Restructure of EMAS to EMToolbox. More modular and flexible troubleshooting toolbox.

3.7.1 CPP – A modular platform

A CPP node consists of two parts, an application part and a platform part. The application part handles the software and hardware that is application specific (for example RBS, RNC, MGw etc). The platform part handles common functions that all nodes need. This can be for example internal communication, supervision, synchronization, processor structure and Operation and Maintenance (OaM) functionality.

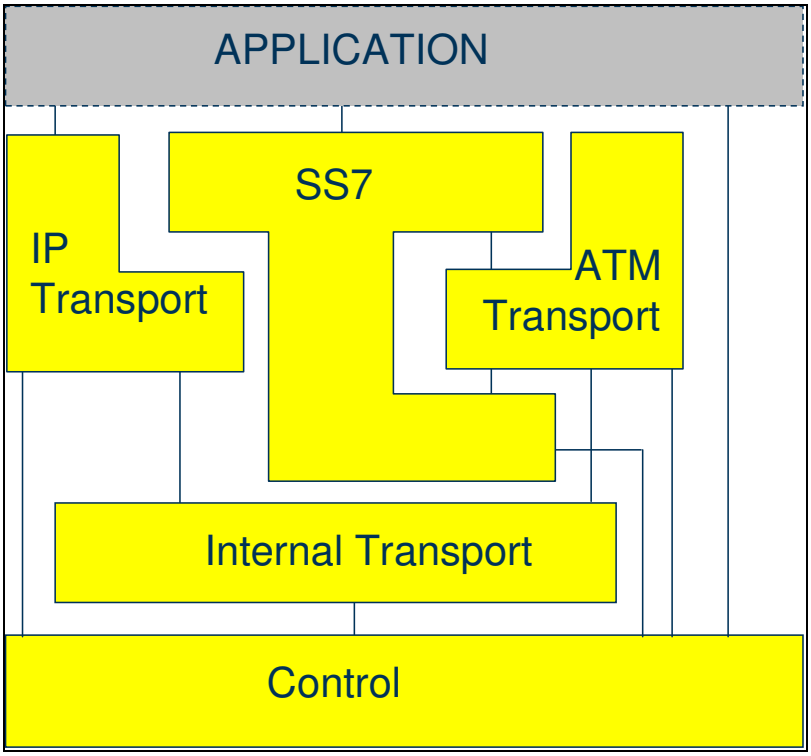


Figure 1-24: CPP Modular Platform

The CPP platform as shown in the figure below is made up of System Modules, which provide different services. The System Modules in CPP are: Control, Internal Transport, IP (Internet Protocol) Transport, SS7 and ATM (Asynchronous Transfer Mode) Transport.

A node is typically built up with several System Modules. Control, Internal Transport and ATM Transport are mandatory in traditional CPP. A node not needing the functionality supported by a module can exclude the whole module.

Below, the five System Modules are described.

3.7.2 CONTROL

The Control module contains the execution environment (e.g. processors, processor communication and file system) and it is mandatory for all CPP nodes. The System Module Control contains the following:

- High availability execution environment on MP (Main Processor) for C, Java and UML applications
- Hardware and Software for local execution environment on MP, BP (Board Processor) and SP (Special Purpose Processor, only used in RNC).
- Load and start of a CPP node
- Support for soft upgrade of SW and HW
- Transport for node internal control plane
- Basic operation and management of the node and support for application management

About 80-90% of the software is the same in CPP and MicroCPP. Also most system functions, OaM interfaces and the file system is reused.

3.7.3 ATM TRANSPORT

The ATM Transport module includes functions for setting up node internal and external ATM connections. The module covers SDH and PDH functions and interfaces, TDM switching functions and ATM signaling support. This module is mandatory in traditional CPP nodes. Only the WCDMA version of the RBS6000 uses this module. ATM is not described in this course.

This module is detachable.

3.7.4 INTERNAL TRANSPORT

This module is needed for traditional CPP nodes where an AMAX (ATM based) switch is used in the backplane. MicroCPP has a new system function for Internal Transport. The traditional ATM switch has been changed to a new Ethernet switch which means new hardware, software, functions and concepts have been included in the scope.

The Internal Transport module also includes network synchronization functions.

3.7.5 IP TRANSPORT

The IP Transport module includes IP user plane functions for external transport, SCTP (Stream Control Transmission Protocol) for signaling support and gigabit Ethernet interfaces. This module is mandatory in Micro CPP.

3.7.6 SS7

The SS7 module provides the service for sending signaling messages between the nodes in a WCDMA network, and it is hence only used in the RBS6000 products targeted for WCDMA.

This module is detachable.

4 MICRO CPP

Micro CPP is the platform that is used as a base for the RBS6000 family. The platform consists of Main Processor (MP) and Board Processor (BP) functionality, an Ethernet backplane switch, Network processors and front connectors to support termination of both ATM and IP/Ethernet traffic.

Micro CPP also supports IP Security (IPSec) termination on the Base Control Module (CBM) that holds all Micro CPP hardware. Two different CBMs exist. One is to support Long Term Evolution (LTE) base stations. The CBM is then placed on a circuit board called DUL (Digital Unit for LTE). The corresponding unit for WCDMA is called DUW (Digital Unit for WCDMA) and it has an own implementation of the CBM.

- ”*MicroCPP*” refers to the platform. In essence, MP, BP and network transport functions found in ”classic” CPP have been collapsed into a single HW module.
- CBM – CPP Basic Module
  - ET parts for IP and ATM transport functionality
  - MP/BP processor part for control/management –plane processing
  - Network synch
  - Ethernet switch used for internal DU transport
  - IPsec (10x ET-MFX board in CPP)
  - GTP-U forwarding
- Two variants
  - CBM for LTE – IP only
  - CBM for WCDMA – IP and ATM

Figure 1-25: MicroCPP

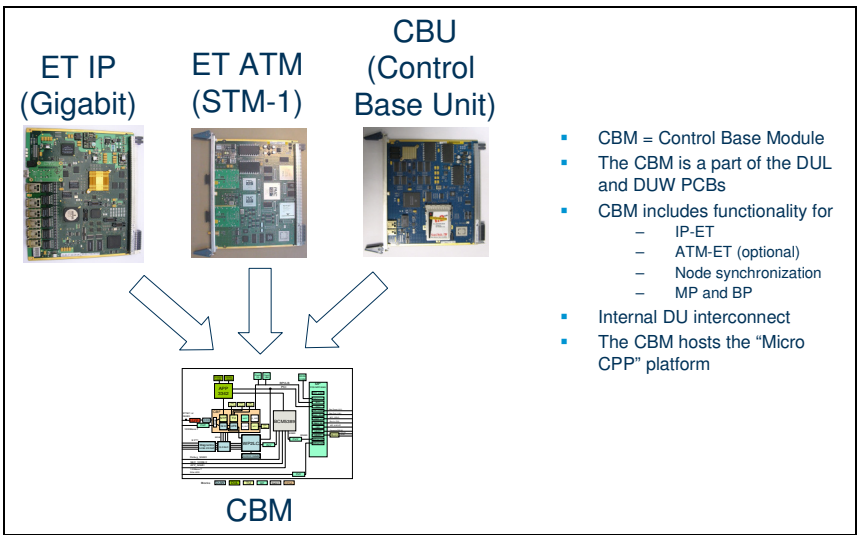


Figure 1-26: CBM ConceptRBS Architecture

The RBS is the implementation of the node denominated *Node B* in WCDMA system.

The Node B is a logical node responsible for radio transmission/reception in one or more cells to/from the UE. The Node B terminates the Iub interface towards the RNC.

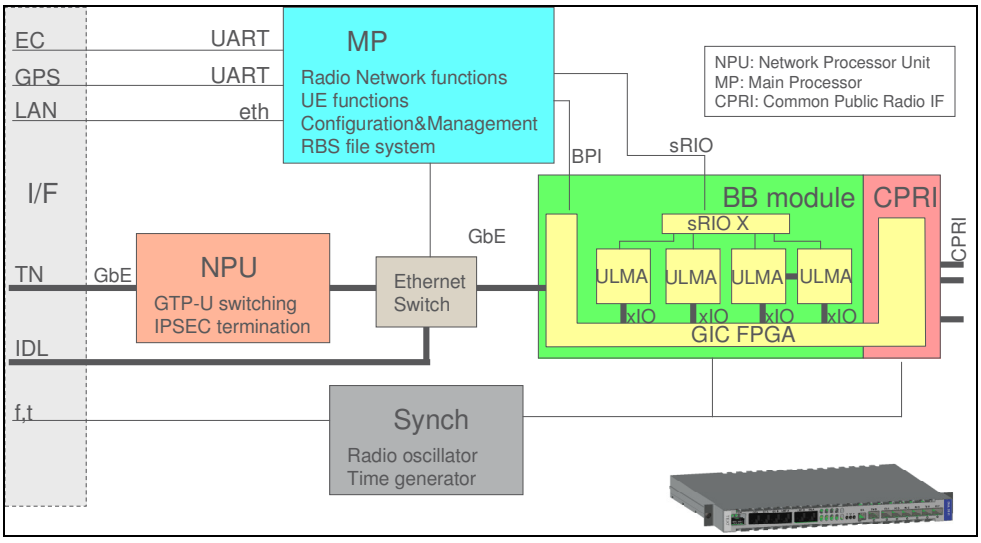


Figure 1-27: CBM + BB = DU

4.1 The General RBS Architecture is Divided into Te Following Functions:

4.1.1 Infrastructure and Platform Functions

The infrastructure and platform functionalities make all parts fit together. It contains the operating system, database, equipment control and internal communication. This function also controls all RBS external equipment like ASC, RET, like power supply, external alarms, climate unit etc.

4.1.2 User Plane functions

The User plane functions include transport, base band, radio and antenna near parts.

The transport functions terminate the user data coming from the RNC on the RBS. In the Baseband Unit all digital processing is performed, like channel coding/decoding, spreading, error protection etc.

Transport and Baseband functions are performed in the Baseband Subrack or in the Digital Subrack of the RBS.

The radio parts mainly care for modulating the data to the radio frequency and power amplification. These functions are performed in the Radio Frequency Subrack or in the Radio Unit Subrack of the RBS.

The antenna near parts function is responsible to process the radio frequency to the antennas. Filtering, combining, dividing etc. are to be done here. It is mainly performed in the Radio Frequency Subrack or in the Filter Unit Subrack of the RBS.

4.1.3 Control Plane functions

The Control plane functions are responsible for both traffic and O&M.

It terminates signaling protocols from the RNC and handles the traffic control functions. The O&M interfaces are also handled on the Control Plane.

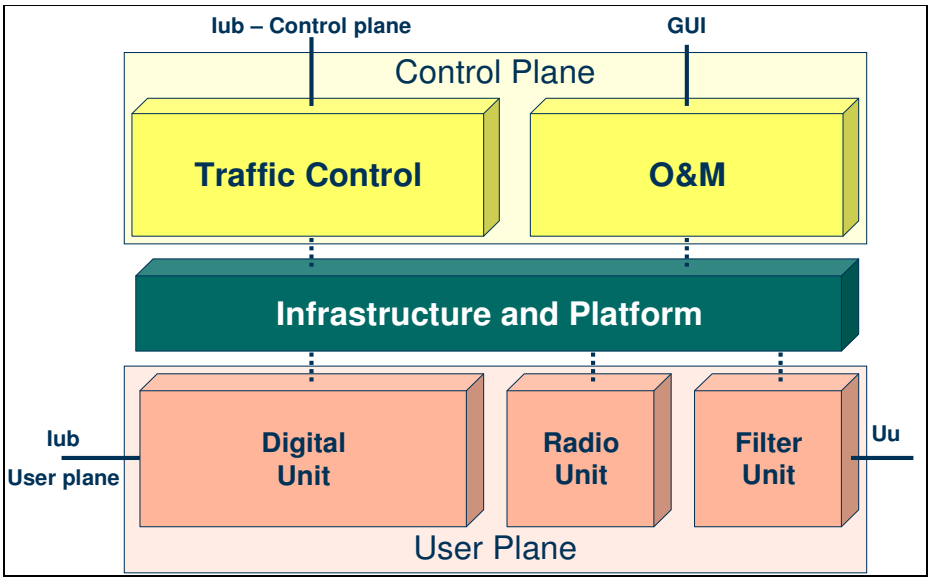


Figure 1-28: RBS Architecture

5 RBS 6000 HARDWARE LAYOUT

In the figures below, depicts the main functional blocks diagram matched to the hardware of a RBS 6000.

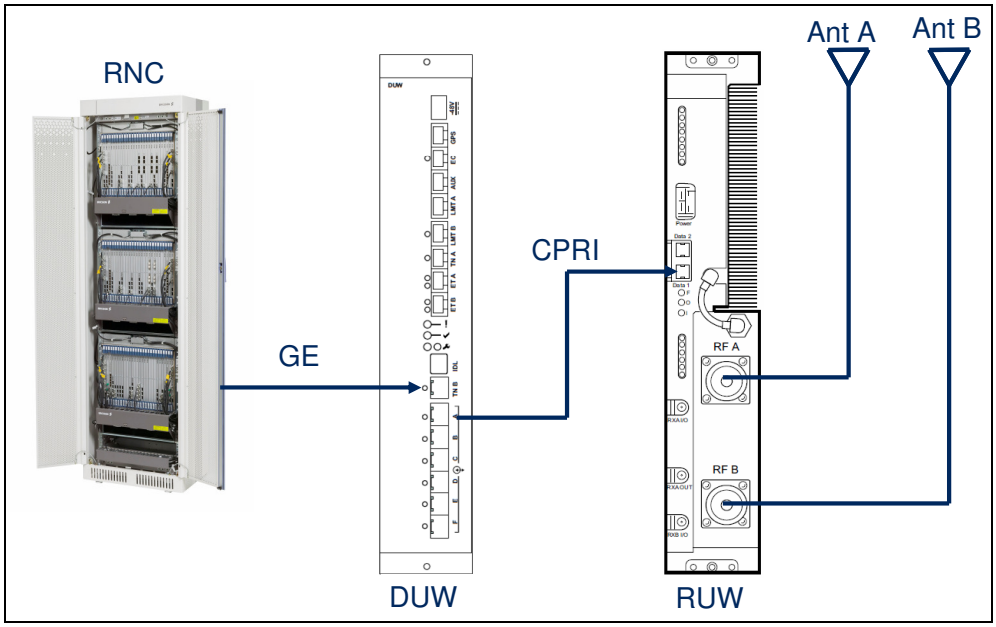


Figure 1-29: Board Functionality, RBS3206 Hardware

5.1 RBS 6000 Hardware

In the RBS 6000 series, there are two boards that combine the functionalities of the various boards in RBS3000.

***Radio Unit** Impletments:*

- Transceiver (TRX)
- Transmitter (TX) amplification
- Transmitter/Receiver (TX/RX) duplexing
- TX/RX filtering
- Voltage Standing Wave Ratio (VSWR) support

***Digital Unit** Implements:*

- Control processing and clock distribution
- Synchronization from transport i/f or GPS
- Baseband processing
- Transport network interface
- RU interconnects
- Site Local Area Network (LAN) and maintenance interface

Figure 1-30: Board Functionality, RBS 6000

The radio shelf in RBS 6000 base stations supports a wide variety of RUs and DUs for all main frequency bands, and any combination of Radio Frequency (RF) technologies (GSM, WCDMA, or LTE in later releases). Each radio shelf supports up to 6 Radio Units

6 CPP Node Redundancy Concepts

CPP nodes can be equipped with different kinds of redundancy to protect the node from quality degradation and faults.

6.1 Core Function Redundancy

Redundancy of the most important functions on the MPs is provided by the fault tolerant core (FTC). The purpose of this function is to provide redundancy in case of processor restart for some core functions. Fault tolerant core supports the redundancy of such basic applications as the system manager, the database, the global file volume /c, the load list provider, the configuration versions etc.

In case the core MP is restarting the fault tolerant core support will make sure that the node continues to operate.

This core software is located to two separate processor boards, which must be placed in the main subrack. One of the processors is in a specific moment acting as master and the other as slave.

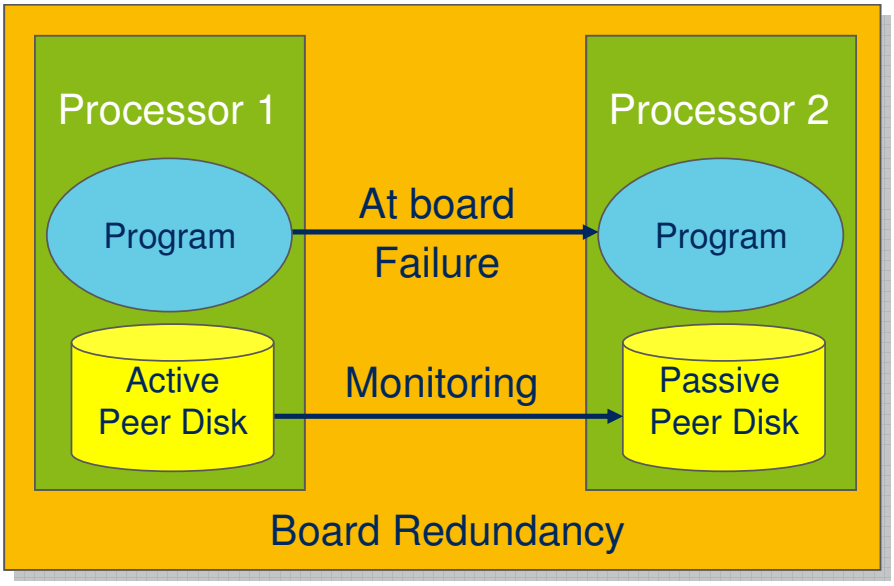


Figure 1-31: Core Function Redundancy

6.2 Reliable Program

Reliable program is a concept primarily aiming to protect against HW faults in either of two processors in the MP Cluster, but this functionality might also help in some cases for a SW fault. The data and program execution is running on a Master and Stand-by processor. At a fault or software upgrade a switch over to the other processor is executed.

The reliable program concept also includes support for storing of program state data. This support makes it possible to retrieve state data as it was before the processor or program crashed, or the program was activated for execution in another processor.

A Reliable Program Uniter (RPU) is a controlling and addressing entity providing one common addressing unit for two Reliable Programs, or more correctly, for parts of two Reliable Programs.

The Reliable Program support makes it also possible to configure a node for load sharing between programs (designed for load sharing) in different processors. This is achieved by distributing a number of RPUs among different MPs.

There are several possibilities to configure for redundancy.

- n+n redundancy.
- n+1 redundancy. One of the processors is made into a stand-by for all the others.
- 1+1 redundancy. A special case of n+1 redundancy where the application is distributed only on two processors.

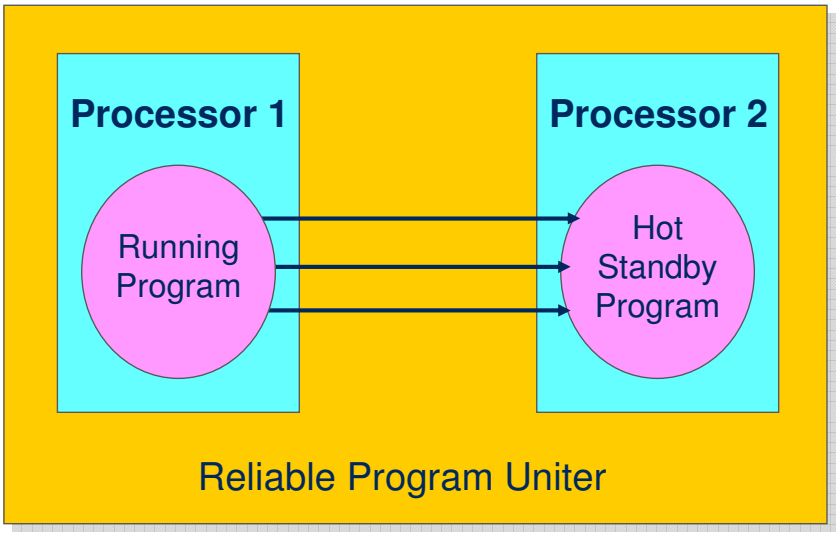


Figure 1-32: Program/Board Redundancy

6.3 Link Redundancy

Link redundancy provides redundancy for link failures. For this purpose, movable connection end points (CEP) are used to switch to a redundant board connected to a redundant link. If Multiplex Section Protection (MSP) is used, multiplex section protection groups (MSPG) are used for SDH Links.

Moveable Connection End Point. The processor in charge of the full UTRAN Signaling protocol stack can be allocated directly by the CPP node, thus reducing the configuration of redundancy at higher layers. This is especially used for Iub links, reserving only once the bandwidth needed for signaling protocols.

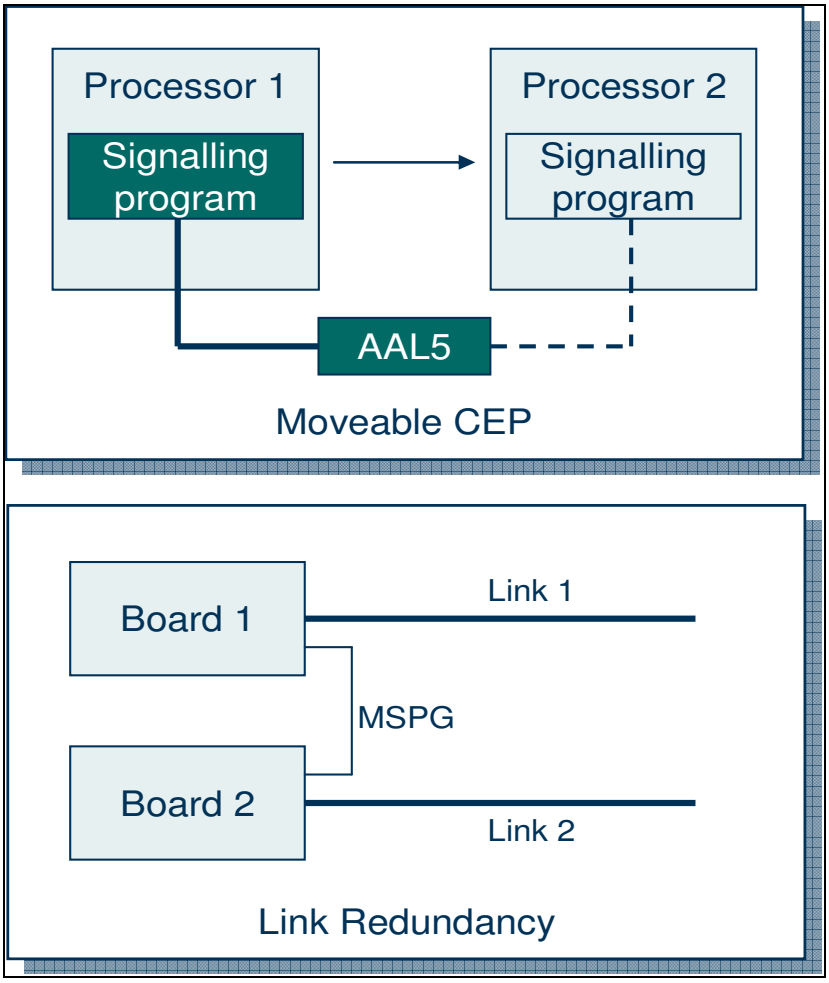


Figure 1-33: Moveable Connection End Point and Link Redundancy

7 MMI (MAN MACHINE INTERFACE)

Man Machine Interface (MMI) a maintenance mode button is introduced in RBS 6000 together with related support functions in WCDMA RAN. This allows the service technicians on site to lock/unlock boards without connecting a PC to the base station. Lock/unlock

Common MMI appearance for DUW and DUL is introduced to make it easier

for the operator to work with DUW and DUL HW appearance will also be the same regardless if there is a DUW, DUG or DUL.

The RBS 6000 will use LED INDICATOR colors as follows:

• Red color indicates a major fault. It is used to show that there is a HW fault that must be handled on that specific unit.

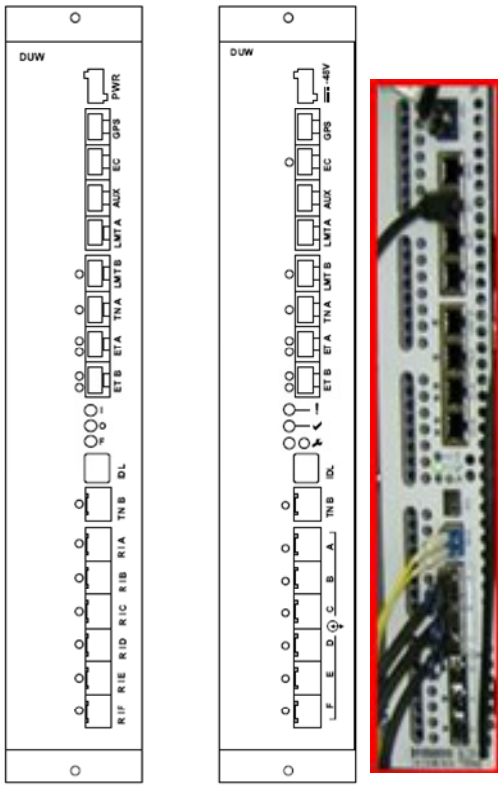
• Green color is used for Indicators that will be lit when the Node is working normally.

• Yellow color indicates a warning. It is used to show that something, it can be a unit or the Node, is not working correctly.

• Blue color is used to indicate that maintenance is ongoing. It is used on Indicators that the operator can influence via for instance the Maintenance Button.

A fully operational RBS shall not have any yellow or red Indicators lit, and there shall be no Indicators blinking.

The Figure 1-34 below illustrates the various DUW interfaces.



Marking	Connector	Description
PWR(1) -48 V(2)	ET20 A	-48 V DC Power
GPS	RJ-45	GPS External interface
EC	RJ-45	EC bus Internal interface
AUX	RJ-45	Auxiliary Bus Internal interface
LMT A	RJ-45	Console Internal interface
LMT B	RJ-45	Site LAN External interface
TN A	RJ-45	Transmission, Ethernet External interface, electrical
ET A	RJ-45	Transmission, 2xE1/T1/J1 External interface
ET B	RJ-45	Transmission, 2xE1/T1/J1 External interface
F (1) ! (2)	-	Fault Optical indicator, red
O (1) ✓ (2)	-	Operation Optical indicator, green
I (1) 🔧 (2)	-	Maintenance Optical indicator, yellow (1) Optical indicator, blue (2)
		Maintenance button (2) For information about the maintenance button, see Section 2.1.
IDL	HSIO	Inter-DU link Internal interface, DU to DU
TN B	SFP	Transmission, Ethernet/ATM External interface, optical
RI A - RI F (1) A - F (2)	6 x SFP	Radio Interface x 6 Internal interface between DU and RU, electrical External interface between DU and RRU, optical

(1) DUW Version 1
(2) DUW Version 2

Figure 1-34: Digital Unit WCDMA Interfaces

7.1 RAN Node File System

The File System allows the application to organize its data in sequential and/or hierarchically organized file system.

A partition manager allows a storage media to be divided into multiple partitions. A partition is a pre-allocated space on the media where a volume may be mounted.

The file system can run on flash disks or flash PROM memory.

The mirror device driver executes on both Core MPs (normal and standby), with both instances communicating with each other in an active/passive nature. The active mirror driver forwards all sector writings to the passive mirror driver, which performs the write on the slave disk.

Initially, the two disks are synchronized by the active mirror driver, which copies all sectors from the master to the slave.

For GPB 5 and lower, each MP includes a flash disk with a capacity of ~1.2 GB alternatively smaller. Two disk partitions are defined.

The BP based boards include a configurable (by component choice) flash PROM with a capacity of up to 64 MB.

Different volumes are mounted on each partition, depending on the configuration of the MP.

Global Volume /c is a fault-tolerant global volume, mounted on the active Core MP, mirrored on the passive Core MP, and useable by all processors. In case of a disk crash or MP hardware failure, the slave disk will become the master. The /c volume is intended for most applications using the disk. It is used for installation of load modules in the node, saving file based logs, html files, user documentation etc.

Local Volume /d on MPs is mainly intended for CPP internal caching of load modules and configuration data (database backup, armament file, etc.). This is done to shorten the restart time.

Local Volume /f on BPs is intended for caching of load modules. This is a volume implemented on the local system flash memory and the extended flash memory, if there is any. /f is intended for both application specific load modules for the DSPs, FPGAs etc. and OSE load modules for the BPs.

Global Volumes /pLNH are global volumes, mounted on non-Core MPs, and useable by all processors. The exact name of these volumes is /p followed by the link handler name. e.g. /p001600. The /pLNH volumes are intended for large disk space users. They are be used by CPP core.

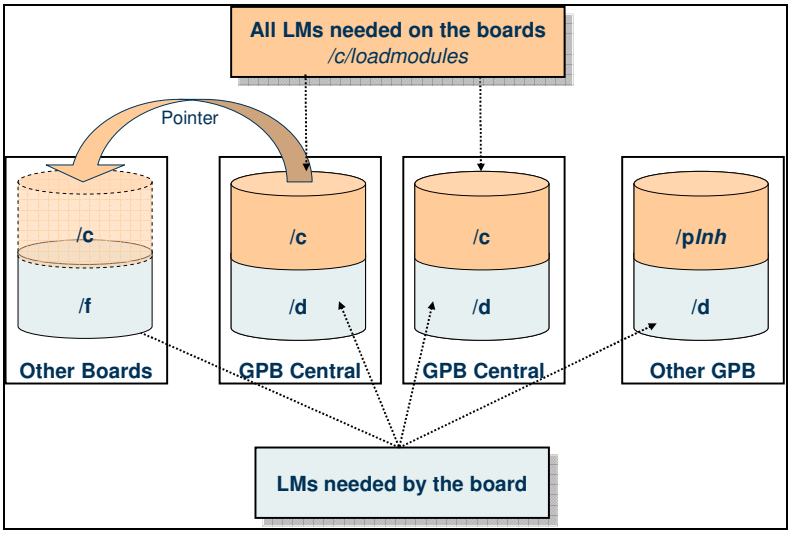


Figure 1-35: The RAN Node File System (RNC3810 focused)

7.2 Configuration Versions

A configuration backup of the node is called a Configuration Version (CV). All resources on the node, defined as Managed Objects (MO) in the MIB, are stored in this CV, and can be restored on the node when needed.

A number of CVs can be stored in the node, and provide as many backup of the whole node, available in case of any node failure or for rolling back to a previous configuration.

Also stored on a CV is a reference to the Upgrade Package to be used along with the CV. It is possible to have CVs referring to different UPs, which can prove useful during an upgrade process. An Upgrade package corresponds to a certain software version of the node.

Configuration Versions are stored under the directory **/d/configuration/cv/<cv name>**

The current Configuration Version is pointed out by the **cv.ptr** file in the **/d** directory. This file contains the name of the current CV directory.

A CV directory contains the following files:

- db.dat Database
- LLP.LMID Loader_server
- ok CV ok?
- Attribute Text information about the CV
- ARMAMENT Start-up file,
- Md5checksum CV checksum using MD5 technology

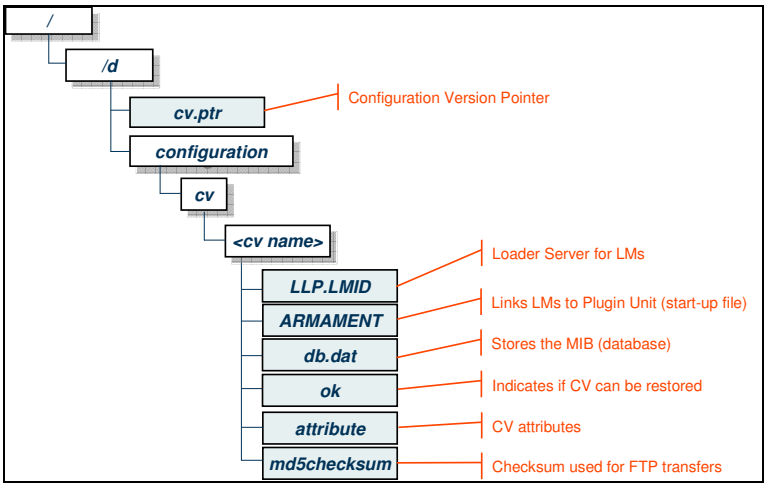


Figure 1-36: CV files in RAN Node File System

7.3 Node Restart Sequence

7.3.1 Cold Restart

The following describes the sequence in which files are used during the startup procedure, for a cold restart.

1. Read contents of **/d/cv.ptr** (e.g. <cv name>)
2. Move to directory **/d/configuration/cv/<cv name>**
3. Check **ok** file. If missing or damaged the node checks for next file in Rollback list and resumes from point 2. The ok file is mandatory and is used as a flag by the Configuration Version SW to indicate a successful creation of the CV. This file is empty when it is used for the first time.

4. Read contents of **LLP.LMID.** It has the name of the loader server Load Module. The loader server is the first Load Module that has to be loaded to the node. It loads the node configuration from the ARMAMENT file and then from the database into the node.

5. The **ARMAMENT** file loads the basic Load Modules to the node. It contains information about the Load Modules that need to be run before the database is in operation. It also indicates the boards that the Load Modules execute on.

6. The **db.dat** file is used to reconstruct the MIB with its Managed Objects when the ARMAMENT load modules have been loaded the node.

If system goes into a cyclic restart, or if the Element Manager will not launch, it is essential to monitor these files in the order in which they are used by the node.

7.3.2 Warm Restart

During a warm restart, only the MIB is loaded from the db.dat file. The following describes the sequence in which files are used during the startup procedure in this case.

1. Read contents of **/d/cv.ptr** (e.g. <cv name>)

2. Move to directory **/d/configuration/cv/<cv name>**

3. Check **ok** file. If missing or damaged the node checks for next file in Rollback list and resumes from point 2.

4. Use **db.dat** to reconstruct the MIB.

7.4 Rollback List

A Rollback List of CVs defines a priority list of CVs that should be used in case an error occurs with the startable CV at node restart.

The decision to move down on the Rollback List is dependent on the number of node restart that occurs in a certain period of time. Both the number of restarts and the period are configurable, using COLI command *cv*.

Another useful file held in the **/d** directory is **cv.bak**. This stores a reference to a CV other than the startable CV. This can be useful in handling some fault situations.

7.5 Load Modules and Programs

The code running on WCDMA RAN CPP hardware is stored in the form of Load Modules (LM).

A Load Module consists of the code and all the parameters and variables required in order to implement a specific task on the node.

An executing Load Module is called a program. A program is running on the Operating System Embedded (OSE) Delta environment. Each program contains one or more processes.

Processes communicate with each other using OSE signals, interfaces between processes are called an Actor.

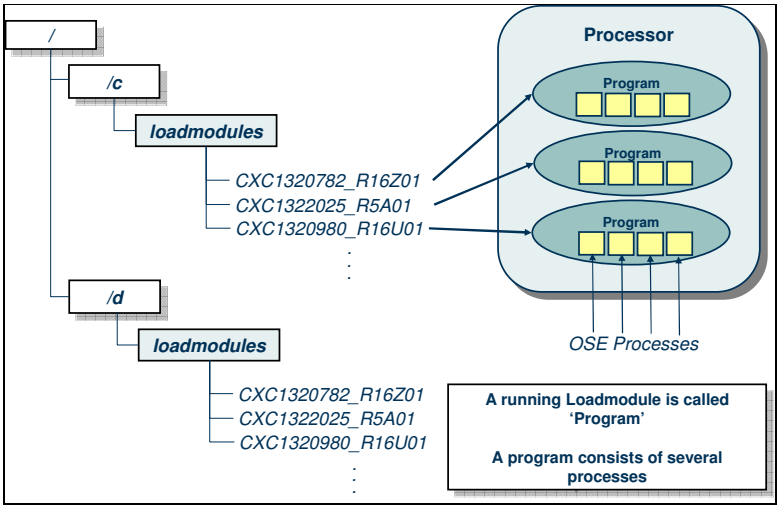


Figure 1-37: Load modules and Programs

In order to uniquely identify Load Modules, it has the following attributes:

- **Name**
  - Explicit name of the load module
  - Examples: ss7mgr
- **Product Number**
  - Identifies the Load Module type
  - Examples: CXC1323082
- **Revision**
  - Identifies the Load Module revision,
  - Examples: R12H01
- **Date of Creation**

The Product Number and the Revision form the file name of a Load Module on the Hard disk, e.g. CXC1323082_R12H01

7.6 Load Module Location

Each General Processor Board (GPB) in a RNC, RBS or RXI contains a hard disk that is divided into **/c** and **/d** partitions.

When downloading Load Modules into the node, for instance before an upgrade, all Load Modules are first downloaded to the Central GPB (slot 10 in RNC, slot 1 in RBS 3206, Slot 20 in an RBS 3202 and slot 2 in RXI). The node itself then handles the distribution of load Modules to all of the other boards.

The **/c** drive is mirrored by the rest of the GPBs inside the cluster, and is accessible by all the boards in the node. It contains all LMs to be loaded on the boards in the node.

The **/d** drive is only locally accessible on the GPB board, and contains the load modules required by this GPB. Load modules on the **/d** drive must be replicated in all the GPBs inside the cluster.

The **/f** drive is only locally accessible on all other boards than the GPB.

Note: The **/c** partition is called **/c2** in backup mode.

7.7 Software Allocations and Repertoires

Load Modules are grouped into **Repertoires**. Each Repertoire has a particular function, and regroups LMs that implement this particular functionality. Follows some examples of Repertoires:

- Repertoire **Cello_AAL2_RBS_MP** regroups the Load Modules needed for handling AAL2 switching in an RBS, and which have to be loaded on the Main Processor of the GPB board of an RBS.
- Repertoire **Cello_ETMC41** regroups all the LMs that need to be loaded on an ET-MC41 board.

A Software Allocation (SWA) is the linking of one or several repertoires to one or several slots on the CPP node. Following some example of SWAs for an RNC:

- SWA **GPB_Module** links:
  - Repertoires:
    - RNC_Module_MP,
    - Cello_Common_MP,
    - Cello_AAL5_MP,
    - Cello_AAL2_RNC_MP,
    - Cello_AAL2_Cen_Rh_MP,
  - To slots 14, 15 and 16 in the Main Sub rack, used by GPB boards handling the RNC modules of the Main Sub rack.

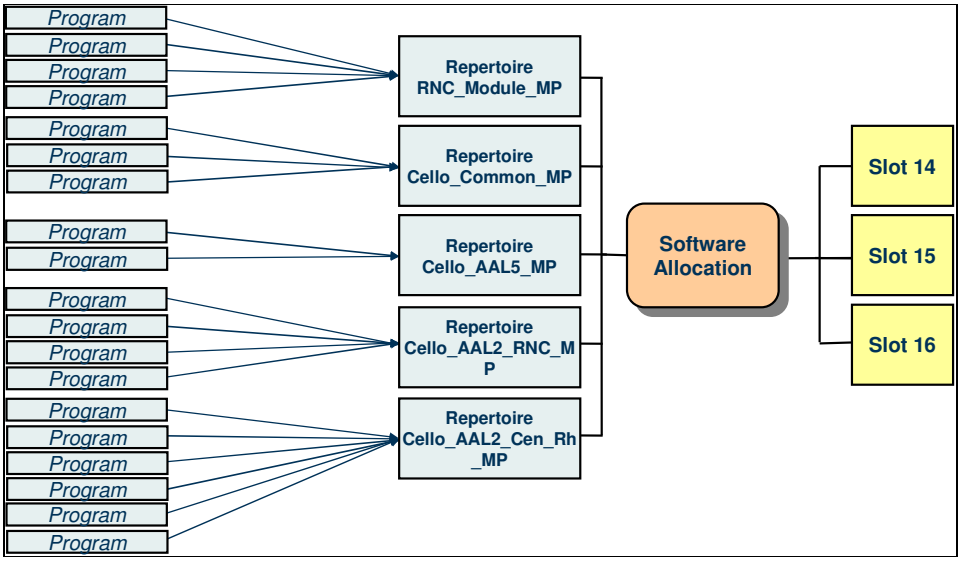


Figure 1-38: Program, Repertoire and Software Allocation

8 MANAGED OBJECT MODEL

In order to implement parameters over the physical and logical resources of the WCDMA RAN nodes, an abstraction of these resources is defined, based on Managed Objects (MO).

These MOs are then made accessible to external management applications via a Management Information Base (MIB).

Three layers are implemented to model the node's resources: Resource Layer, Management Adaptation Layer and Service Layer.

8.1 Resource Layer

The Resource Layer comprises physical and logical resources involved in carrying and processing the UTRAN traffic within a WCDMA RAN Node. This layer is structured according to the hardware and software components within the node.

The Resource Layer contains a number of internal, non operator-configurable resources. Consequently, only an abstraction of the Resource Layer is presented to the management applications

8.2 Management Adaptation Layer

The Management Adaptation Layer raises the abstraction level from the Resource Layer to a higher level suitable to be handled by management applications through the Service layer. Only resources that are needed for configuration and supervision are modeled.

The Management Adaptation Layer makes the configuration of logical resources possible, even if the underlying physical resource does not already exist on the node. This facilitates processes, and reduces resource downtime, during upgrades, reconfigurations and network extensions.

The Management Adaptation Layer presents the Service Layer a model of the node's resources based on Managed Objects. Each Managed Object is an instance of a Managed Object class defined in the Managed Object Model (MOM).

There is an RNC MOM, an RBS MOM and an RXI MOM, specific for each node's type and version.

8.3 Service Layer

8.4 Management Information Base (MIB)

In order to configure a WCDMA RAN Node, management applications need to get access to the Managed Objects. The Service Layer implements a Management Information Base (MIB) that centrally stores all Managed Object instances on the node, the relationships between Managed Object instances as well as it provides access to these Managed Object instances.

8.5 Configuration Service

The Service Layer provides a number of services for different purposes: Alarm Service, Product Inventory Service, Performance Monitoring Service and of course Configuration Service.

The Configuration Service ensures that configuration applications can configure MO instances in the MIB.

8.6 Access

The Service Layer provides external access to the MIB and MOs via the Common Object Request Broker Architecture (CORBA) protocol carried over the Internet work Inter-ORB Protocol (IIOP) protocol.

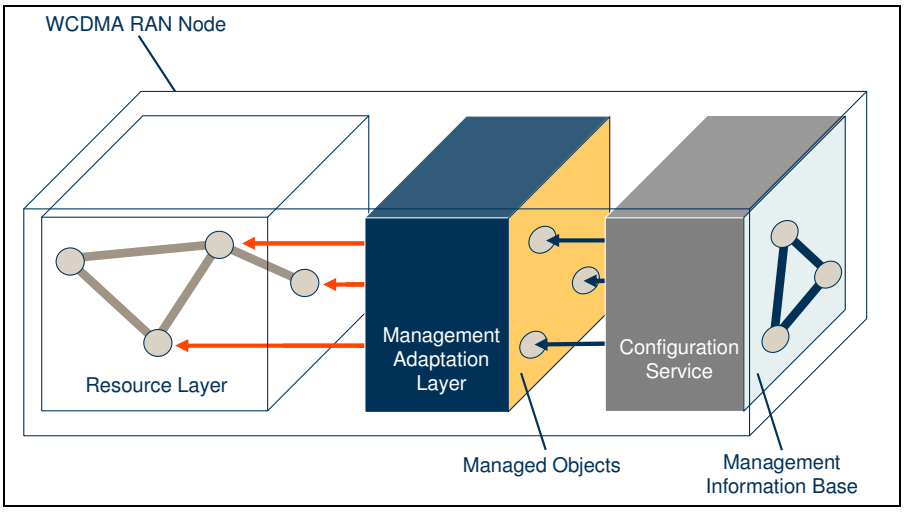


Figure 1-39: WCDMA RAN Node Resources Modeling

8.7 Managed Object Class

An MO represents a resource in the node, either a physical resource such as a plug-in unit or a fan, or a logical resource such as a software program or a protocol.

A number of MO classes exist to model all the resources needed for the node configuration and supervision. There are also parameters associated with the resource, which are called the MO attributes.

The MO can be configured by setting suitable values for the MO attributes. The state and configuration of the resource represented by the MO can be monitored by reading these attribute values.

There can also be MO actions related to the resource. For example, an MO representing an executable program might have an action called "restart".

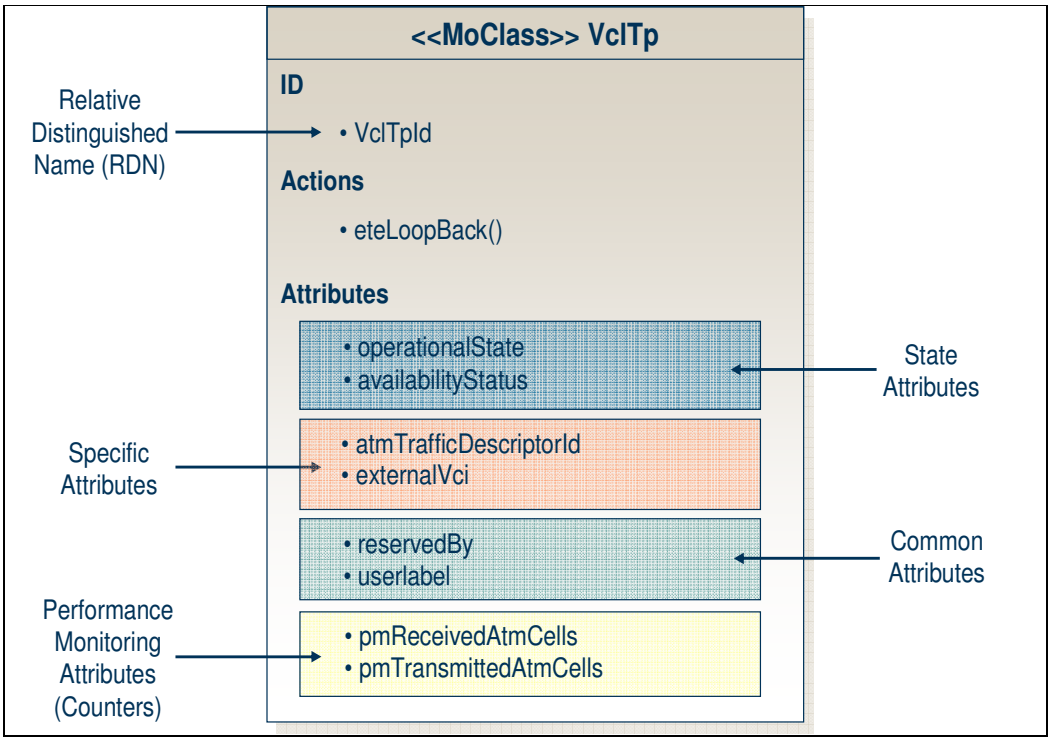


Figure 1-40: Managed Object (MO)

9 SUMMARY

After this chapter the participants should be able to:

1 Understand the system concepts, redundancy and configurations in CPP

1.1 Use COLI/AMOS commands to understand Fault Tolerant Core (FTC) concept and Reliable Program Uniter concept

1.2 Understand how link redundancies work in both ATM and IP based transport interfaces

1.3 Understand the concept of moveable Connection End Point (Mv CEP)

1.4 Understand the File system in a CPP based node

1.5 Be able to perform an emergency recovery of a CPP based node from a backup placed outside of the node

1.6 Describe the Subscriber Capacity RNC and RBS with New Hardware and Software.

1.7 Be able to interpret Managed Object attributes to explain how interfaces are configured from a CPP based node using Element Manager and AMOS

Figure 1-41: Summary of Chapter 1

Intentionally Blank

2 Troubleshooting Tools: Fault Management

Objectives

After this chapter the participants will be able to:

- 2 Use the applications in OSS-RC, Element Manager and COmmand Line Interface (COLI) that are important during a troubleshooting procedure.
- 2.1 Start and understand when to use the following applications in OSS-RC: Transport Network Viewer, Job Manager, Alarm List Viewer/ Alarm Status Matrix, WCDMA RAN Performance Measurements, Common Explorer GUI, Node Status Analyzer and Cabinet Equipment Viewer, Advanced Managed Object Scripting (AMOS) interface and Node Command Line Interface (NCLI).
- 2.2 Lock and restart boards and nodes including the soft/hard lock concepts.
- 2.3 Check the status of the Manage Object to find out the health of the node.
- 2.4 Understand when COLI is used and when Element Manager/NCLI are used.
- 2.5 Describe Enhanced Health Check Filter File.
- 2.6 Describe Product Inventory improvements.
- 2.7 Understand the WCDMA RAN load expert.
- 2.8 Understand Supervision of SP pool and 4 Way Receiver Diversity with DUW.
- 2.9 Understand changes in FM Events and Changes ROP

Figure 2-1: Objectives of Chapter 2

Intentionally Blank

1 OVERVIEW

Performance Data Analysis and Fault Management procedures are the two main components of the troubleshooting procedure.

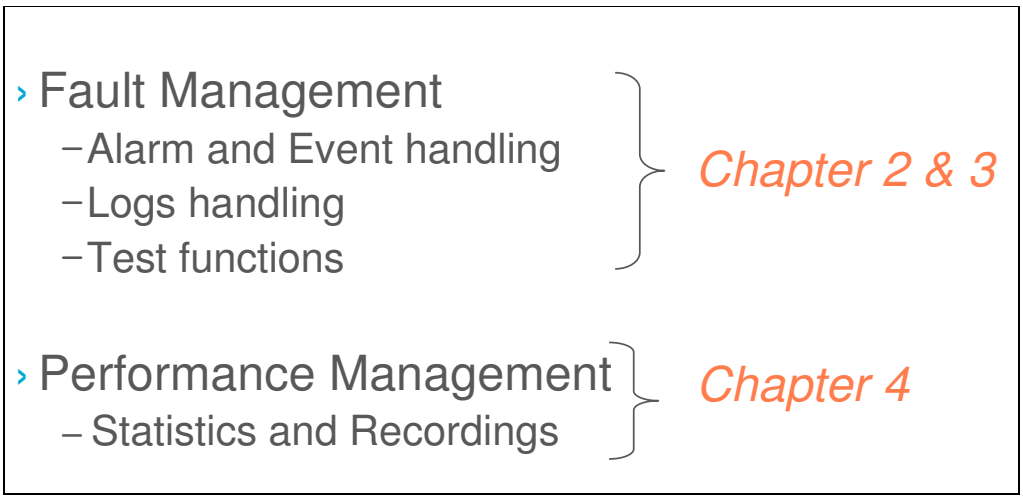


Figure 2-2: Tools used for troubleshooting

This chapter looks gives an introduction to the Fault Management procedures. Performance Management details are given in Chapter 4. User interfaces (applications in the OSS-RC and Element Manager) related to fault handling procedures also included. Having a good knowledge of all the applications could prove useful during troubleshooting.

2 FAULT MANAGEMENT

Fault Management ensures that WCDMA RAN operates correctly and informs the operator about the faults and actions needed to correct the faults in the RAN.

It does this by detecting and isolating faults, by forwarding alarm and FM event notifications to subscribers like OSS-RC and the NMS.

It also handles the operational state of the resources affected by the fault.

2.1 Fault Categories

Figure 2-3: Fault Categories

Each potential fault in WCDMA RAN is grouped into one of the following categories:

- **Hardware failures:** malfunction of a physical resource within an NE.
- **Software problems:** includes software bugs or database inconsistencies.
- **Functional faults:** failure in some functional resource in an NE, where no hardware component can be found responsible for the problem.
- **Overloading:** loss of some or all of the NEs specified capability due to overloading.
- **Communication faults:** communication failure, for example between two RNCs or between two operating systems.

2.2 Fault Management Functions

2.2.1 Fault Management Model

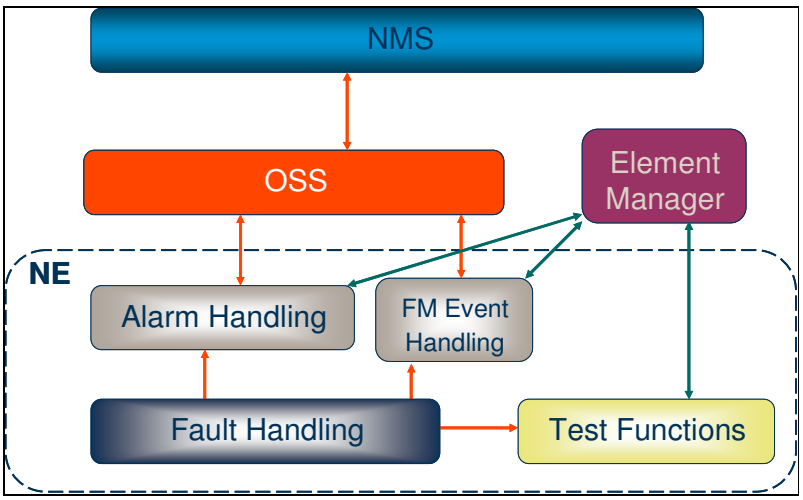


Figure 2-4: Fault Management Model

2.2.2 Fault Handling

The Fault Handling function represents the lowest level of fault management. It is performed close to the fault cause location. This function monitors the system for faults and sends out an alarm report if it detects a fault. The Fault Handling function also covers state handling.

2.2.3 Alarm Handling

This function handles any alarms the WCDMA RAN issues. It maintains a list of active alarms then distributes these and logs the alarm history.

2.2.4 FM Event Handling

This function handles Fault Management-related events and distributes the FM events and logs the FM event history. One example of an FM event is "RNC node restart completed".

2.2.5 Test Functions

In addition to the automatic supervision of functions for monitoring the NE's while in use, the operator can request manual tests to verify specific functions.

2.2.6 Fault Management Principles

The Fault Management functionality within the different Management System layers is shown in the figure below.

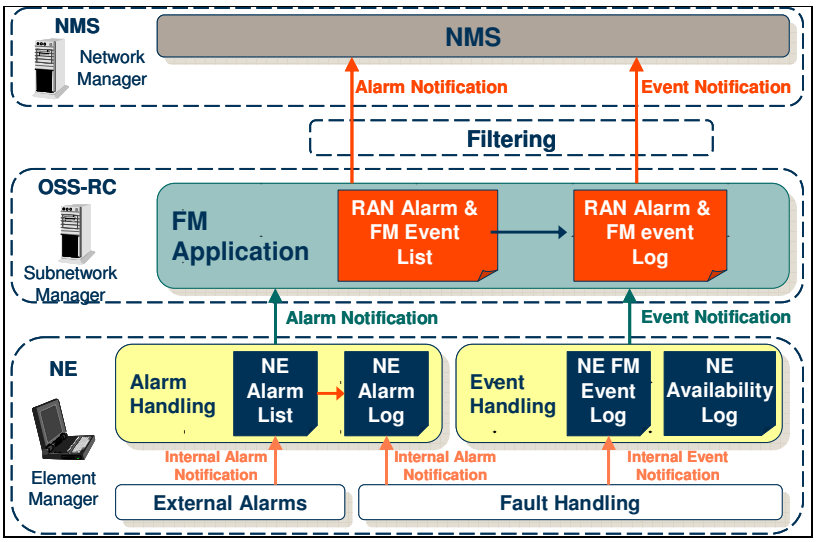


Figure 2-5: Fault Management Functionality

2.2.7 Network Management Layer

This layer is responsible for the management of the complete network, which can consist of multiple systems, both third generation (3G) and second generation (2G) mobile systems. The Network Management System (NMS) is responsible for handling Network Management functions common to several networks or functions, such as Fault Management, that impact one or more subnetworks. The NMS is not a part of the WCDMA RAN.

2.2.8 Subnetwork Management Layer

This layer mediates alarms and FM events from the NE's to the NMS and it also contains FM applications at RAN level.

The alarm and notification interface that OSS-RC provides to the NMS is the Management Interface, Mu.

OSS-RC provides means to view the status and properties of the RNCs, the RBSs, and the RANAGs in the WCDMA RAN. With the OSS-RC FM application the operator can view and administer the alarms, the FM events, and the alarm logs (including FM events). OSS-RC Fault Management applications includes as standard functionality such as Alarm List Viewer (ALV), Alarm Status Matrix (ASM) and Alarm Log Browser (ALB). It also offers the FM Expert System (FMX) which is an optional product.

The Sub network Manager applications Network Status Display (NSD) and Node Status Analyzer (NSA) are aimed for monitoring and troubleshooting activities. They are both optional applications.

The main purpose of NSD application is to show a real time overview of network status and to provide support for high level troubleshooting. NSD provides a combined, context based view of alarm status, key performance traffic indicators and state indicators for RNC, RBS, Cells and Channels. By sorting and filtering in different GUI views it is possible for the operator to quickly identify the source of a problem.

The NSA application is designed to be used by skilled users for fast identification of problem areas. The user receives an alarm or similar indication that the service level has degraded. The user will then launch NSA based on cell identity or RBS node identity and get fast indications of the area causing the service degradation. The user can then conduct the required corrective action from NSA directly or if more advanced actions are required launch the Element Manager or other applications in OSS-RC to correct the problem. In case more information is needed than is presented in the start window of NSA the user can continue to locate the fault by requesting more information. The NSA provides the user with the means to get a collected and comprehensive view of the status of an RBS from one single application. NSA displays the status for the RNC related information, Iub link, RBS node and RBS hardware.

2.2.9 Element Management Layer

The Element Management Layer consists of functions for a single RNC, RBS, or RANAG, such as the configuration of the equipment in an RBS. The RNC, RBS, and RANAG Element Managers (EMs) handle Fault Management on the Network Element level. The EMs provides a Graphical User Interface (GUI) for viewing and administering alarms, alarm logs, and FM event logs. The operator can open the EM application from OSS-RC, or access it directly on site with a web browser.

3 TOOLS/APPLICATIONS For TROUBLESHOOTING

3.1 OSS-RC

3.1.1 Alarm List Viewer (ALV) and Alarm Status Matrix (ASM)

Alarms triggered on a WCDMA RAN Network Elements are forwarded to the Fault Manager in OSS-RC. Then, Alarm Status Matrix displays the alarms, and alarm details are available in Alarm List Viewer. Integrated in these applications are also alarm logs.

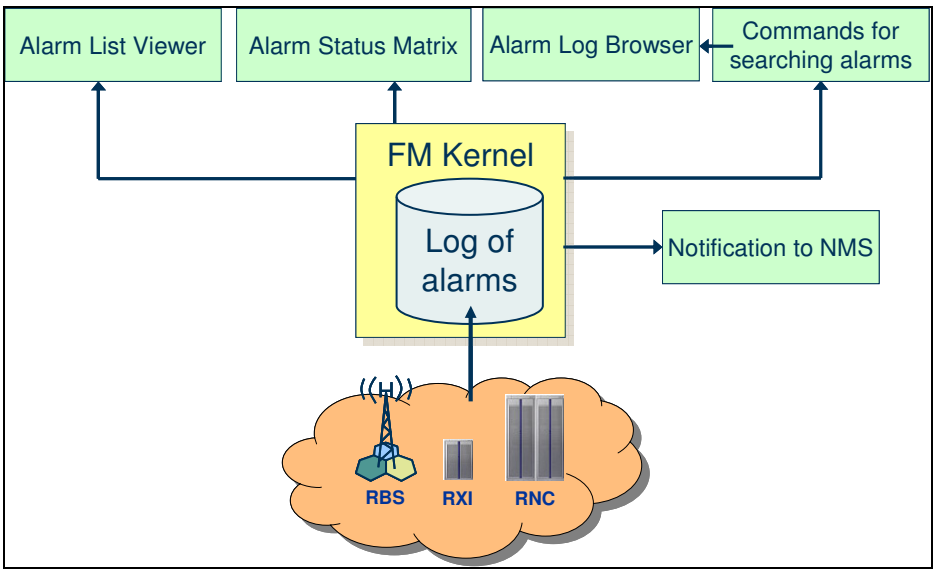


Figure 2-6: Alarm Notification Framework

3.1.2 Alarm Status Matrix

The Alarm Status Matrix gives an overview of the current alarm situation in the network. The Alarm Status Matrix allows you to do the following:

- Supervise several objects in a compressed view,
- Configure the user interface to show certain severities, toggle compact view on and off and dynamically add or remove rows and columns to change the number of objects possible to view,
- Access other applications: Start the Alarm List Viewer to view details about current alarms from a specific supervised object; Start the Alarm

Log Browser to access all logged alarms for a specific supervised object.

Synchronize the alarm list in the Fault Manager with the alarm list in a supervised object.

3.1.3 Alarm List Viewer

The Alarm List Viewer shows the complete alarm situation for one or more network elements in the network. To ensure a complete overview of the most important information the following features exist:

- Each Alarm List Viewer window can consist of any number of individual alarm lists.
- Each Alarm List shows the alarms for one or several network elements divided into one or several lists, where the sorting and filtering can be set for each list.
- The most important information for each alarm is shown in the alarm list, with one line per alarm. What information to show, in which width and in which order are defined for each list.

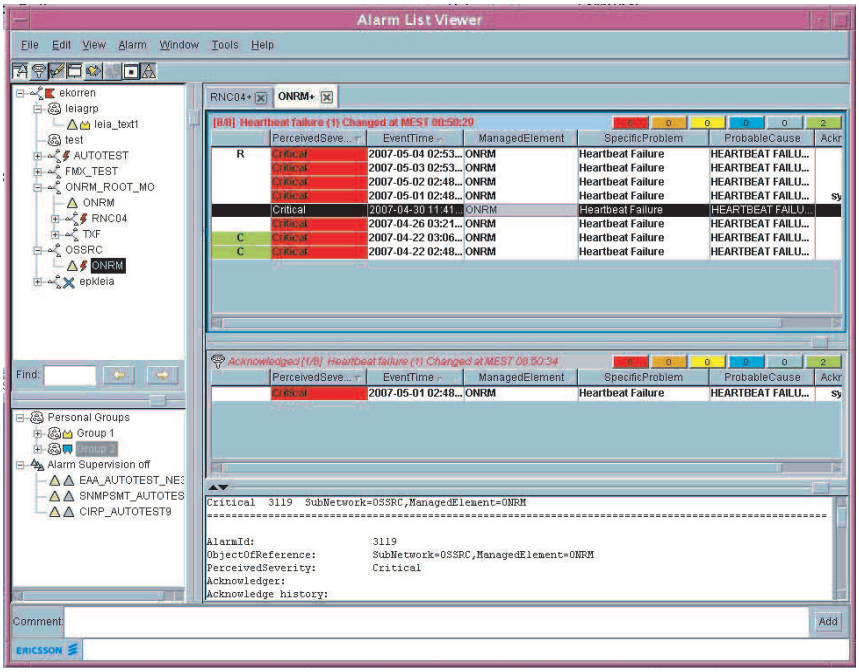


Figure 2-7: Alarm List Viewer

3.1.4 Alarm Log Browser

The Alarm Log Browser supports the user with functions to find and retrieve information about any alarms stored in the alarm log.

The user can select:

- The search criteria by specifying alarm attributes and which values or value ranges these attributes shall have
- Whether to present the retrieved alarms directly or to present the result of an analysis made on the retrieved alarms
- Whether to display the result or save it to a file

If the retrieved alarms are to be presented, the user can select the sorting criteria for the output.

If the result of the analysis is to be presented, the user can select whether the result is to be in an easy-to-read format or in a format suitable for import to a spread-sheet program

3.1.5 Alarm Handling Procedure

When a fault occurs in an NE, two different types of alarms can be issued:

- A primary alarm

  The primary alarm contains information about the original fault (the root cause).

- Corresponding secondary alarms

  The secondary alarms contain information about high level fault impacts generated by the original fault.

For example, an equipment fault in an RBS impacting a radio cell generates both a primary equipment failure alarm and a secondary service alarm in the RNC, indicating the degraded service in the cell.

The majority of fault situations produce only one primary alarm. A general rule is that the RBS reports only primary alarms and the RNC reports secondary alarms, unless the primary alarm appears in the RNC (in which case the RNC reports it).

The following diagram shows how to handle the alarm. Note that the procedure is the same whether one handles the alarm from the OSS-RC or from the Element Manager.

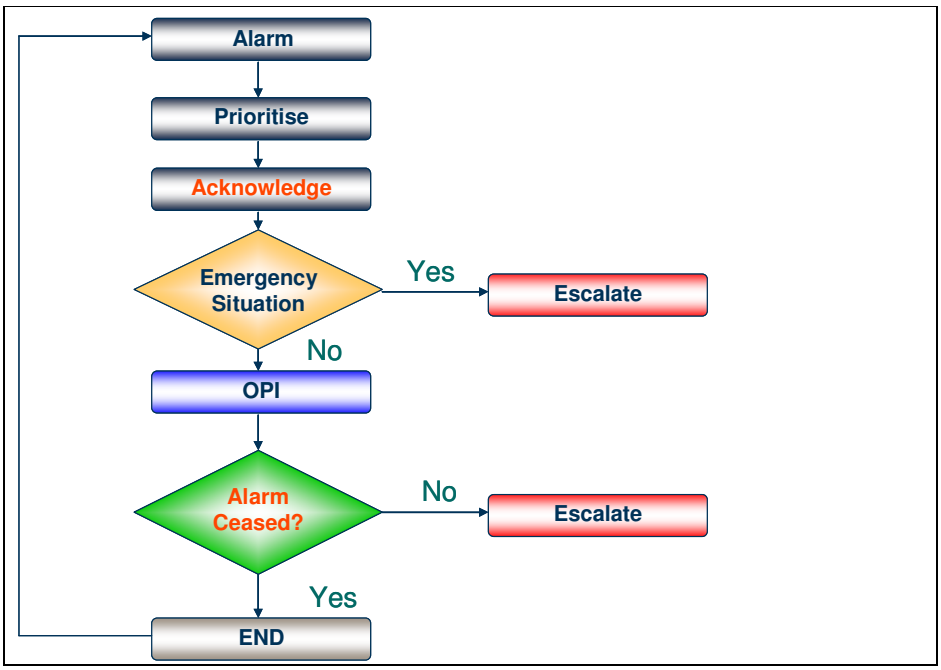


Figure 2-8: Alarm Handling Flow

Each alarm has the following details:

- **Perceived Severity**: Critical, Major, Minor or Warning,
- **Alarm ID**, unique within OSS-RC,
- **Event Time**,
- **Object of Reference**, pointing to the resource linked to the alarm,
- **Probable Cause,**
- **Specific Problem,** used to search within ALEX for the appropriate OPI.

3.1.6 FM Event Handling

The NE's normally generate an FM event when something of importance happens that does not trigger an alarm, but is considered significant enough to be presented for a user. FM events are stateless notifications. The NE's stores information regarding the event in the FM Event Log.

The Event Handling function, which is partly located in OSS-RC and partly in the RNCs, RBSs, and RANAGs, handles WCDMA RAN FM events. When a FM event occurs in an NE, the Event Handling function sends an FM event notification to OSS-RC. The FM events can be viewed in the OSS-RC FM applications.

FM Event is sent on the same interface as alarms from the NE's.

3.1.7 Network Status Display (NSD)

The Network Display Status (NSD) application displays information relating to network operation and performance. The information displayed ranges from high level summary information for the network to details of alarms for particular Network Elements. NSD displays network status information relating to fault management, performance management and configuration management information for WCDMA RAN and provides support for high level troubleshooting.

NSD is an optional OSS feature. Note that this application is integrated within the Common Explorer.

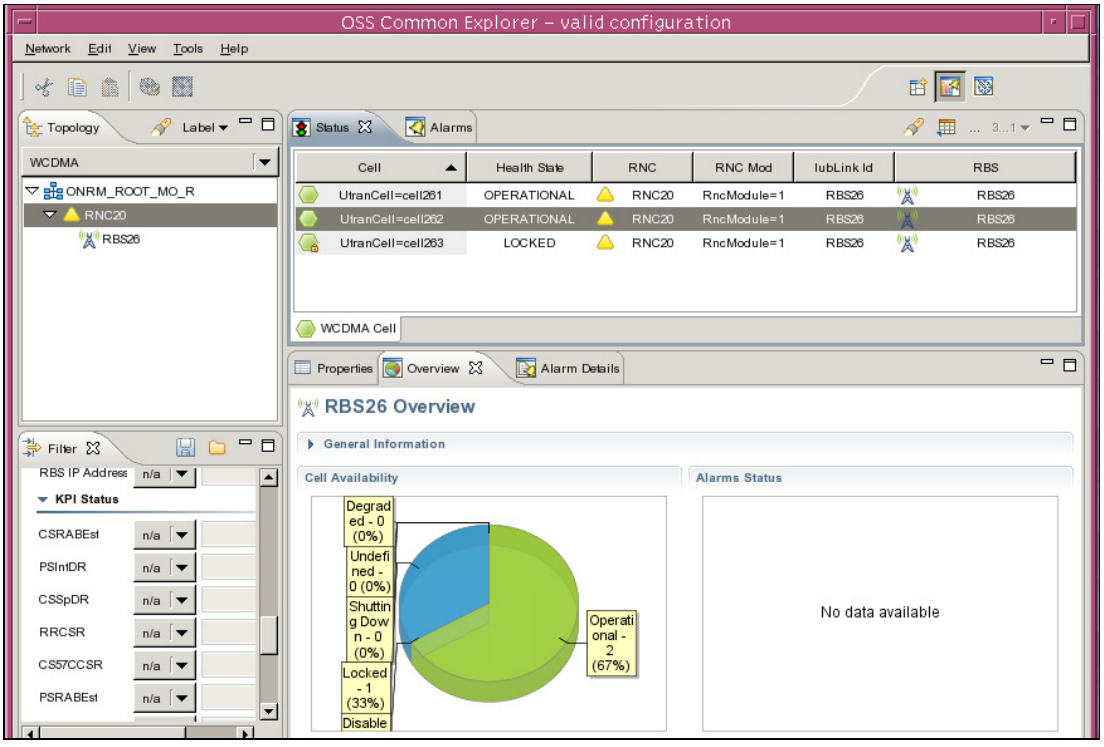


Figure 2-9: Network Status Display

NSD is an all-in-one application that displays various types of information in the following views:

- Topology
- Overview
- Status

- Filter
- Progress
- Alarms and alarm details
- Properties
- Logs
- Performance
- Traffic recording
- Outline

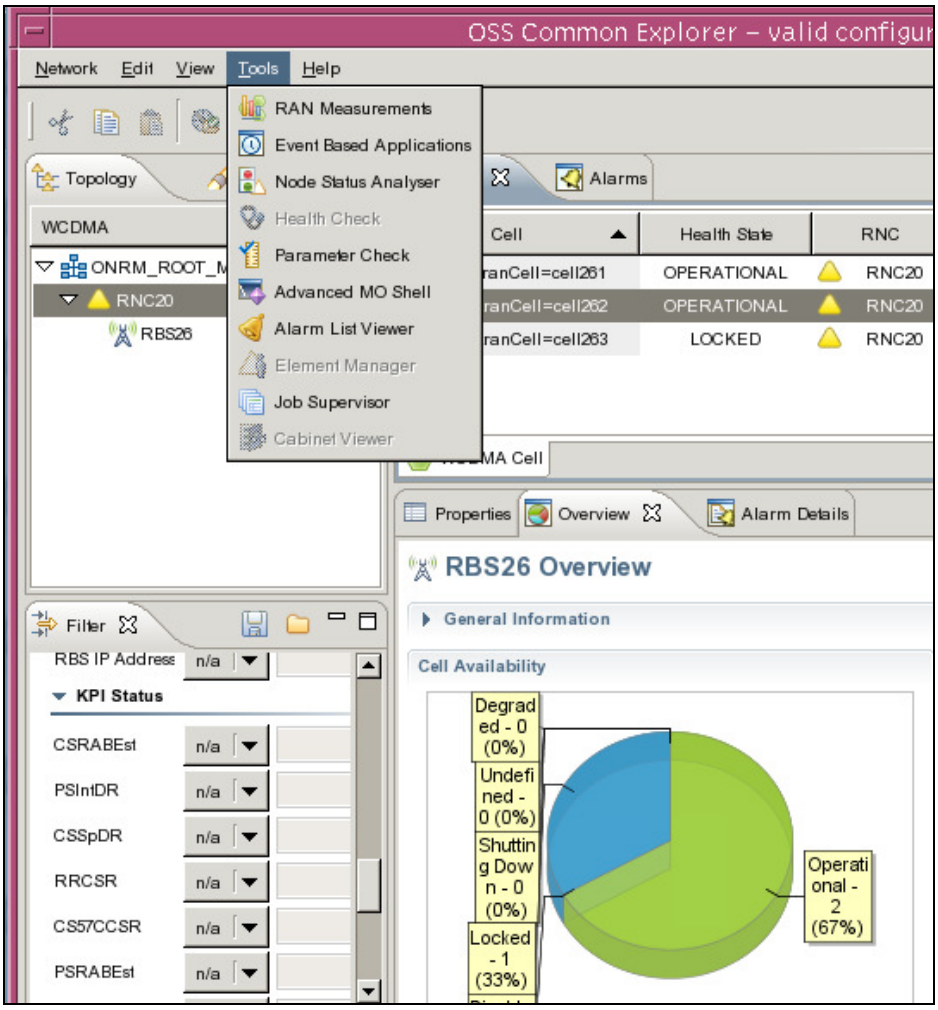


Figure 2-10: Network Status Display ..2

3.1.8 Transport Topology Viewer

Although not strictly a Troubleshooting tool, Transport Topology Viewer can be used to check the consistency of the transport links (in terms of VPI and VCI values) in the WRAN network. Furthermore, it can be used to perform End-to-end loopback tests on the VP and VC levels.

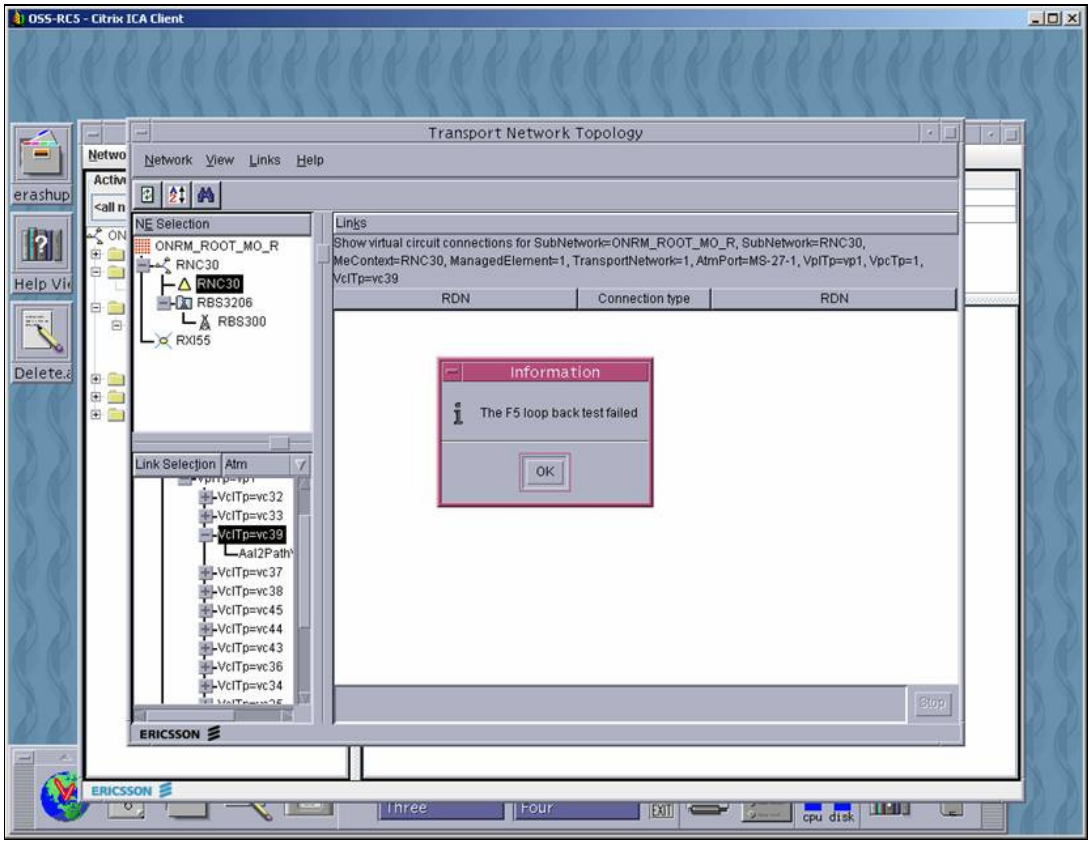


Figure 2-11: Transport Topology Viewer

3.1.9 Job Manager

The “Job Manager” application in OSS-RC consists of three separate applications which together implement certain tasks in the WRAN network elements. These three applications are the Task Editor, the Job Editor and the Job Supervisor.

A 'Job' is a series of activities that can be executed to fulfill a work order. An activity executes a single task, for example, to create a signaling link. In the context of the Job Manager application, the architecture of a job is as follows:

Job - contains one or more activities. A job cannot contain other jobs. Jobs do not have other dependencies, that is, they run independently of other jobs.

Activity - is a component of a Job. An activity may have a dependency on another activity within the same job. An activity executes a single task.

Task - is the basic building block within an Activity. It contains parameters that are set for a particular activity. A task operates on a single network element. A task contains no timing information.

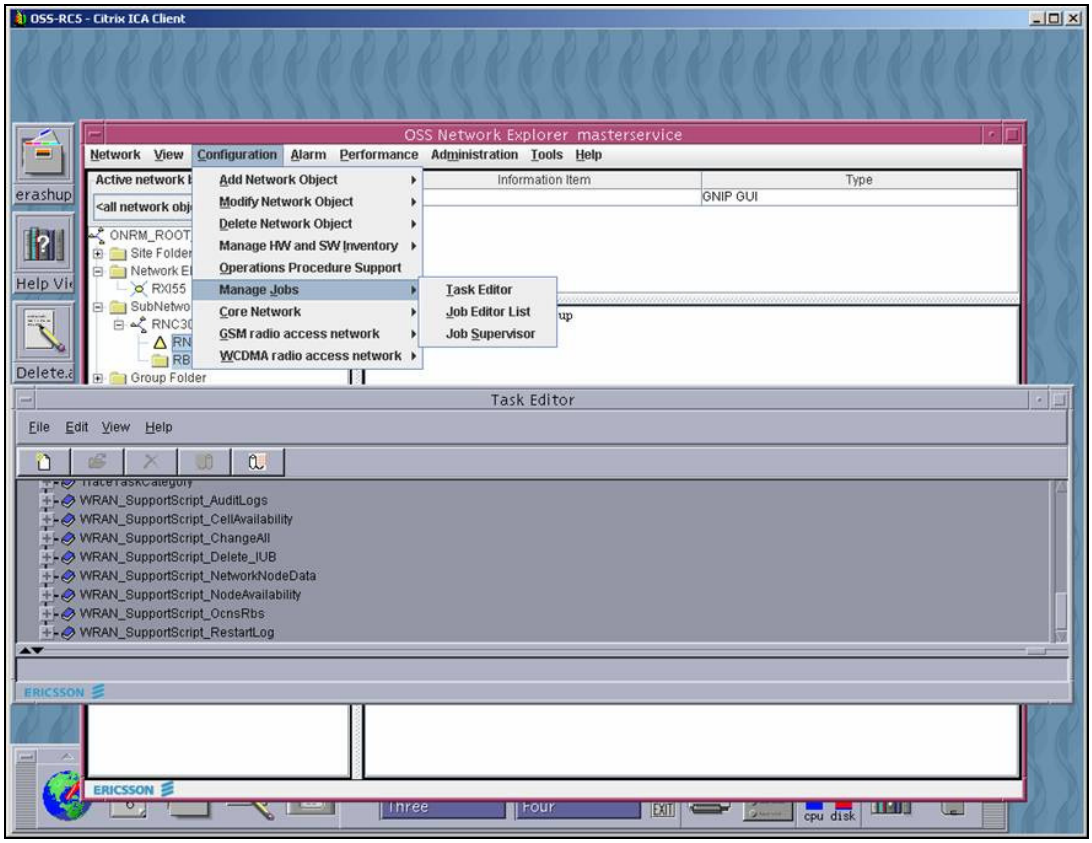


Figure 2-12: Job Manager Applications

From an operational point of view, the main workflow while creating and executing a job is as follows:

1 In the Task Editor, a Task Category is created. In the task category, a task is defined. The task is linked to a script. Usually, as a user, since most of the common tasks are already available in the OSS-RC, the Task Editor step can be ignored.

2 In the Job Editor, a Job Category is created. In the Job Category, a Job is created. In the Job, one (or several) activity is defined. The activity is linked to a task. During this process the network element (s) on which the activity should run is pointed out.

3 From the Job Supervisor, the Job that was created is scheduled (to run immediately, or schedule or to run periodically.) The output of the job is also displayed in the Job Supervisor. Job Supervisor also has a log of the previous Jobs.

From a troubleshooting point of view, the Job Manager is an interface that can be used to extract certain logs and perform health checks in the system. However, these functionalities are already implemented through a GUI interface in the OSS-RC. Therefore, although it runs in the background quite often, it is an interface one does not use directly.

4 COMMON EXPLORER

Common Explorer is the primary application in OSS-RC to work towards the WRAN (and LTE) nodes. The functions of the CEX can be summarized as:

- Network Topology or Configuration Viewing

CEX Views display detailed information relating to objects and their associations and relationships with other objects in the network. Some View settings can be used to limit the scope of displayed objects in other Views to display information of a particular type. CEX's viewing facilities can be useful for network monitoring and trouble shooting.

- Updating Network Topology Configuration

Using various CEX Views, Network Elements can be configured and managed, and objects added or deleted, as permitted by the View context. Individual attributes may be configured or bulk transactions may be used to make multiple changes to the network.

- Starting Applications

A number of operation and maintenance applications can be started from CEX.

- Network Status Monitoring and Trouble Shooting

CEX Views can be used to monitor network status, review the health of the overall network or particular parts of it, and examine various performance indicators. Views may be used to check consistency, assess performance, and if problems or potential problems are identified, take remedial action.

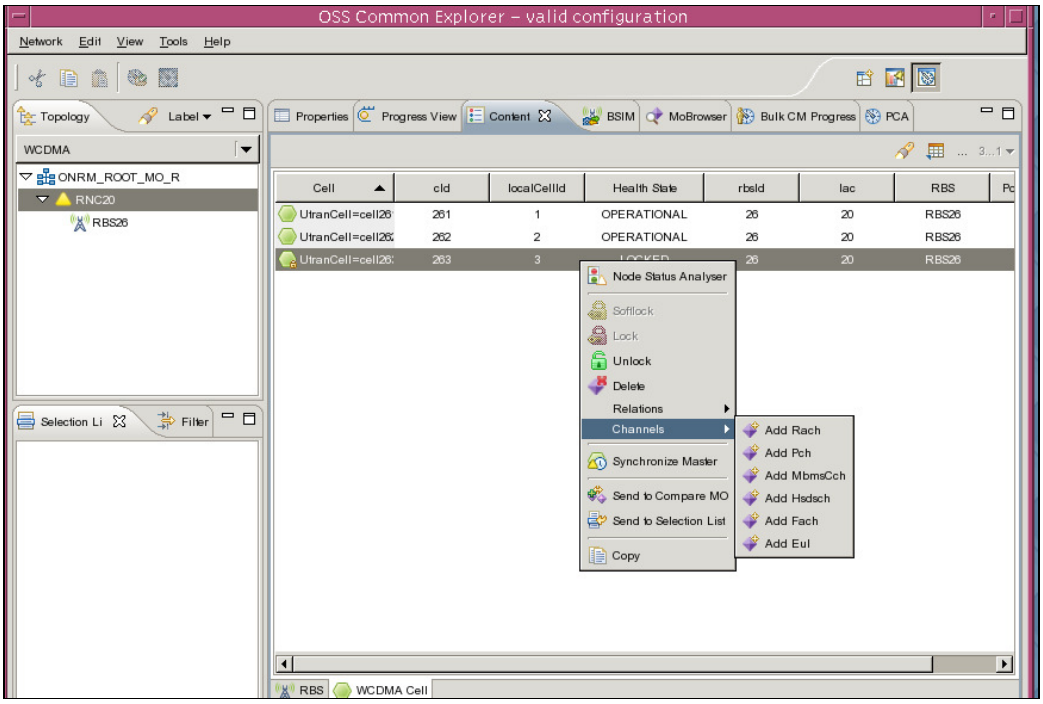


Figure 2-13: Common Explorer "Contents"

4.1 Network Status Analyzer and Cabinet Viewer

The NSA shows the end-to-end link from the RNC and the RBS ends, together with the alarms, and errors from the two nodes. From a troubleshooting point of view, because the alarms from the RNC and the RBS are displayed in the same window, it is easier to see the relation of the problem in one node to the alarm in the other.

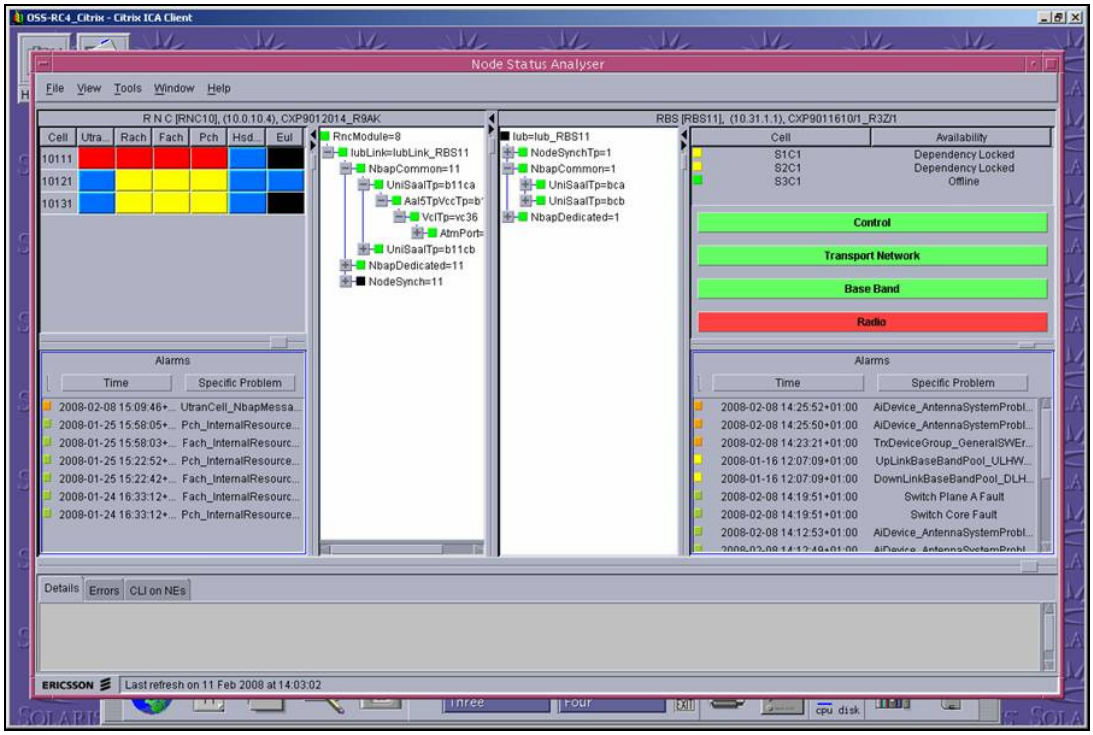


Figure 2-14: Node Status Analyzer

The Cabinet Viewer displays a graphical view of the actual layout of the RBS type including all selectable objects, and their LED status. A selectable object can be a board, a unit, a subrack within the RBS, or the RBS cabinet itself. Information about the power system and the external alarm ports is also shown.

It is also possible to test a board, and save logs from the node (both in the node level and the board level) in a file in the OSS-RC.

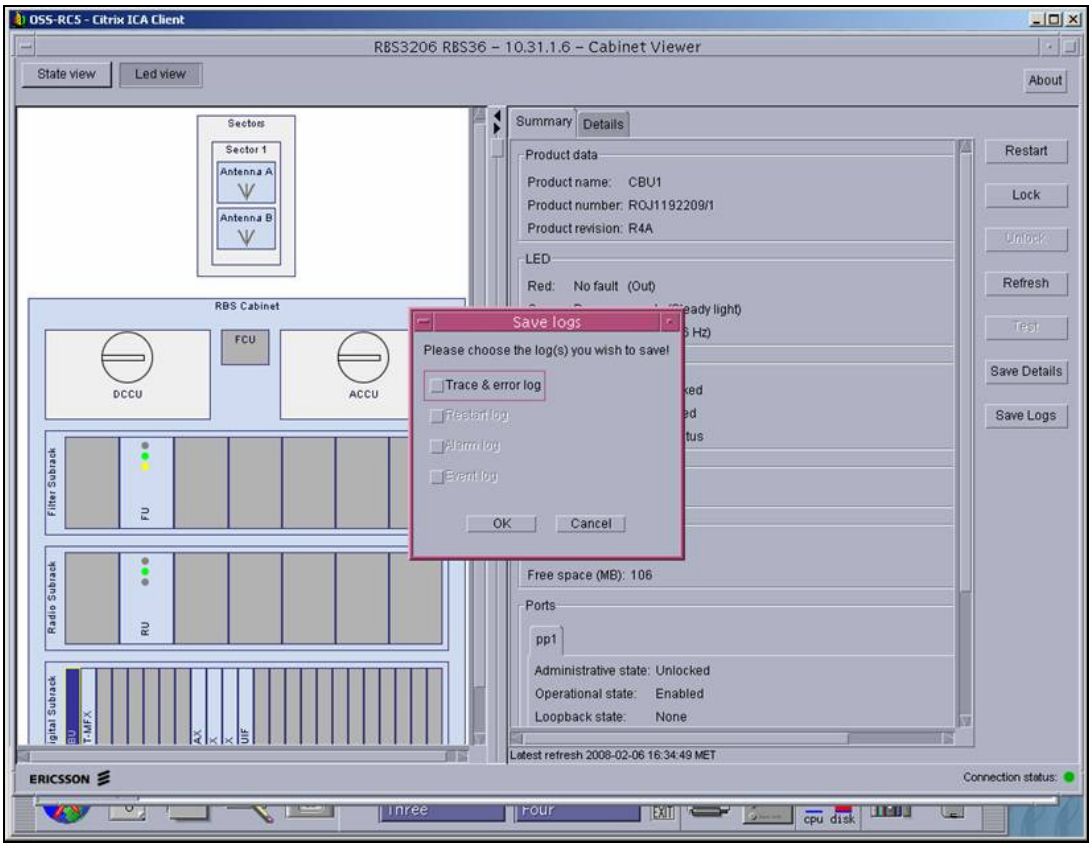


Figure 2-15: Cabinet Viewer

The Cabinet Viewer allows the following actions:

- Restart the RBS
- Restart a Board or Unit
- Lock/Unlock a board or Unit
- Test a Board or Unit
- Extract Logs from board
- Save Board Details

4.2 Health Check

Health Check allows an operator to choose a number of criteria to perform a health check on, and displays the output in an html file (and save the results in both html and .xml formats). The diagram below shows the health check criteria.

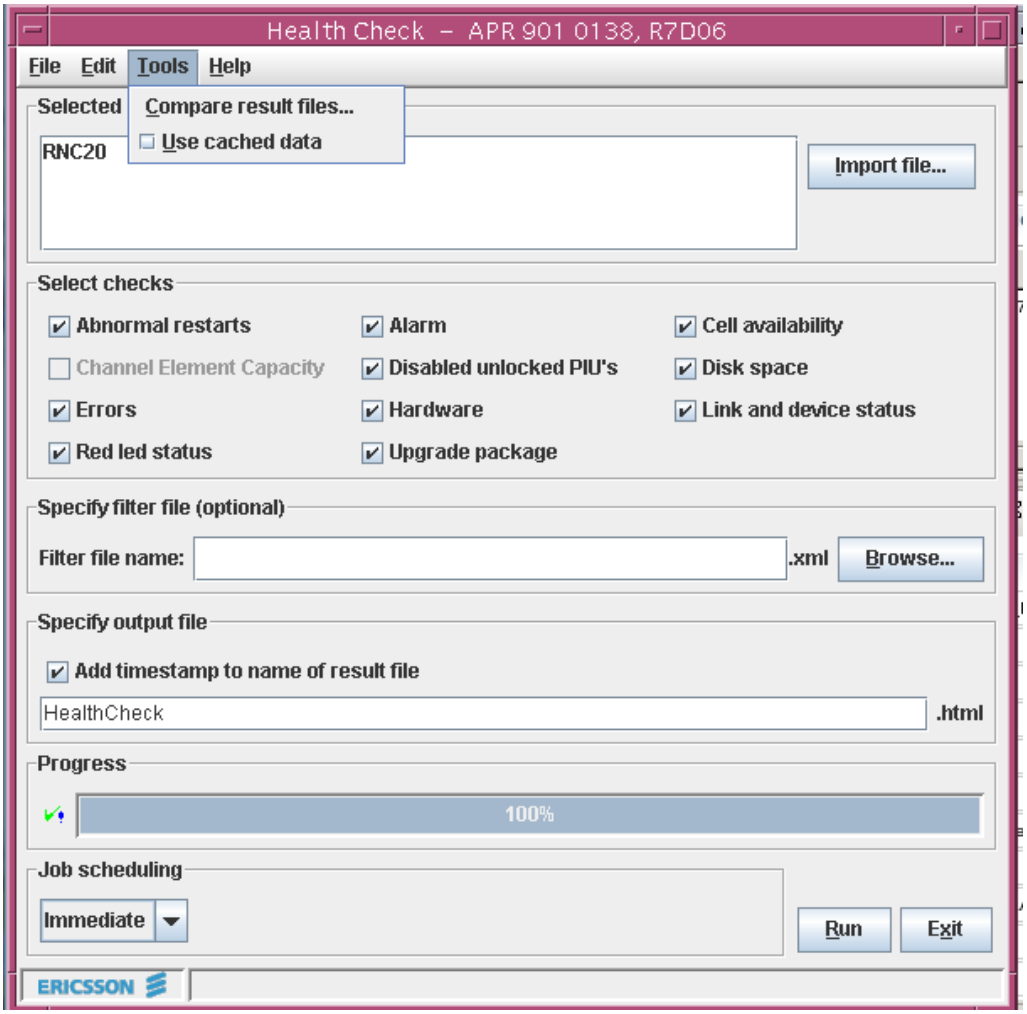


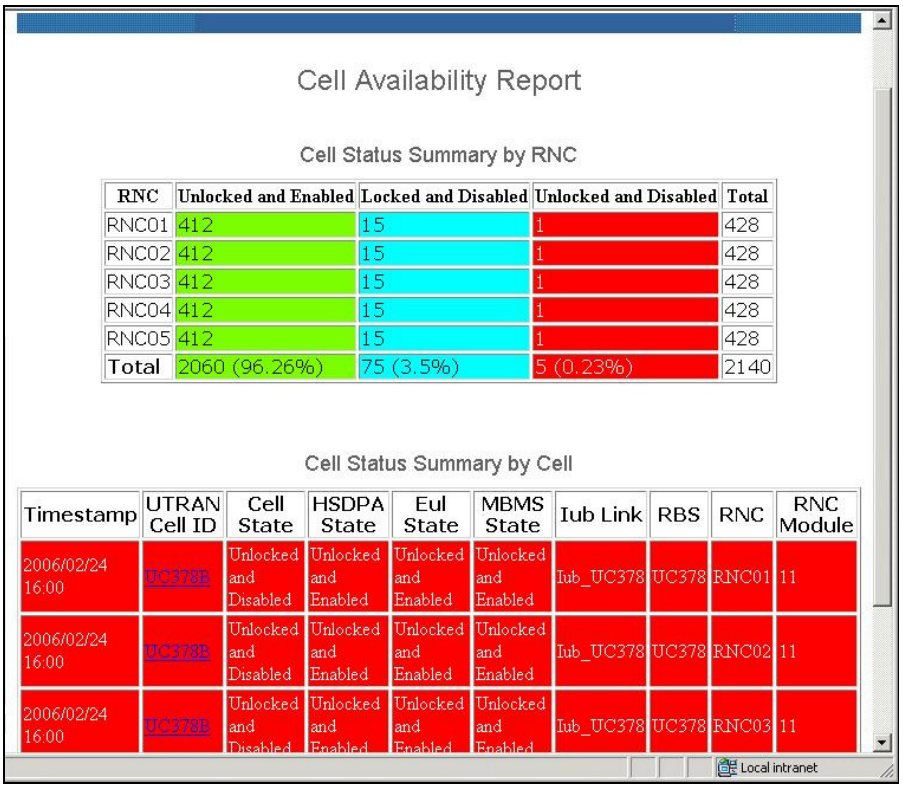
Figure 2-16: Health Check

From a troubleshooting point of view, health check could be looked as interface through which alarms, abnormal restarts could be collected.

In W11B Enhanced Health Check Filter File is included. This is an improvement of the filter file that is used by the OSS Health Check script to check if the installed RBS HW is supported by the currently loaded Software. The improved filter file provides a better possibility to pin-point non supported HW, for example HW not being SW loadable (not in the UCF file). This leads to fewer failures at RBS upgrade.

4.3 Cell Availability

The Cell Availability application parses the Statistics ROP files (that are generated in the NE and collected by OSS-RC) and presents the stats related to cell down time in a web-interface. When used together with the Bulk Configuration Management export file, the cell availability can be sorted per RncModule, IubLink, etc. It is also possible to save the details in a file.



Cell Availability Report

Cell Status Summary by RNC

RNC	Unlocked and Enabled	Locked and Disabled	Unlocked and Disabled	Total
RNC01	412	15	1	428
RNC02	412	15	1	428
RNC03	412	15	1	428
RNC04	412	15	1	428
RNC05	412	15	1	428
Total	2060 (96.26%)	75 (3.5%)	5 (0.23%)	2140

Cell Status Summary by Cell

Timestamp	UTRAN Cell ID	Cell State	HSDPA State	Eul State	MBMS State	Iub Link	RBS	RNC	RNC Module
2006/02/24 16:00	UC378E	Unlocked and Disabled	Unlocked and Enabled	Unlocked and Enabled	Unlocked and Enabled	Iub_UC378	UC378	RNC01	11
2006/02/24 16:00	UC378E	Unlocked and Disabled	Unlocked and Enabled	Unlocked and Enabled	Unlocked and Enabled	Iub_UC378	UC378	RNC02	11
2006/02/24 16:00	UC378E	Unlocked and Disabled	Unlocked and Enabled	Unlocked and Enabled	Unlocked and Enabled	Iub_UC378	UC378	RNC03	11

Local intranet

Figure 2-17: Cell Availability Report

Disabled and unlocked cells are not what are desirable in the network, and therefore it calls for actions. These cells should already have issued alarms, of course.

4.4 WCDMA RAN Load Expert

The WCDMA RAN load expert has the capability to find bottlenecks in the network, to do trend analysis and to predict a future capacity upgrade among the monitored network resources.

The feature provides a tool that is realized in the form of pre-defined reports in ENIQ. It was first introduced in W10B as a basic version including only one network resource, channel element load. In W11B, the tool has been enhanced with trend analysis and forecasting and now supports an increased number of monitored network resources in the RBS; downlink code tree load, downlink transmitted carrier power, uplink load, number of HSPA users. The load is measured based on PM counters in WRAN.

WRAN measurements are started from OSS-RC where there is a list of existing counters provided to point out what measurements that can be started. The measurements are started in a selected WRAN node and results can be fetched in the end of each measurement period. The results are stored as files and are not suitable for viewing. For that, there is a tool called ENIQ that provides a presentable form for viewing these results in form of reports.

This feature introduces new pre-defined reports in ENIQ that shows the load in an operator's WCDMA RAN or a chosen parts of it. WRAN measurements are the source to what load is presented in the reports.

The measurements are started by activating PM counters. Based on the counters, a set of formulas are specified for each network resource to give a view of the network load on different parts of the network.

Here follows a list of these network resources which can be monitored:

4.4.1 Downlink code tree load

Downlink physical channels are distinguished by means of downlink channelization codes, selected from a code tree. There is only one downlink code tree available for each cell, therefore it is a limited resource. Codes for CCH are statically allocated by the RNC and a certain amount of codes is reserved for HSDPA. Codes for DCH are allocated dynamically by the RNC, and the remaining codes can be allocated to HSDPA by the RBS.

4.4.2 Downlink transmitted carrier power

Downlink transmitted carrier power is a limited resource in WCDMA system. The same pool of power is shared between all physical channels carrying all user plane traffic and all control plane signaling.

4.4.3 Uplink load

DCH and EUL uplink traffic generates interference which adds to the background noise creating an uplink noise rise. From a capacity perspective, a cell cannot handle more than a certain amount of noise rise/load due to the risk of instability.

4.4.4 Number of HSPA users

The number of HSPA users is limited by licensing in uplink and downlink. When the number of users reaches the license limit, no more HSPA users are admitted to the cell.

4.4.5 Channel element load

A channel element (CE) is used as a measure of board capacity and radio bearer capacity. Channel element utilization is checked against both licensed capacity and hardware capacity to predict required license limit increase and hardware capacity expansion.

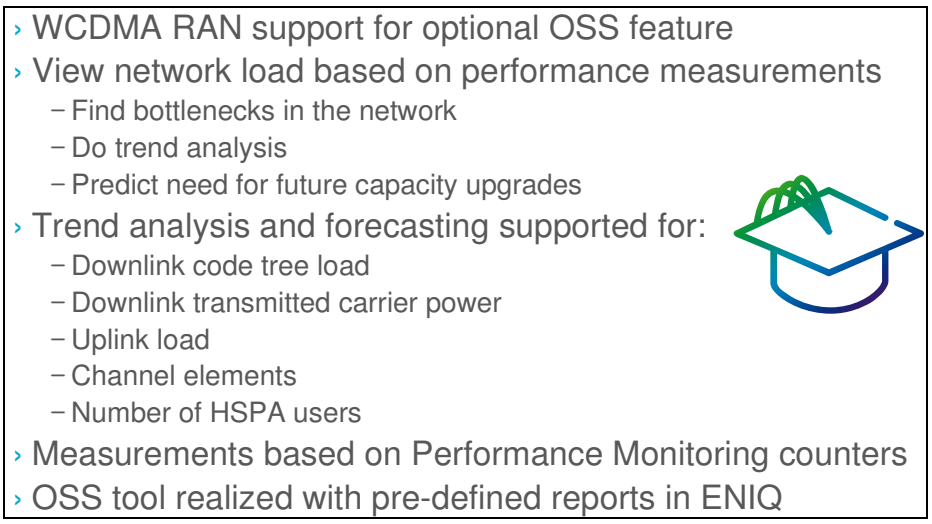


Figure 2-18: RAN Load expert

The WCDMA RAN load expert is a new optional feature that enables operators to view the network load of their WCDMA RAN as a result based on performance measurements. It has the capability to find bottlenecks in the network, to do trend analysis and to predict a future capacity upgrade among the monitored network. This feature is an optional feature in ENIQ. It is handled with one license in resources. The load is measured based on PM counters in WRAN.

ENIQ, the same license as for the feature Channel Element Load

Activation:

The activation of this feature consists of three steps, both made in OSS-RC:

• Activate the license in ENIQ. A description of this is available in the CPI for ENIQ.

• Start the PM counters in OSS-RC for the load measurements of each network resource. A description of this is available in the CPI for OSS-RC.

• Start the WCDMA RAN load expert report. A description of this is available in the CPI for ENIQ

The following network resources can be monitored:

4.4.6 Downlink code tree load

Downlink physical channels are distinguished by means of downlink channelization codes, selected from a code tree. There is only one downlink code tree available for each cell, therefore it is a limited resource. Codes for CCH are statically allocated by the RNC and a certain amount of codes is reserved for HSDPA. Codes for DCH are allocated dynamically by the RNC, and the remaining codes can be allocated to HSDPA by the RBS.

4.4.7 Downlink transmitted carrier power

Downlink transmitted carrier power is a limited resource in WCDMA system. The same pool of power is shared between all physical channels carrying all user plane traffic and all control plane signaling.

4.4.8 Uplink load

DCH and EUL uplink traffic generates interference which adds to the background noise creating an uplink noise rise. From a capacity perspective, a cell cannot handle more than a certain amount of noise rise/load due to the risk of instability.

4.4.9 Number of HSPA users

The number of HSPA users is limited by licensing in uplink and downlink. When the number of users reaches the license limit, no more HSPA users are admitted to the cell.

4.4.10 Channel element load

A channel element (CE) is used as a measure of board capacity and radio bearer capacity. Channel element utilization is checked against both licensed capacity and hardware capacity to predict required license limit increase and hardware capacity expansion.

4.5 Advanced Managed Object Scripting

- AMOS is a text-based Operation and Maintenance (O&M) client providing access to the following services:
- Alarm Service (AS)
- Configuration Service (CS)
- File Transfer (FTP/HTTP)
- Inventory Service (IS)
- Log Service (LS)
- Notification Service (NS)
- OSE Shell (COLI)
- Performance Measurement Service (PM)

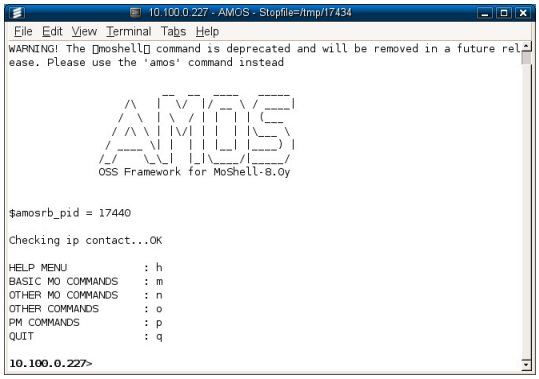


Figure 2-19: AMOS overview

Access to all services is supported in both secure mode (secure CORBA, SSH, SFTP) and non secure mode (non secure CORBA, TELNET, FTP). From that perspective, AMOS is an EXTREMELY powerful interface.

AMOS is an optional feature in the OSS-RC. (This used to be the internal Ericsson tool “MoShell”). The use of AMOS is described in the Appendix 1.

4.6 Network Element Logs retrieval from OSS-RC

It is possible to retrieve logs from the WRAN network elements and store them in OSS-RC. Different applications (mentioned above) retrieve different logs. The types of logs available in the network elements are explained in the later in the chapter.

4.7 Element Management GUI

Besides the applications in the OSS-RC (some of which are optional features there), the Element Management GUI and the Object Explorer GUI are the two most important interfaces with which one operates the WRAN node.

From a trouble shooting perspective, both these GUIs show the **operational state** of the Managed Objects (MOs), which, if they are disabled, suggest that there is a problem. However, there are other tools that might be useful to complement the information in the GUI.

To ensure that the configuration is complete, it is also advisable to make sure that the field 'reserved' is displayed as Yes in the Element Manager. However, one should note that there are certain MOs which do not be reserved- example, the Aal2RoutingCase in RBS need not be reserved.

Note that the EM GUIs allow the active Alarm List, Alarm Log and Event Log to be displayed, which are important parts of troubleshooting process.

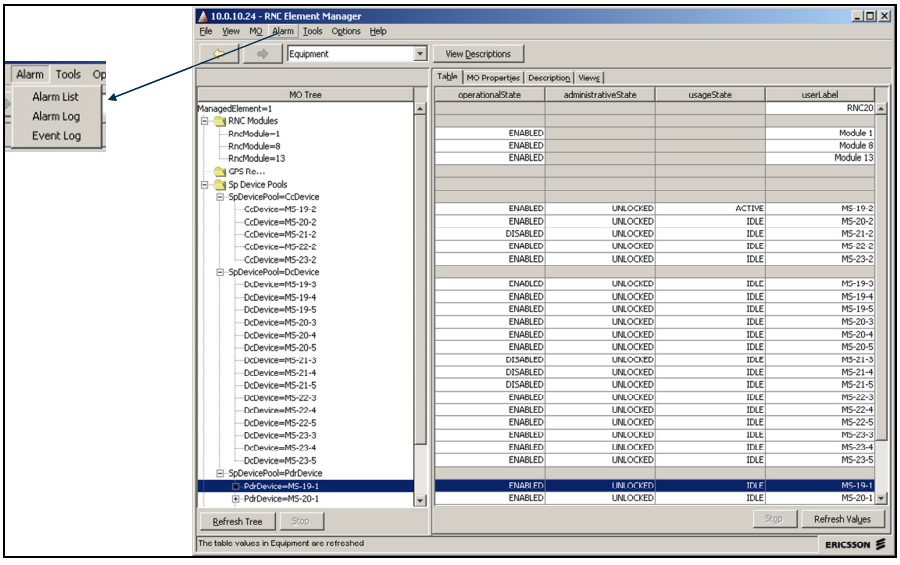


Figure 2-20: Alarm/ Events with Element Manager

Note that the EM and OE GUIs can also be started from the OSS-RC.

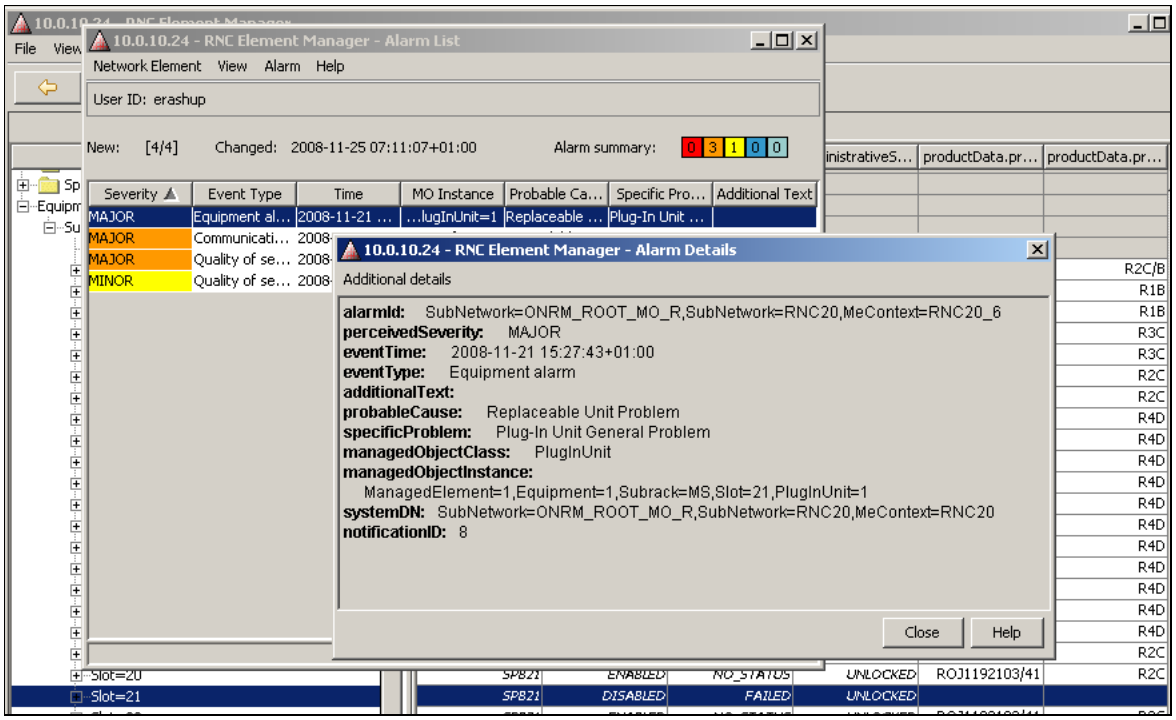


Figure 2-21: Alarm Details

4.8 Node Command Line Interface

The Node Command Line Interface is a shell command interface, which is used to manage a node in the network, by manipulating Managed Objects (MO). Just like with the Element Manager and the Object Explorer, it can be used to configure the CPP network element.

NCLI is accessed via the normal COLI in the central GPB (or CBU) in the CPP based node.

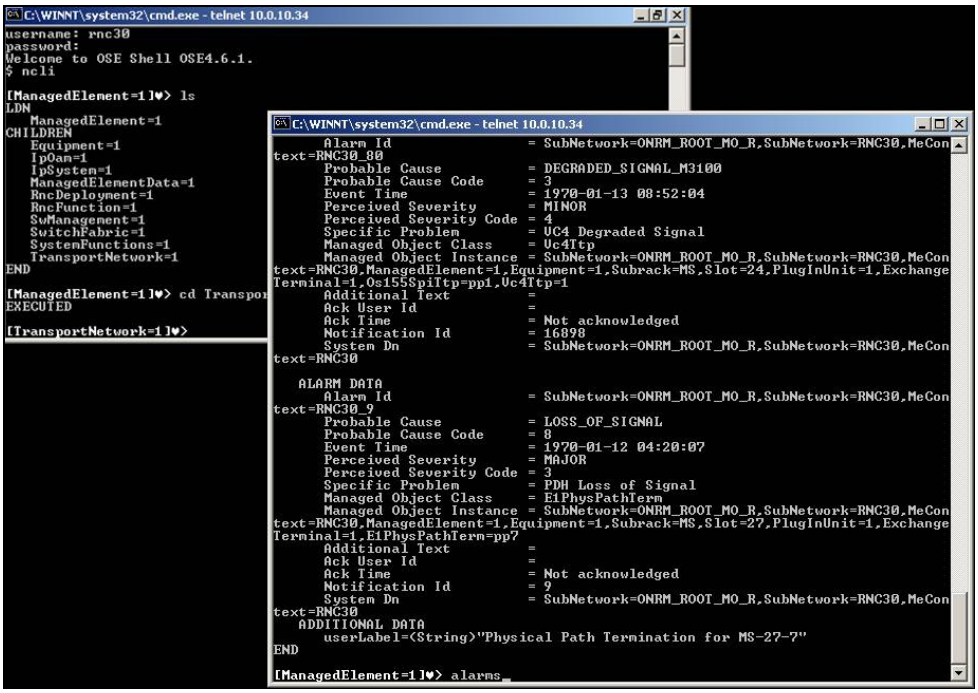


Figure 2-22: Node Command Line Interface (NCLI)

From a troubleshooting perspective, this interface is useful for listing active alarms, and for checking the parameters of the Managed Objects.

4.9 COmmand Line Interface

While the Node Command Line Interface (NCLI) works with the Managed Objects, the normal COLI (sometimes referred to as CLI) is a shell interface in the operating system in the processor. Therefore, this interface does not work with Managed Objects, which are what the normal O&M interfaces (most OSS-RC applications, EM, NCLI, OE) work with.

While an operator should always use the MO attributes/actions, together with the Performance Stats and recordings to troubleshoot problems in the network, COLI is an interface one should be aware of, since it allows access to logs and tracing possibilities in a CPP node. Note that trace initiation should only be done by Ericsson personnel. The logs and trace results are useful while analyzing faults in the node.

4.10 Miscellaneous

4.10.1 Test Functions

After actions are taken by the operator, tests are performed, either manually or automatically by the system. Depending on the result of these tests, more actions might be needed.

- In addition to automatic supervision an operator can manually test some functions
- Can perform the following manual tests:
  - RBS Equipment Tests
  - ATM end-to-end loopback tests

Figure 2-23: Test Functions

The operator can perform the following manual tests:

- Equipment Tests

This function makes it possible for the operator to perform a self-test on most of the boards, especially the RBS Device boards and Auxiliary Units that are processor controlled. Note that the equipment test can also be performed while performing a restart with the 'cold with test' rank.

- Asynchronous Transfer Mode (ATM) end-to-end loopback tests

This function makes it possible for the operator to perform an ATM end-to-end loopback test. This test checks link connections between nodes in the network.

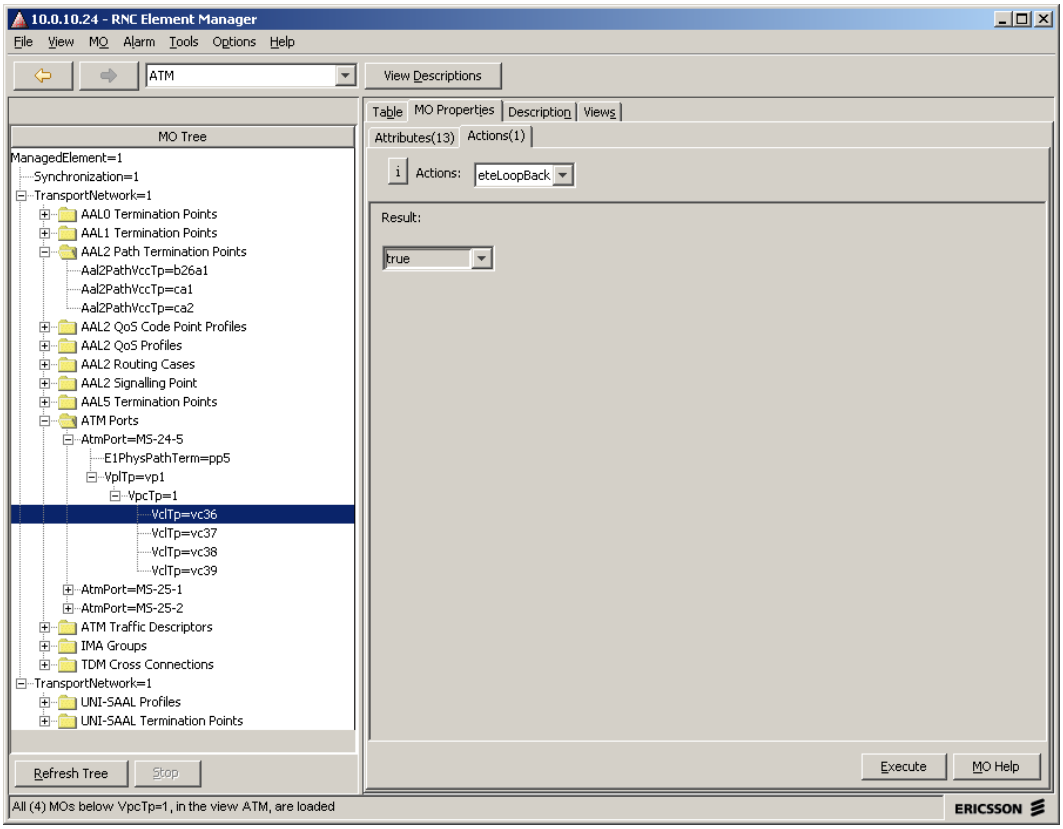


Figure 2-24: End-to-end loopback test

4.10.2 Man Machine Indicator (MMI)

A maintenance mode button is introduced in RBS 6000 together with related support functions in WCDMA RAN. This allows the service technicians on site to lock/unlock boards without connecting a PC to the base station. Lock/unlock is made possible by pressing a button on the concerned board.

Common MMI appearance is introduced to make it easier to work with DUW, DUG and DUL. HW appearance will be the same regardless if there is a DUW, DUG or DUL

The RBS 6000 uses LED INDICATOR colors as follows:

Red color indicates a major fault. It is used to show that there is a HW fault that must be handled on that specific unit.

Green color is used for indicators that will be lit when the Node is working normally.

Yellow color indicates a warning. It is used to show that something, it can be a unit or the Node, is not working correctly.

Blue color is used to indicate that maintenance is ongoing. It is used on Indicators that the operator can influence via for instance the Maintenance Button.

A fully operational RBS shall not have any yellow or red Indicators lit, and there shall be no Indicators blinking.

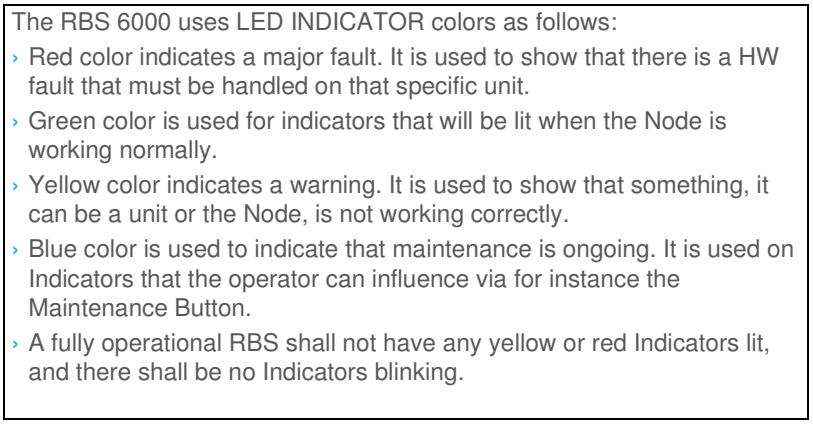


Figure 2-25: MMI on the Plug-in-Units

4.10.3 Product Inventory

The product inventory feature makes it possible to keep control of the installed base of RBS, RNC and RXI nodes and their current status regarding hardware, software and license keys. The UTRAN nodes reports via events when the inventory data - HW, SW or license – is changed. The inventory function in OSS-RC uses these events to keep the inventory base updated with the network at all times. Via OSS-RC the information can be forwarded to an external system, such as an asset manager.

For SW and HW products the following information is available:

- Product number
- Product revision
- Product name
- Production date (only for replaceable hardware units, loadable software and upgrade package)

- Serial number (only for replaceable hardware units)

For SW products the following information is also available:

- The currently executing upgrade package on the node
- The load modules constituting each upgrade package on the node

For license keys the following information is available:

- License key identity
- License key description (Feature or Capacity name)
- Validity period
- Capacity limit (only for Capacity features)

> - In W11B Auto Inventory Events are introduced to inform the inventory function in OSS-RC that inventory data of a node is changed.
>
> The following operator control is available for the feature:
>
> - **lastHwPiChange**, indicates the timestamp when the last hardware product information is changed.
> - **lastUpPiChange,** indicates the timestamp when the last software product information is changed.
> - **lastLicensingPiChange**, indicates the timestamp when the last licensing product information is changed. This is triggered by the action updateLicenseKeyFile

Figure 2-26: Product Inventory

The information is maintained when changes are made. Site-specific parameters are set via the element manager at site commissioning. If a replaceable hardware unit is removed, for instance for repair, it is possible to read the information for that unit at the repair centre. It is possible to compare the software on a selected set of nodes with a reference node to identify differences.

In WCDMA RAN W11B the following is new:

Auto Inventory Events are introduced to inform the inventory function in OSS-RC that inventory data of a node is changed.

This feature is enhanced with the function Inventory Auto Adjust with the purpose to enable OSS-RC to swifter discover changes to the HW, SW and/or Licenses in the managed nodes. Instead of the previously available methods where either the operator manually orders synchronization or that OSS-RC synchronizes at scheduled intervals, the node shall be able to signal to OSS-RC that change has occurred.

The new CM Notifications introduced for a central item (SMO), impacts the traffic model for O&M and transport network.

The following operator control is available for the feature:

• **lastHwPiChange**, indicates the timestamp when the last hardware product information is changed.

- **lastUpPiChange,** indicates the timestamp when the last software product information is changed.

• **lastLicensingPiChange**, indicates the timestamp when the last licensing product information is changed. This is triggered by the action updateLicenseKeyFile

4.10.4 Lock/Unlock Procedure

While locking a configured resource (which is actually a Managed Object), there could be two options: Soft Lock and Hard Lock.

- **Soft lock:** the execution of the applications, or functions, is shut down gradually. The applications are given some time to finish their tasks up to a certain point. After that, the board is locked.
- **Hard lock:** application is terminated and the resource is locked immediately.

Figure 2-27: Hard and soft locks

If soft lock takes too long, one can interrupt it by performing hard lock or by unlocking the board.

It is also possible to lock other resources, like cells, device, etc. Similar logic of hard and soft lock applies as it is for equipment.

4.10.5 Restarts

While restarting a board or the whole node, one of the following restart ranks could be chosen:

- A **warm restart** restarts the node with the same software. The software is not reloaded from the disk.
- A **refresh restart** reloads the software from the disk, restarts the node, and resets parts of the hardware.
- A **cold restart** reloads the software from the disk, restarts the node, and resets all hardware.
- A **cold with test restart** reloads the software from the disk, restarts the node, and resets and tests the hardware.

Figure 2-28: Restart Ranks

Traffic is affected in all these situations. The amount of traffic disturbance depends on the restart rank chosen.

It is also required to provide a reason why the restart is being performed. This is only used for statistics.

SUMMARY

After this chapter the participants should be able to:

2 Use the applications in OSS-RC, Element Manager and COmmand Line Interface (COLI) that are important during a troubleshooting procedure.

2.1 Start and understand when to use the following applications in OSS-RC: Transport Network Viewer, Job Manager, Alarm List Viewer/ Alarm Status Matrix, WCDMA RAN Performance Measurements, Common Explorer GUI, Node Status Analyzer and Cabinet Equipment Viewer, Advanced Managed Object Scripting (AMOS) interface and Node Command Line Interface (NCLI).

2.2 Lock and restart boards and nodes including the soft/hard lock concepts.

2.3 Check the status of the Manage Object to find out the health of the node.

2.4 Understand when COLI is used and when Element Manager/NCLI are used.

2.5 Describe Enhanced Health Check Filter File.

2.6 Describe Product Inventory improvements.

2.7 Understand the WCDMA RAN load expert.

2.8 Understand Supervision of SP pool and 4 Way Receiver Diversity with DUW.

2.9 Understand changes in FM Events and Changes ROP

Figure 2-29: Summary of Chapter 2

3 Logs and Traces

Objectives

After this chapter the participants will be able to:

3 Investigate the purpose and the location of the various types of logs in a CPP based node

3.1 Know the location and purpose and read Alarm and Event logs.

3.2 Explain how Ericsson Local Support enables traces in the process of troubleshooting, and uses the target monitor application to capture the traces.

3.3 Find out the location and purpose of Error Log, Post Mortem Dump(PMD) Log and Availability Log.

3.4 Find out the purpose and location of the Security and Audit trail logs.

3.5 Perform data collection to include in the Customer Service Request (CSR) when a problem is suspected in the WRAN network.

3.6 Explain Trace improvement, UE-ID sent to RBS in all Radio link Setup messages.

3.7 Describe Trace Overload Protection and RNC Throughput Capacity.

3.8 Describe RBS MP Load sharing between DUW's.

Figure 3-1: Objectives of Chapter 3

Intentionally Blank

1 OVERVIEW

This chapter lists the logs available in a CPP based node, and also introduces the principles of tracing. Some CLI commands are also listed that could be useful for troubleshooting purposes.

As mentioned in the previous chapter, bad performance statistics (in the form of Key Performance Indicators) and alarms are primary reasons for starting troubleshooting activities. Therefore, the logs are tracings can be considered as complements to the troubleshooting procedure.

2 Logs in CPP

There are several logs that are available in the CPP based nodes which are useful during troubleshooting. It is often required that when a problem is escalated in the form of a Customer Service Request (CSR), these logs are included as attachments.

- Alarm Log
- Event Log
- Error Log
- Post Mortem Dump
- Audit Trail Logs
- Availability Log
- System Log
- Trace Log for Software Handling and Upgrades
- Hardware Inventory Log
- Trace and Error Log

Figure 3-2: Logs in a CPP node

These logs, together with the performance stats and recordings provide Ericsson enough information to pinpoint the reason for the problem, and suggest a solution on how to fix the problem.

2.1 Alarm Log

If the “administrative State” of a Managed Object is ‘unlocked’ but its “operational State” is ‘disabled’ it indicates that there is a problem, and in most cases, an alarm is issued from the Managed Object. Alarms are issued so that the operator of the node should act on it to fix it. When the alarm is issued, it is not only displayed in the Active Alarm List, but it is also logged in the Alarm Log.

If the alarm is cleared (either by operator intervention or by itself), the active alarm list will clear it, while the alarm log will note down the time it was cleared.

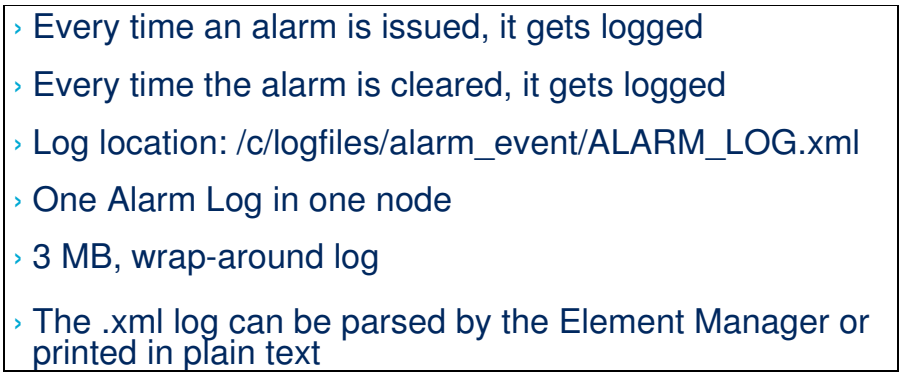


Figure 3-3: Alarm Log

The alarm log is saved in the CPP node in the following file:

/c/logfiles/alarm_event/ALARM_LOG.xml

The format of the log entry is shown below:

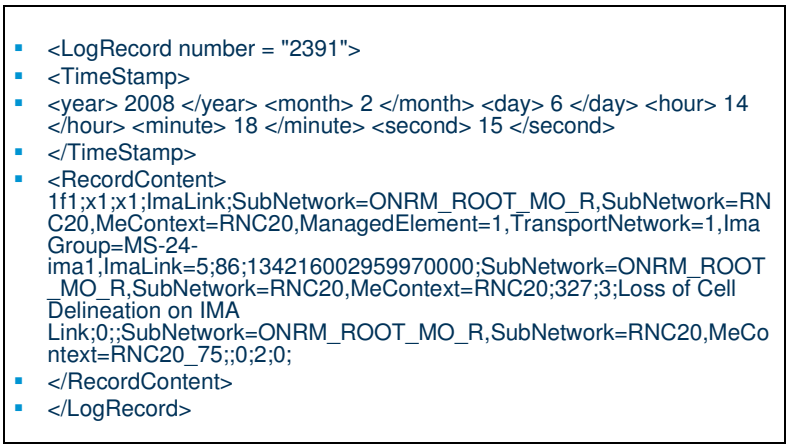


Figure 3-4: Alarm Log- example

This xml log can be parsed by the Element Manager into readable text format, exported to a file or printed. It is also possible to retrieve (through ftp) these logs into the OSS-RC by various applications there.

Alarm Log is a wrap log that has a default size of 3Mb. There is only one Alarm Log in one CPP node.

2.2 Event Log

An event is an occurrence in the system that is worth being logged. It does not require any manual intervention, however, from a troubleshooting point of view, it is advisable to check the contents as a routine.

- An event notification is something of importance, but that does not trigger an alarm. Events are stateless.
- Log Location: /c/logfiles/alarm_event/EVENT_LOG.xml
- One Event Log in one node
- 500 KB, wrap-around log
- The .xml log can be parsed by the Element Manager or printed in plain text

Figure 3-5: Event Log

The event log is saved in the CPP node in the following file:

/c/logfiles/alarm_event/EVENT_LOG.xml

The format of the log entry is shown below:

```
<LogRecord number = "1279102">
<TimeStamp>
    <year> 2008 </year> <month> 2 </month> <day> 14 </day>
    <hour> 4 </hour> <minute> 16 </minute> <second> 18 </second>
  </TimeStamp>
<RecordContent>
1z1;0;z1;E1Ttp;SubNetwork=ONRM_ROOT_MO_R,SubNetwork=RNC30,MeContext=RNC30,ManagedElement=1,Equipment=1,Subrack=MS,Slot=24,PlugInUnit=1,ExchangeTerminal=1,Os155SpiTtp=pp1,Vc4Ttp=1,Vc12Ttp=43,E1Ttp=1;382738;1222043137800600000;SubNetwork=ONRM_ROOT_MO_R,SubNetwork=RNC30,MeContext=RNC30;E1Ttp_NOCRC4MFA;;0;
  </RecordContent>
</LogRecord>
```

Figure 3-6: Event Log. example

This xml log can be parsed by the Element Manager into readable text format, exported to a file or printed.. It is also possible to retrieve (through ftp) these logs into the OSS-RC by various applications there.

Event Log is a wrap log that has a default size of 500Kb. There is only one Event Log in a CPP node.

2.3 Error Log

Every processor in the CPP node keeps a record of restart information. This log is the error log, and is retrieved from the processor with the CLI command “llog”.

- A record of restart information
- Every processor has its own error log
- CLI command “llog” is used to print the log
- Useful for figuring out the restart time, and the process that led to the restart for that particular processor.
- During a restart, there could be a reference to a post-mortem-dump (pmd) that is created during the crash.

Figure 3-7: Restart log (Error log)

An example of the error log is given below:

```
› 8.  Time          :  11-02-14, 03:52:30
›     Error code    :  0x80010109 (Reported via CELLO:ERI IF)
›     File:line     :  osa_board_mgr.c:1783
›     Process       :  ose_cmd_reload
›     Queued signals :  0
›     Owned signals  :  1
›     OS Run Mode    :  BACKUP
›     Error Information:
›     Reload ordered by operator command
```

Figure 3-8: Error Log .. example

The list error log is useful while figuring out what is the process that led to a restart for that particular processor. In some restart, there could also be a reference to a post-mortem-dump (pmd) that is created during the crash.

When the node restart time and the board restart time do not match, then it is obvious that something led to the board to restart. This could be due to a faulty hardware, or because of someone performing the restart.

2.4 Post Mortem Dump

If a program crashes, information about the error and all the relevant information at the time of the crash is saved in the Post Mortem Dump (PMD) area, and then saved on a file.

- If a program crashes, information about the error and all the relevant information at the time of the crash is saved in the Post Mortem Dump (PMD) area, and then saved on a file.
- Log location example: /c/pmd/102/002600/0xffffff55.pmd
- Ericsson second/third line supports can decode the pmd (0xffffff55.pmd) Therefore, these post mortem dumps should be included as attachments in processor-problem related CSRs
- The dump includes, among many other information, the trace and error log, the restart log, the loaded programs in that processor, etc.
- CLI command “dump” could be used to work with the post mortem dump while it is still in the memory, and not saved as a file yet.

Figure 3-9 : Post Mortem Dump (PMD)

The PMD files are saved in the central GPB in the following directory:

```
$ ls

Directory '/c/pmd/102/002600/'

0xffffff55.pmd
```

In the example above, 002600 points out that the board is slot 26 in the Main Subrack (which is the same as the linkhandler of that board- see CLI command “lhsh”). The pmd (0xffffff55.pmd) cannot be decoded without some decoders. Therefore, these post mortem dumps should be included as attachments for Ericsson to decode. The dump includes, among many other information, the trace and error log, the restart log, the loaded programs in that processor, etc.

The CLI command “dump” could be used to work with the post mortem dump while it is still in the memory, and not saved as a file yet.

2.5 Audit Trail Log

The Audit Trail Log logs commands and MO operations entered on the node. The log also indicates if the access was made locally (serial interface) or remotely (telnet, FTP, SSH, SFTP, CORBA).

- Log successful commands and successful MO operations entered on the node.
- There are two audit trail logs:
  - Log used by the CORBA interface
    - Log Location: /c/logfiles/audit_trail/CORBA_AUDITTRAIL_LOG.xml
  - Log used for Shell interface (CLI and FTP)
    - Log Location: /c/logfiles/audit_trail/SHELL_AUDITTRAIL_LOG.xml
    - 3MB, wrap-around log
- These logs are always activated

Figure 3-10: Audit trail logs

There are two audit trail logs: one log used by the CORBA interface and one used for commands from the other interfaces.

The log is always activated and it cannot be switched off.

2.5.1 Shell Commands and FTP/SFTP Logging

The Audit Trail for shell commands and SFTP/FTP includes all successful operations. All shell commands are logged, both those that affect the node and those that just observe the node. Operations that change the file system are also logged.

An Audit Trail log record contains:

- User identity
- User terminal name, including IP address when applicable, for: shell, serial, SSH, SFTP or FTP
- Type of operation:

  COLI Access: Command, including all parameters
  File Access: File operation (creates, append or write)

The file is located physically at the following directory:

/c/logfiles/audit_trail/SHELL_AUDITTRAIL_LOG.xml

The format of the file is as follows:

```
› Example of the SHELL_AUDITTRAIL_LOG.xml
› <LogRecord number = "5602">
›   <TimeStamp>
›     <year> 2011 </year> <month> 2 </month> <day> 14 </day>
›     <hour> 21 </hour> <minute> 44 </minute> <second> 14
  </second>
›   </TimeStamp>
›   <RecordContent>
›     <User> jack </User>
›     <Termname> /telnet_147.214.151.100:3584 </Termname>
›     <Event> Coli access granted </Event>
›     <Info> echo "LOGGING INTO BOARD: 0 13" </Info>
›   </RecordContent>
› </LogRecord>
```

Figure 3-11: Audit Trail Logs- example

The file is about 3MB big and is a wrap log.

2.5.2 CORBA Audit Trail

The Audit Trail for CORBA includes only operations for modifying the MOM. In more detail, this means:

- Only operations that modify the MOM (create, delete, set, action) are logged.
- Only successfully committed operations are logged

Log records contain:

- User identity
- Type of operation (create, delete, set, action)
- MO instance (LDN)
- Attribute name (only applicable for set operations)
- Attribute value (only applicable for set operations)
- Timestamp

The file is located physically at the following directory:

/c/logfiles/audit_trail/CORBA_AUDITTRAIL_LOG.xml

2.6 Availability Log

- Availability Log is used for In Service Performance (ISP) measurements in a CPP based node to find out when the node became unoperational and operational again
- Although the ISP is usually taken into account on the node level, the log does have information about the board and programs also.
- Log Location: /c/logfiles/availability/CELLO_AVAILABILITY2_LOG.xml
- 2 MB big

Figure 3-12: Availability Log

The logging of spontaneous node restart and Complete Exchange Failure (CEF) is improved. Uptime is saved in persistent memory every second and used as time stamp for out-of-service event when the node recovers after spontaneous restart. In case of CEF and no time stamp, it's specifically stated that out-of-service time stamp is missing. Program restarts are logged as program events and partial-loss-of-service is logged when more than 20% of the node resources are unavailable.

Availability log size is increased from 1 to 2 Mb since to minimize the risk of filling the log in less than 1 month.

Below is an example of the entry in the Availability log regarding a restart.

```
<LogRecord number = "1679">
 <TimeStamp>
  <year> 2007 </year> <month> 8 </month> <day> 31 </day>
  <hour> 11 </hour> <minute> 7 </minute> <second> 57 </second>
 </TimeStamp>
 <RecordContent>
  <NodeEvent/> <OutOfService/>
  <EventReason> ShutdownCommand </EventReason>
  <AdditionalInfo>
   <CppCore>
    <RankWarm/>
    <Cause> ExtRestartRequest </Cause>
   </CppCore>
```

```
            </AdditionalInfo>
           </RecordContent>
          </LogRecord>
         ----
          <LogRecord number = "1680">
  <TimeStamp>
   <year> 2007 </year> <month> 8 </month> <day> 31 </day>
   <hour> 11 </hour> <minute> 7 </minute> <second> 57 </second>
  </TimeStamp>
  <RecordContent>
   <NodeEvent/> <OutOfService/>
   <EventReason> UnOperational </EventReason>
   <AdditionalInfo>
    <CppCore>
     <RankWarm/>
    </CppCore>
   </AdditionalInfo>
  </RecordContent>
 </LogRecord>
-----
------
 <LogRecord number = "1694">
  <TimeStamp>
   <year> 2007 </year> <month> 8 </month> <day> 31 </day>
   <hour> 11 </hour> <minute> 8 </minute> <second> 25 </second>
  </TimeStamp>
  <RecordContent>
   <NodeEvent/> <InService/>
   <EventReason> Operational </EventReason>
   <AdditionalInfo>
    RNC Node Restart Completed, 11:08:25
   </AdditionalInfo>
  </RecordContent>
 </LogRecord>
```

The availability log is located in the following file in the CPP node.

/c/logfiles/availability/CELLO_AVAILABILITY2_LOG.xml

2.7 System Log

- This is a log for the whole system in one central place
- Log Location: /c/logfiles/systemlog/
- Upto 10 files, each file 1 MB big

Figure 3-13: System Log

The system log consists of a set of files. These files are located in the `/c/logfiles/systemlog` directory. There can be a maximum of ten valid System Log files present at any one time. Each file is no larger than one megabyte in size. Each file has a unique filename. The filename consists of a prefix and a suffix. The prefix of the filename is a five-digit ASCII sequence number. The filename suffix is the string `"syslog"`. Thus, filenames range from `"00000syslog"` to `"99999syslog"`. The oldest file has the lowest prefix and the newest file has the highest prefix.

It is possible that the System Log service detected that a System Log file has been corrupted in some way. In this case, the corrupt file is present in the directory but it has been renamed by the addition of the `".corrupt"` suffix to its filename. Such corrupt files are not counted in the set of ten valid System Log files.

System Logs are located in the following directory:

/c/logfiles/systemlog/

Below are examples of System Log record which are created automatically by the System Log service:

```
› recid=18,format=STRING,event_type=0,facility=SYSLOG,u
id=0,gid=0,pid=65867,pgrp=131397,severity=INFO,time=M
ay  1 11:38:52 2011    Cancel upgrade supervision
timer,size=158


› recid=5,format=STRING,event_type=0,facility=SYSLOG,ui
d=0,gid=0,pid=65744,pgrp=131275,severity=INFO,time=Ja
n1 00:58:49 2011    Creating new crash information
envelope: initiated by: smn=00, apn=26, suua=00
reporting: ecode=0xF0F0F0F3, restartDomain=BASIC,
restartType=PROCESSOR generated:
pmdId=0x0000001a.pmd,  date=11-01-01,
time=00:58:22,size=354
```

Figure 3-14: System Log - example

recid= A System Log record identifier, an integer that uniquely identifies the record in the System Log.

The value is the ASCII representation of an unsigned integer value.

format= This element specifies the format of the variable portion of the log record.

The value is set to "STRING", to denote that the variable data consists of a single null-terminated string, or is set to "NODATA" to denote that the log record contains no variable data portion.

event_type= An identification code for the event type. This is an integer that identifies the type of event, for which the log record was generated. All System Log service clients use the default value zero.

facility= Event facility code. This identifies the part of the system that generated the log record. The value is a string. Only two strings are used :

"SYSLOG" for the system

"LOGMGMT" for the System Log Service itself

uid= Effective user identity of the process associated with the log record. The value is an integer.

gid= Effective group identity of the process associated with the process that generated the log record. This is set to zero.

pid= Process identity of the process that generated the log record. The value is the ASCII representation of an unsigned long value.

pgrp= Process group identity, to which the process that generated the log record belongs. This is the OSE process block identity and it is an integer.

severity= This specifies the severity of the event for subsequent action by the system. The value is an ASCII string. Currently only three values are used :

"CRIT". A critical condition that threatens the availability of a significant portion of the system.

"ERR". An error.

"INFO". An information message.

time= This is the time at which the event was logged. The time value is an ASCII string representation of the time. There is no comma after this string and a new line is written automatically. The variable length string begins on a new line.

size= This is the size of the log record in bytes, excluding the size element itself. The system may write an event-specific piece of information in each log record. This information is between the "time=<*time string*>," and "size=<*integer*>" strings.

2.8 Trace Log for software handling and upgrades

- Also called the Trace.log
- Log Location: /c/systemfiles/cello/cma/su/trace/Trace.log
- 2 MB (default)
- Useful while troubleshooting install and upgrade related problems

Figure 3-15: Trace log

Here is an example of the log entries from the Trace.log.

```
› Date:   2011-05-01, Time: 11:37:59.256,  SU_SW: 6.0 LSV14 - R60MA07
› Class:  UpgradePackageMoImpl
› Method: changeToState ( int aToState, boolean aSendNotification )
› Log:    Sending AVC Event - state: Upgrade executing

› Date:   2011-05-01, Time: 11:37:59.328,  SU_SW: 6.0 LSV14 - R60MA07
› Class:  UpgradePackageMoImpl
› Method: actionConfirmUpgrade ( Coordinator c )
› Log:    End Action - Confirm Upgrade, continue upgrade initiated

› Date:   2011-05-01, Time: 11:38:04.602,  SU_SW: 6.0 LSV14 - R60MA07
› Class:  UpgradePackageMoImpl
› Method: setProgressHeaderValue ( int aProgressHeader )
› Log:    Sending AVC Event - progressHeader: The upgrade phase is initiated
  and the system state is set to upgrade mode

› Date:   2011-05-01, Time: 11:38:04.741,  SU_SW: 6.0 LSV14 - R60MA07
› Class:  UpgradePackageUpgrader
› Method: run()
› Log:    The upgrade supervision timer is set to 14400 seconds and started.
```

Figure 3-16: Trace.log – example

2.9 Trace Overload Protection

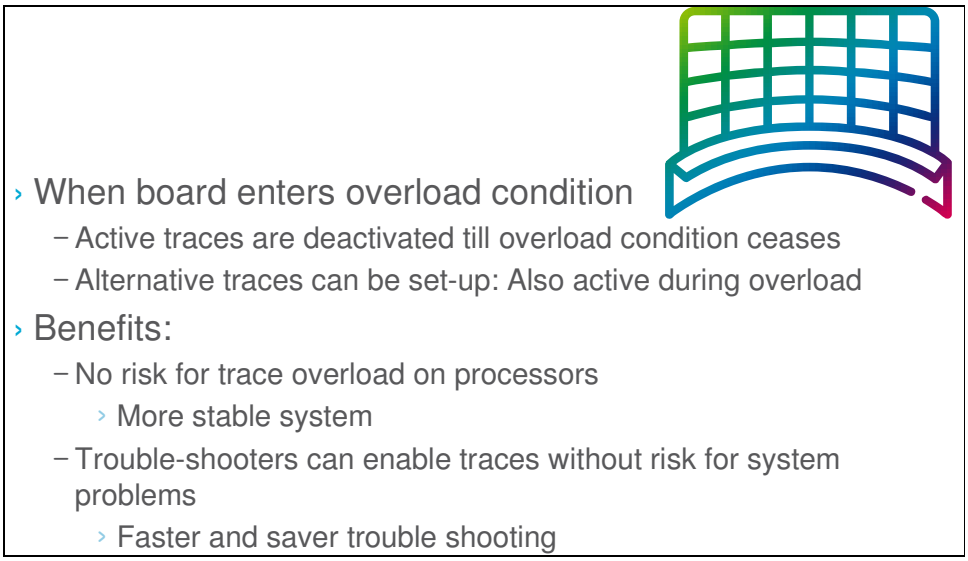


Figure 3-17: Trace Overload Protection

The Trace and Error function includes an overload protection mechanism that reduces the risk of overload, this might occur during heavy tracing. This is done using an alternate trace group setting to be used in high load situations, that is, if CPU usage exceeds 95%.

A Trace Overload Protection function shall be implemented in RNC node to prevent that Trace & Debug activities may cause severe overload problems in RNC.

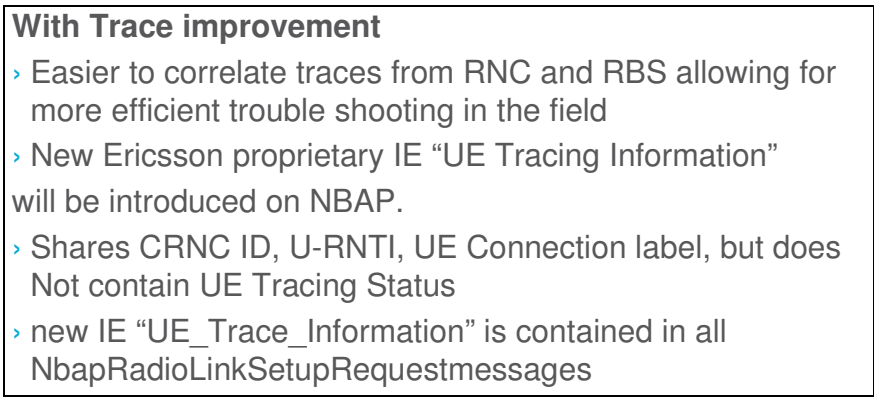


Figure 3-18: Trace Overload Protection

2.10 Node Persistent Logging

This improvement from the CPP platform and accordingly available for any CPP based application; RNC, RBS, RXI, LTE etc.

Node Persistent Logging is a new option to store Trace and Debug data on local file system. These logs can later be fetched with (S)FTP. By using the disk, loss of log entries due to overflow in Trace and debug log or due to IP communication problems can be avoided. This gives the possibility to get trace data for nodes where host transmission is slow or sometimes interrupted.

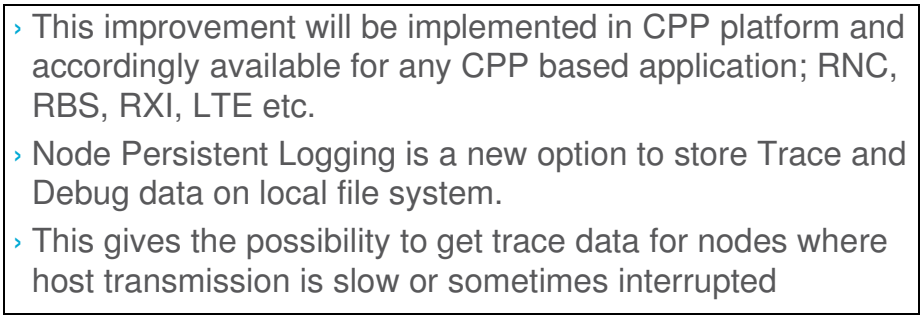


Figure 3-19: Node Persistent Logging

2.11 Hardware Inventory Log

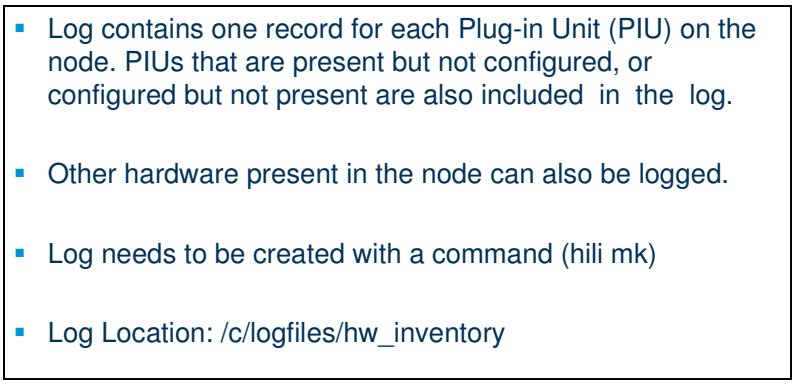


Figure 3-20: Hardware Inventory Log

Once created, the command ‘cat’ can be used to read the log, or downloaded to an external machine with ftp/sftp.

Below is an example of how the hardware inventory log could look like.

```
<LogRecord number = "25">
  <RecordContent>
    <PlugInUnitInformation/>
    <PiuType> Bp </PiuType>
    <PiuAddress>
      <SwitchModuleNumber>  0 </SwitchModuleNumber>
      <SwitchPortNumber>   26 </SwitchPortNumber>
    </PiuAddress>
    <UnitHwPid>
      <ProdNo> ROJ1192232/1</ProdNo> <ProdRev> R2C </ProdRev>
      <ProdName> ET-MFG </ProdName> <ProdDate> 20101127</ProdDate>
      <ProdSerialNo> TU8B268404 </ProdSerialNo>
    </UnitHwPid>
    <ConfiguredHwPid>
      <ProdRev>  R2 </ProdRev>
    </ConfiguredHwPid>
    <UnitTestStatusOk/>
    <UnitTestResult>
       2011-05-01 11:41:35 Others   Tests OK BHRO  TEMI 1:Restart bh  =OK  All hw tests
passed
    </UnitTestResult>
  </RecordContent>
 </LogRecord>
```

Figure 3-21: Hardware Inventory Log - example

2.12 Trace and Error LOG

The Trace support on CPP based nodes is implemented by the trace part of T&E package.

- Used to manage the tracing and recording function in CPP
- Used to record trace events in processes on MPs, BPs and SPs.
- Each processor in the node has its own trace log
- Wrap-around log saved in the volatile memory that survive restart but not power-off

Figure 3-22: Trace and Error function

The trace log resides in volatile memory, and survives a restart of the system, but not a power off. The T&E log wraps around when the log filled and the oldest stored log entry is overwritten with the most recently stored log entry.

The CLI command “te” is used to mange tracing. Here is a list of te command options in a CPP node.

```
Synopsis

        te <subcmd> <operands>

Subcommands

config :Configure saved traced groups.

default: Set trace groups to default.

disable: Disable trace groups.

disk: Log to disk.

enable: Enable trace groups.

filter: Set and reset trace bus filter.

log: Trace and error logging.

monitor: Enable or disable log monitoring.

preset: Preset trace groups during interception

save: Save current enabled trace groups.

status: Display trace group status.
```

- Working with the trace and error funtion:
  - te <subcommand> <operand>

Subcommands:	Operands:
config	Subcommand dependant
default	
disable	
disk	
enable	
filter	
log	
monitor	
preset	
save	
status	

Figure 3-23: Trace and Error

Among these subcommands of te, the “te default”, te disable”, “te enable”, “te log” and “te status” are used the most.

2.12.1 Trace and Error Log

The trace and error log (te log) has the following operands:

```
te log <operand>

    Operands:
    clear
    freeze [-grp <group>] [<message> [<count>]]
    read [-rel | -mon] [<time>]
    resume
```

Figure 3-24: Trace and error log

Below the different operands are explained further.

`te log clear:`

This subcommand clears the Trace and Error log. All currently stored log entries are removed.

`te log freeze [-grp <group>] [<message> [<count>]]:`

This subcommand freezes the Trace and Error log. A frozen log disables any further logging. If no option is specified, the log is frozen immediately.

Settings made with the command, te log freeze do not survive a board restart.

`grp is specified:`

The log is frozen when the trace event matches the group.
Can be one of the following:
check, error, enter, return, info, trace1 to trace9 , state_change, bus_send, bus_receive, rec_sig, send_sig, param, user1 to user4.

`message is specified,`

The message will be matched with every logged message. If a match is found, the <count> operand controls the number of additional messages that are logged when a matching freeze message has been found. If no <count> is specified, zero is assumed and the log is frozen immediately when the freeze <message> is matched. The match is performed as a substring match, that is, the freeze <message> can be present anywhere in the logged message.

te log read [-rel | -mon] [<time>] :

This subcommand reads the Trace and Error log. During the reading of the log, all logging is disabled, that is, the log is frozen.

mon Dump

The log to the monitor instead of to the shell itself. A monitor must be connected when using this option.

NOTE: The -rel option cannot be used in combination with the mon option, as the monitor itself controls the format of the timestamp.

rel

Present the timestamps for the log entries, using the time relative to when the node was started. If this option is not specified, the default is to display the timestamps using absolute time.

<time>:

The maximum age, in seconds, of the log entries to be read.

te log resume:

This subcommand immediately resumes the logging in the Trace and Error log. It also clears not yet activated freeze commands.

Since trace and error logs wrap around, it overwrites. For this reason, and to keep the processor load to a minimum, only a limited number of traces are enabled by Ericsson.

Among these operands, the “te log read” is used the most, as this reads the events recorded in the trace log. The log layout is shown below:

[1997-10-02 15:31:54.510] *OMCSF_teTest2 test.c:619*
TRACE2:The board has been loaded

[2007-06-28 08:00:37.832] *Act_aal0Server_proc aal0ceprofactory.c:672* **INFO:A50I Initialized**

The trace and error has four fields: when , how (although usually missing) , *where* and **what** fields.

Figure 3-25: te log read- example

When: time when the entry was made

How: This field is usually empty, but might have some entry (like * and L) if the log was frozen or dumped to a monitor at the time of the entry. This is not shown in the example above.

Where: The name of the process that has generated the log, together with the file name and the line number

What: It consists of the Trace Group to which the log entry belongs and the actual logged message.

TRACE GROUPS

A trace group is an identity used by T&E macros. Events to trace are found within the process code and traced by means of "Trace macros". A macro is responsible for logging of one event, which is given as text strings.

- Trace groups define the events to trace and record in the trace and error logs.
- Trace Groups include the following:
  CHECK, ERROR, ENTER, RETURN, INFO, REC_SIG, SEND_SIG, TRACE1, TRACE2...TRACE9 STATE_CHANGE, BUS_SEND, BUS_RECEIVE, SEND_SIG, PARAM, INTERFACE, OBJECT, USER1 ..USER4

Figure 3-26: Trace Groups

It is beyond the scope of this training to describe what each of these trace groups mean.

2.12.2 Enabling and Disabling Traces

Note that it is NOT advised to start traces without understanding what the traces do, and what their impacts have on the system.

- ”Enable traces”:
  Format: te enable ”tracegroup” ”processName”
  Example: te enable trace1 RE*
  te enable all NBAP_ASN
- ”Disable traces”:
  Format: te disable ”tracegroup” ”processName”
  Example: te disable trace1 RE*
  te disable all NBAP_ASN
- Put all the traces to the default state:
  Format: te default

Figure 3-27: Enabling and disabling traces

Although there are other formats of the command, the most common format to enable trace is the following:

```
te enable <Trace Group> ... <item>
```

The <Trace Group> can be one of the following, although the first five in the list are always enabled by default: check, error, enter, return, info, trace1, trace2,.. trace9, state_change, bus_send, bus_receive, rec_sig, send_sig, param, user1, ..user4

The <item> is whose trace groups should be enabled. It is possible to specify more than one item by using a wildcard (*) at the end of the name.

```
te enable trace1 RE*
```

In the command above, the trace group ‘trace1’ is enabled in all the processes whose name start with RE. Once it is enabled, the ‘trace1’ log from those processes will be visible in the command ‘te log read’.

To disable traces that were started for some tests/troubleshooting purposes, the following command could be used:

```
te disable trace1 RE*
```

However, if all the traces need to be reset to their default values, then the command '`te default`' could be executed.

2.12.3 Monitoring Trace and Error Log

- **te log read – reads the trace and error log from a board**
- **To continuously monitor the trace and error log, then a monitor program needs to be run**
- **Target Monitor is the convenient way to monitor te log output realtime, and from a number of boards:**
  Format: tm <–option> <parameter>

  Example of the most common use is:
  CLI> tm –tcp –win 1 =>gives "handle#" and 'port#'
  CLI> tm –status
  CLI> tm –attach "handle#" xxyy00 (xxyy00 is the board-position)

  DOS/PC> telnet IPAddress "port#"

Figure 3-28: Monitoring the traces

While it is possible to use the trace and error command (te log read with the –mon operand or the command "te monitor"), it is easier to use the command "tm". The tm stands for target monitor, and allows the user to monitor te log read from more than one processor (=link-handlers) at the same time.

```
tm  -option <parameter>
```

The -option and <parameter> combination could be:

```
 -abs<handle> Set absolute time timestamp for all logs
attached to this monitor.

 -attach <handle> <linkhandler1> [...linkhandlerN]
Attach a list of linkhandlers to this monitor, N <= 50.

 -channel    N/A Print current channel type.

 -close <handle1> [ handleN] Close the specified
monitor(s) N<= 10

 -detach <linkhandler1> [...linkhandlerN] Detach the list
of linkhandlers from their respective monitors.
```


```
 -disconnect N/A Disconnects everything.

 -mode <load|trace> Modifies the priority of all monitors
to favour traffic load (low monitor priority) or traces
(high monitor priority).

 -pm [on | off | list]  Start/stop/display PM-counters.

 -rel <handle>    Set relative time timestamp for all
logs attached to this monitor.

 -rm N/A   Delete a stored configuration

 -save  N/A    Save the current monitor configuration.

 -status  N/A    Print current configuration.

 -tcp  -win <X>     Select TCP-mode; sub-option -win is
used to specify how many TCP-monitors should be opened. X
<= 9.
                  Note that a syslog monitor will be added
automatically if not already active, bringing the total
number of TCP-monitors up to 10

 -udp <IP-address>    Select UDP-mode, the IP-address to
specify is typically your workstation IP-address.
```


3 SUMMARY

After this chapter the participants should be able to:

3 Investigate the purpose and the location of the various types of logs in a CPP based node

3.1 Know the location and purpose and read Alarm and Event logs.

3.2 Explain how Ericsson Local Support enables traces in the process of troubleshooting, and uses the target monitor application to capture the traces.

3.3 Find out the location and purpose of Error Log, Post Mortem Dump(PMD) Log and Availability Log.

3.4 Find out the purpose and location of the Security and Audit trail logs.

3.5 Perform data collection to include in the Customer Service Request (CSR) when a problem is suspected in the WRAN network.

3.6 Explain Trace improvement, UE-ID sent to RBS in all Radio link Setup messages.

3.7 Describe Trace Overload Protection and RNC Throughput Capacity.

3.8 Describe RBS MP Load sharing between DUW's.

Figure 3-29: Summary Chapter 3

4 Performance Management Overview

Objectives

After this chapter the participants will be able to:

4 Be able to tie together the Performance Statistics and Performance Monitoring in the process of troubleshooting in WRAN Network.

4.1 Be able to initiate statistics from the OSS-RC.

4.2 Be able to initiate performance recording (e.g. UETR) from the OSS-RC.

4.3 Be able to read counter values and tie them to the situations in the network.

4.4 Explain Enhanced Health Check Filter File.

4.5 Genrate some faults and Show how to step by step resolve the faults.

Figure 4-1: Objectives of Chapter 4

Intentionally Blank

1 OVERVIEW

Performance Data Analysis and Fault Management procedures are the two main components of the troubleshooting procedure.

Chapter 2 looked into the fault management principles and the tools associated with them.

This chapter looks into Performance Management principles and applications in Ericsson's WCDMA RAN. Later in the chapter, the individual applications in the OSS-RC and the Element Management are also listed.

2 PERFORMANCE MANAGEMENT in WRAN

WCDMA RAN generates performance data in the Network Elements (NEs). The performance data consists of PM statistics known as PM counters, PM recordings: IMSI/ and or Cell filtered recordings, and General Performance Event Handling: GPEH performance events. The performance data is collected regularly by OSS-RC. This performance data can either be converted to a database format to be stored persistently if ENIQ is available or exported to an external management system NMS or WCDMA RAN external tools via OSS-RC.

Performance Management provides data on the WCDMA RAN performance with respect to accessibility, retainability, and integrity. WCDMA RAN has several Performance Management applications that gather and process performance data. This data can be used to monitor key Quality of Service (QoS) indicators, optimize network performance, identify trends, and troubleshoot problems in WCDMA RAN.

The basic performance applications are Performance Statistics, Performance Recording and General Performance Event Handling (GPEH). These applications have GUIs that are accessed using OSS-RC.

The applications for Performance Management are physically located in OSS-RC except for Ericsson Network IQ that is running on a separate server. The node-level applications are physically stored in each node RNC, RXI, and RBS. The applications for Performance Management functionality can be accessed from anywhere in the O&M Internet.

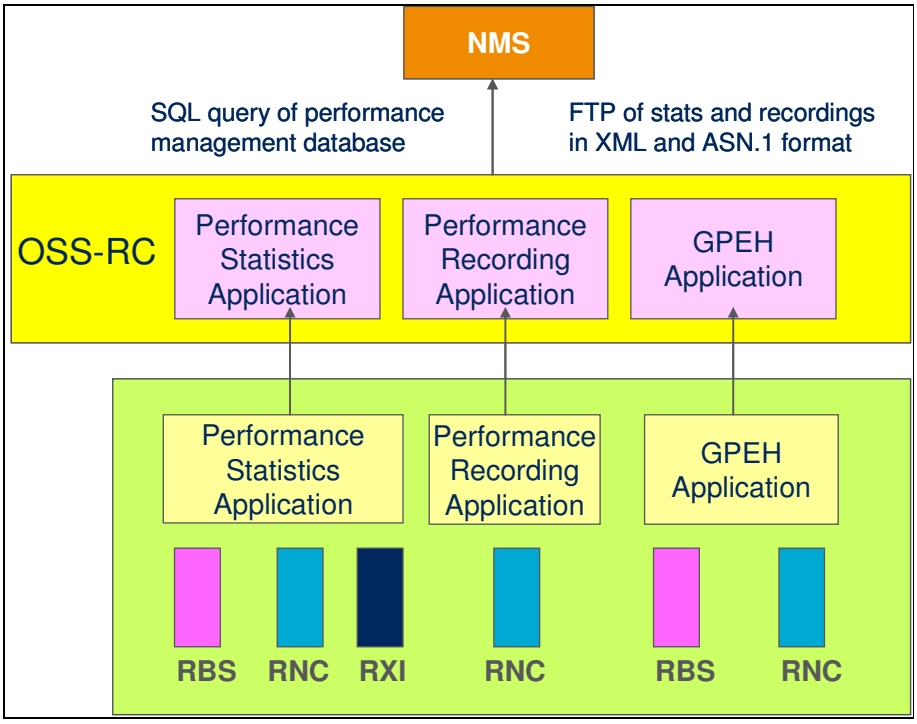


Figure 4-2: Performance Management Applications

2.1 Performance Statistics

The performance statistics are generated from measurements on the radio and the transport network's live traffic. The Performance Statistics data is made up from a number of predefined counters.

It is possible to subscribe to each radio or transport network counter through the OSS-RC Graphical User Interface (GUI). By activating a counter, measurements are started. The values are periodically collected and stored in the OSS-RC.

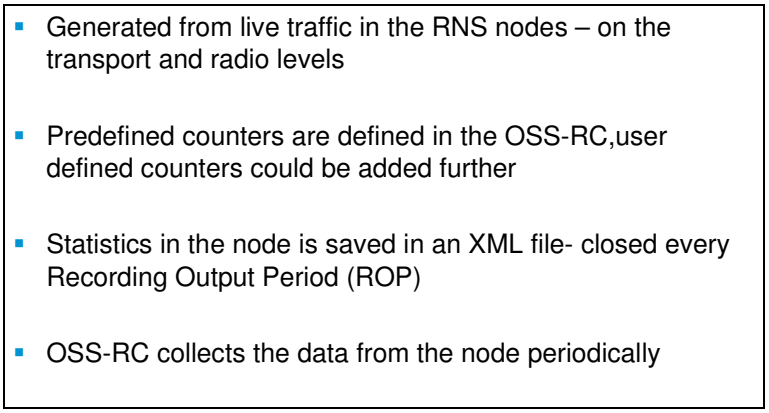


Figure 4-3: Performance Statistics..1.

The statistical data is fetched by OSS-RC from the network elements every Result Output Period (ROP) on a file in XML format. The ROP file is fetched every 15 minutes. The ROP file is automatically compressed using GZIP. Once OSS-RC has retrieved the result, it saves the data for 3 days (default value), unless some other value is configured. For added RBSs, statistics are automatically started in the RBS.

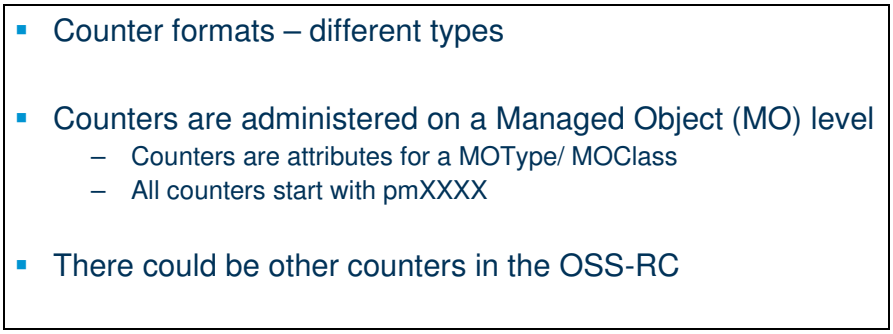


Figure 4-4: Performance Statistics..2

2.1.1 Counter Formats

There are seven types of counters available in the system:

- `Peg counter` is a counter that is increased by 1 at each occurrence of a specific activity.
- `Gauge counter` is a counter that may be increased or decremented depending on the activity in the system.
- `Accumulator counter` is a counter that is increased by the value of a sample. It indicates the total sum of all sample values taken during a certain time. The name of an accumulator counter begins either with pmSum or pmSumOfSamp
- `Scan counter` is a counter that is increased by 1 each time the corresponding accumulator counter is increased. It indicates how many samples of a certain quantity have been taken. A scan counter can therefore be considered a specific kind of peg counter. Due to these types of counters, it is possible to get the average value of all samples by dividing the accumulator counter by the scan counter. The name of a scan counter begins with `pmSamples`
- `TrigACC counter` is a counter that is increased by the value of a sample and the sampling is only done when there is some activity. It indicates the total sum of all sample values taken during a certain time. The sampling is only done if there is some activity ongoing for the measured entity. The name of an TrigAcc counter begins either with `pmSumAct`
- `TrigSCAN counter` is a counter that is increased by 1 each time the corresponding TrigAcc counter is increased. It indicates how many samples have been read, and added to the related TrigAcc counter. The sampling is only done if there is some activity ongoing for the measured entity. A TrigScan counter can therefore be considered a specific kind of peg counter. Due to these types of counters, it is possible to get the average value of all samples by dividing the TrigAcc counter by the trigScan counter. The name of a TrigScan counter begins with `pmSamplesAct`
- `TrigSQR counter` is a counter that is increased by a square value of a sample and the sampling is only done when there is some activity. It indicates the total sum of square sample values taken during certain time. The sampling is only done if there is some activity ongoing for the measured entity. The name of the TrigSQR counter begins with pmSumSqr.

TrigACC and TrigSCAN counter pairs define general purpose counters for averaging of any type of value. Values are added to the TrigACC counter when a defined trigger occurs and the number of occurrences of the trigger is held in the corresponding TrigSCAN counter. An average value can be calculated by TrigACC / TrigSCAN. A trigger can be anything, for example, an incoming 3GPP message, an internal system event or similar, and is defined for each counter pair

TrigSQR works together with TrigACC and TrigSCAN counter pairs, but not all TrigACC and TrigSCAN pairs have also TrigSQR type

- `PDF measurements` are a list of range counters. A value is sampled (read) periodically. If the value falls within a certain range, the range counter for that range is incremented. All range counter values are collected and stored in a ROP file at the end of each reporting period. All range counter values are collected and stored in a ROP file at the end of each reporting period.
  For example, if SIR values are split into three ranges: Range1 = [-11 dB..-4dB], Range2 = [-4dB..+4dB], Range3 = [+4dB..+20dB] and a value is read every 3 minutes over a 15 minute period (values = -10, -3, +5, +5, +6), then the three RangeCounters will be reported as RangeCounter1 = 1, RangeCounter2 = 1, RangeCounter3 = 3.
- `Discrete distributed measurements` are a series of values recorded during a reporting period. Each series of values may be of one of the following measurement types:

Accumulated over a measurement period and read at the end of each measurement period (a gauge or peg counter)

Averaged over the duration of a measurement period

Read at a specific time (the measurement time), within the measurement period (at a specific frame)

At the end of a series of consecutive measurement periods (the reporting period) all measurement values are collected and stored in a ROP file.

For example, if a SIR value is read every 3 minutes over a 15-minute period (values = -10, -3, +5, +5, +6), then 5 DDM measurements are reported as Meas1 = -10, Meas2 = -3, Meas3 = 5, Meas4 = 5, Meas5 = 6.

- `Calculated statistics counter` is a counter whose value is determined by other counters. This is performed in the OSS-RC Statistics database. The ROP files are opened in order to be transferred into the database and the calculations are done by the database itself during this process. This means that these counters are not available when the Statistics Database is not present.

Note:

The names of all the counters created in NEs start with `pm`, while the names of the OSS-RC calculated statistics counters start with `cm`

2.2 Counter Classification

There are two general classification of the statistic counters. First, they can be grouped depending on where they are generated, that is at which NE:

- RNC counters
- RBS counters
- RXI counters
- OSS-RC counters - computed counters (cm) only available with Ericsson Network IQ (ENIQ).
- OSS-RC EBS counters - created by using the GPEH event recording.

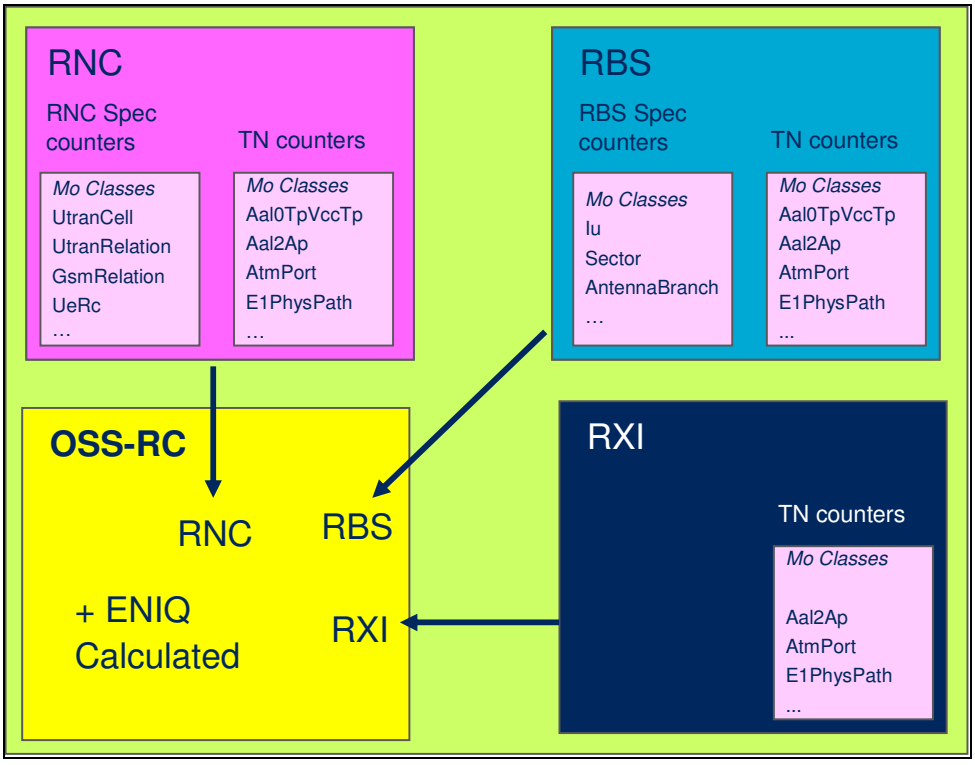


Figure 4-5: Counters grouped in OSS-RC based on their origin

2.3 Counter Limitations

There are limitations on the total amount of counter instances that can be simultaneously active in each NE. (Numbers presented refers to counter values, e.g. a peg counter is counted as one, while a PDF counter contributes with one for each range.)

- RNC
  - 1 700 000 counters with R4 HW or later on the O&M board.
  - 72 000 counter instances active in one scanner activation
  - 216 000 simultaneous counter instances active.

Note that these limits apply when up to 15 scanners are running on the RNC, including the two predefined scanners. With more scanners active, the capacity is lower

- RBS - 22,000 counters
- RXI - 65,000 counters

Before activating statistics profiles it is recommended to check if the number of counters would exceed the limits for each NE.

2.4 Measurement Administration

OSS-RC acts as a centralized point for the initiation and collection of performance data for the WCDMA RAN. The nodes provide a GUI (Graphical User Interface) and as also a machine-machine interface allowing OSS-RC to administer the statistics. The Performance Management function, allocated in OSS-RC, allows an OSS-RC operator to create Subscription Profiles and subscribe to Performance Monitoring and through GUI subscribe to performance recordings.

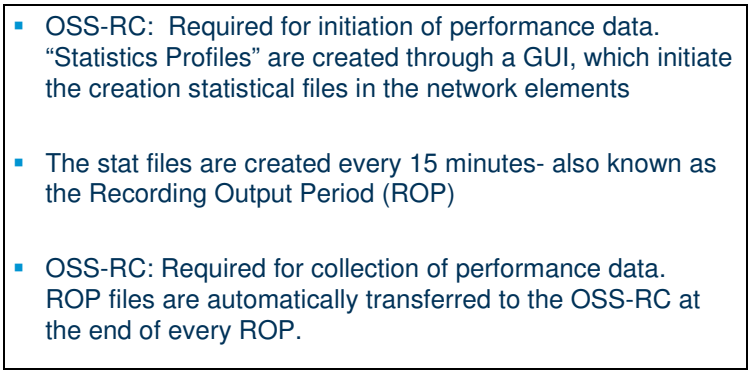


Figure 4-6: Measurement administration – overview

As a result of the collection one statistic file will be generated per RNC and one statistics file will be generated per RBS. Collected data files remain in OSS-RC for 3 days for statistical files before being deleted by OSS-RC. These values can be changed by the modifying parameters `statisticsFileStorageDays`. The minimum value is 24 hours and the maximum is limited only by available hard drive disk space. All ROP files are automatically fetched after the end of every ROP.

2.5 Counter Activation

In order to monitor the statistics counter values throughout the time, specific counters have to be *active*. Only when a counter is active values are generated, collected and can be analyzed.

In this chapter, an overview is presented of the counter activation process using the OSS-RC GUI.

2.6 Statistics Profiles

For the activation of one or more counters the user has to define a *statistics profile*. A statistics profile is an entity in the OSS-RC GUI that helps users to manipulate counter administration.

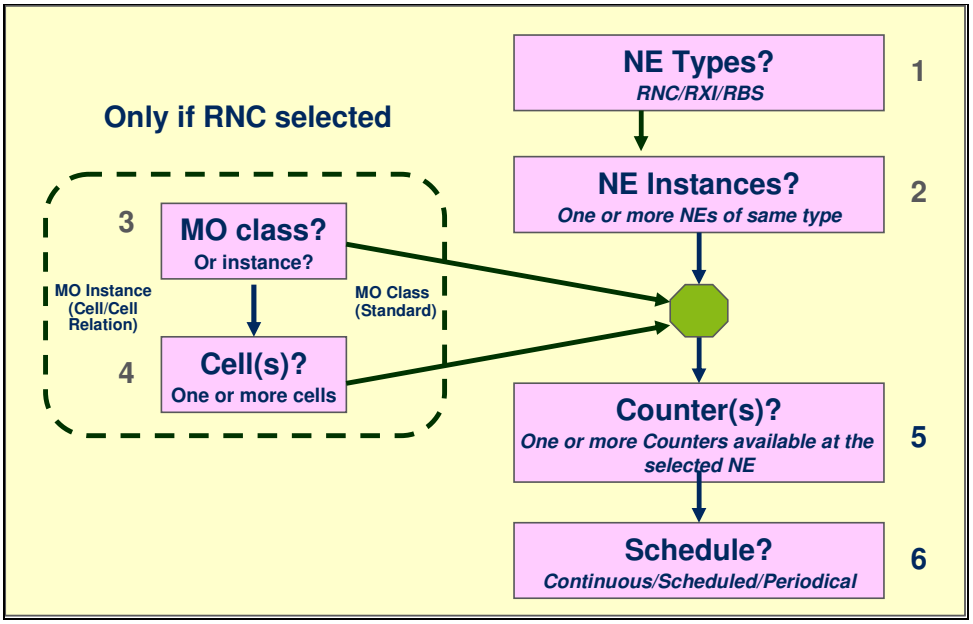


Figure 4-7: Statistics profile creation

One profile is typically defined in the steps shown corresponding to the figure above:

4 NE type (RNC, RBS or RXI)

5 One or more NE instances

It is possible to include one, more or all NE instances into a profile. All the actions performed at the profile will be automatically performed for all the selected NE. Note that if a new NE is setup in the RAN, it will not automatically be included in the existing user-defined profiles. In order to update such a profile it would be necessary to manually suspend it, include the new NE and then resume it.

6 MO Class (Standard) or MO Instance type (Cell/Cell Relation, VclTp, GsmRelation, Hsdsch, Rach)

The MO Class is normally selected, that is all the existing MO Instances are wanted for a statistics profile.

In addition to this it is possible to select only specific cells to be performance monitored. This choice is available for Utran Cell, Hsdsch, Rach, Utran Relation and Gsm Relation counters and for Transport Network link MO's: VclTp, Aal0TpVccTp, Aal1TpVccTp, Aal5TpVccTp and Aal2PathVccT.

7 MO instance selection

If the MO Instance type is selected for a profile in the previous step, it is possible to select the wanted cells or links. Only for these cells, the counters (selected in the next step) will be possible to handle through the profile. Note that the MO instance selection is always done on the cell level by the user. The selection of Hsdsch, Rach, Utran Relation and Gsm Relation is done automatically based on the cell selection.

8 One or more selected counters (only the counters available to the NE type of the profile)

Note that for RNC specific and Transport Network counters it is possible to include and activate more than once the same counter for the same NE for the same MO Instance. This would result in duplicating the data in the statistics ROP files, causing unnecessary increase of their size. It is highly recommended that each counter is activated only once, that is through only one profile. PMS GUI will warn the operator about duplicated counter subscriptions but it will not prevent users to duplicate counters.
For RBS specific counters this is not valid - duplicated subscriptions to counters already included in other measurements, active or suspended, will fail.

9 Scheduling

A user-defined statistics profile can be defined as:

- Continuous - the profile is always on and state changes is only performed by the user
- Scheduled - the profile starts at a specified point in time and stops after a specified duration
- Periodic - the profile starts at a specified point in time, stops after a specified duration. This is repeated a specified number of times in regular intervals.

Example: The profile "XYZ" is an RNC type profile of type Standard (MO Class) for the RNC01 and RNC02 and is including the counters `pmCellDowntimeAuto` and `pmCellDowntimeMan`. It is of type continuous.

Once created, the profile may be *resumed* (state changed to active) or *suspended* (state changed to suspended).

It can also be modified, but only while in state suspended.

After activating the profile, its status and details can be checked in the OSS GUI.

2.7 User-defined and Pre-defined Profiles

When a profile is created by a user in OSS-RC it is called *user-defined profile*. As opposed to user-defined there are *pre-defined profiles* which are available immediately after the startup of new NEs. They exist on the RNC and RBS, but not on the RXI. *Pre-defined profiles* don't have to be specially created and they cannot be modified.

2.8 Statistics Scanners

Normally, users need to know only about profiles and counters in order to use performance statistics. The statistics profiles are only visible in the OSS-RC GUI. The actual communication between OSS-RC and NEs is performed on another level - through *statistics scanners*. A statistics scanner is a sort of instance of a specific profile at a specific NE.

The figure below shows the concept of profiles, scanners and counters. RNC node is used as example (same applies to RBS and RXI nodes).

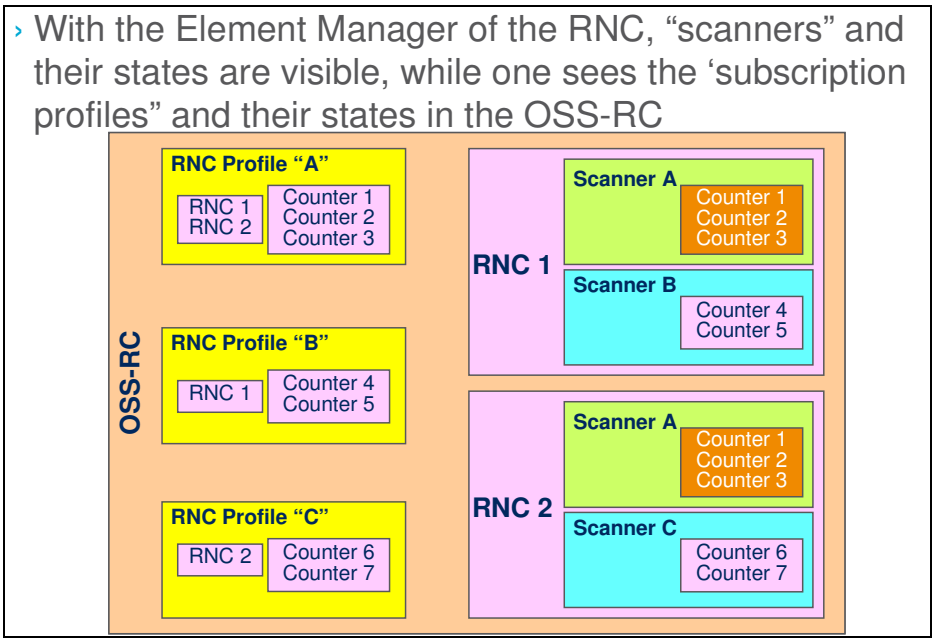


Figure 4-8: Profiles and Scanners

One profile (for example the RNC Profile "A") may be mapped to a corresponding scanner ("A") at several NE instances (RNC1 and RNC2). Another profile (RNC Profile "B") may be mapped to only one NE instance, and so on.

As mentioned in the previous chapter profiles can be active or suspended. When active, all corresponding scanners at selected NEs are also active. When a profile is inactive, all its scanners are also inactive.

Normally, the users handle all the performance statistics administration through profiles. For troubleshooting purposes, or other special cases, it is also possible to resume, suspend and monitor each scanner individually through the OSS-RC GUI and EM. Note that through the EM it is not possible to resume a scanner, one can only list and stop.

2.9 Counter Collection

Once the wanted profiles and counters are activated, their values are generated and being collected. When they get suspended, the generation and collection stop.

The overview of the process of counter collection and analysis is shown in the figure below.

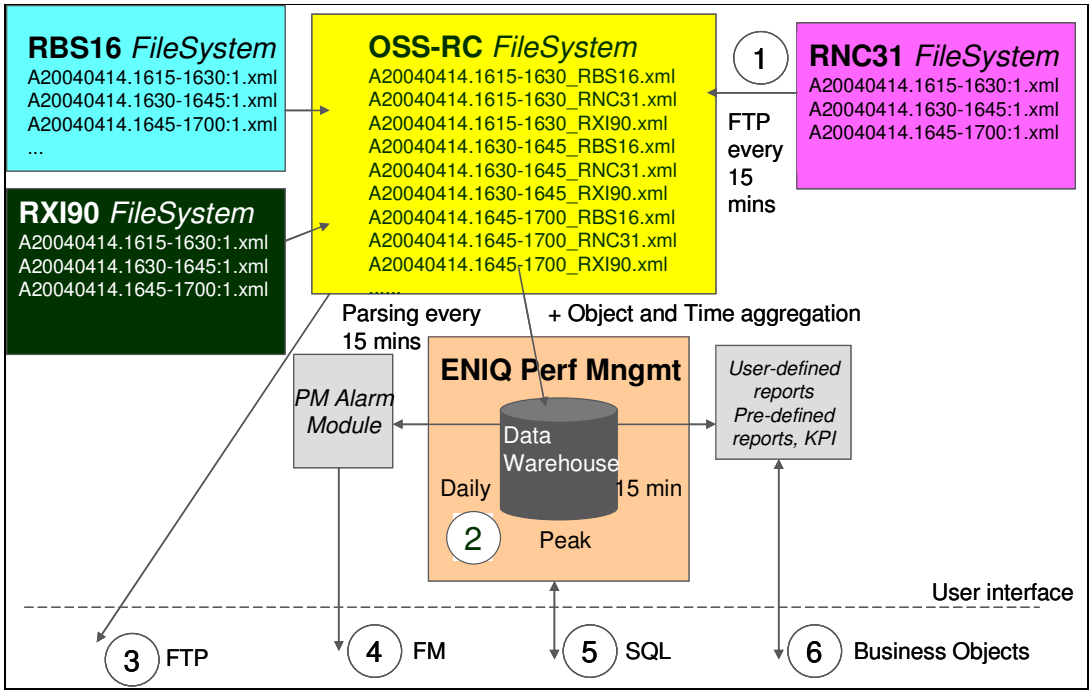


Figure 4-9: Counter collection and analysis

When a counter is active, a value is generated every Result Output Period (ROP). At the end of each ROP, all the counter values produced during that period are stored in one ROP file at the NE. This file is in XML format and compressed using GZIP. The ROP period supported is 15 minutes.

The PM statistics file has the File Format Version (FFV) tag set to Release 6, as specified in TS32.410 Rel 6.2.0. The PM file has also a new Network Element Software (NESW) tag included in the file and it is set to the SW version of the node.

The statistics ROP files are then collected (1) by the OSS-RC and stored, un-compressed, in its file system for a configurable period of time (these values can be changed by modifying OSS-RC parameters. By default the PM statistics files are stored three days on the OSS file system. If Ericsson Network IQ (optional) is present, the raw counter data can be stored for up to three months and aggregated data for up to three years.

PM statistics file names support the Local NE time as per 3GPP File Naming rules. The file name is subject to configuration in OSS, i.e. the function can be switched on or off in OSS. Default value is on, i.e. Local Time is included in the file name. Local time information in the NE file name is based on the Time Zone information allocated to the NE in the ARNE application in OSS adjusted by the daylight saving information from the OSS Master Server

2.10 Recovery Behavior of Statistics Scanners

Predefined scanners survive a node restart and will automatically come up again after the restart. User-defined scanners will survive a restart if they have been saved in a *CV back-up*.

Both predefined and user defined scanners will automatically recover and come up after system upgrade. This is also valid for when one/few counters have been removed from already defined scanners during system upgrade.

Two exceptions are defined during upgrade. For measurements on:

- MO instance level where an instance is removed during upgrade
- MO class level where all instances of the MO class are removed during upgrade

2.11 EBS-W, Event Based Statistics WCDMA

Event Based Statistics for WCDMA RAN (EBS-W) is an optional feature and part of Event Based Applications (EBA) in OSS-RC. It provides the possibility to create counters outside the Radio Network Controller (RNC) allowing for more flexible counter creation satisfying specific user need.

The EBS-W feature is the same feature as the EBS-GSM but here, it is not based on event streaming from the RNC. It is instead using the GPEH recording ROP files provided every 15 minutes. EBS-W is used to subscribe to the GPEH events, via the Performance Management Subsystem (PMS) interface, and to create counters based on the events in the ROP file. Based on the measurement set-up, the data is post-processed and exported to an XML file. A set of pre-defined counters based on selected Key Performance Indicators together with the option to create own counters (User-Defined counters) are part of the EBS-W concept in WCDMA RAN.

2.12 Observability in Ericsson WRAN

Observability covers all functions in WRAN that serves to monitor and analyze the performance and characteristics of the WRAN system.

The observability model shows different levels of observability targeting different purposes:

- The **Key Performance Indicators** represents the End-user perception of a network on a macro level and are of typical interest for top-level management as well as others within an operator. These numbers are typically used to benchmark networks against each other and to detect areas of problem. The KPIs are mainly calculated from PM counters based on well defined RANAP and RRC procedures. The reliability, granularity and accuracy of this data are critical and data is collected continuously.
- The **Performance Indicators** level represents information on a system level that does not explicitly qualifies in the macro level end-user perspective model, but can indicate whether the system performs good or bad. Performance Indicators do not necessarily give enough details to allow a full detailed troubleshooting. The data can also be used for planning and dimensioning. This data, typically PM counters and also RES counters, is collected continuously.
- The **Procedure level** observability represents deeper troubleshooting and measure system characteristics measurements. It involves measurement on signaling and procedure/function levels to investigate problems detected on higher levels. The amount of data on this level is enormous and these measurements are generally user-

initiated for a specific purpose and area of the network, thereby limiting the scope of the measurements. The typical source for this data is GPEH events and the recording functions UETR and CTR.

2.13 Performance Recording

Performance Recording applications are used to collect performance events and radio related measurements used for detailed network performance analysis, troubleshooting and network optimization. RAN performance monitoring is crucial for different user categories but in particular optimization engineers. They are the main users of the Performance Recording data ensuring the best operational conditions in the network. Their typical tasks include:

- Detecting performance degradations in the network, enabling the operator to take actions preserving high traffic accessibility and quality.
- Troubleshooting the network and providing solutions for potential improvements.
- Monitoring and optimizing the network performance to improve subscriber-perceived quality, or better utilization of the installed resources.
- Providing network planners with detailed information needed for dimensioning and configuration obtaining optimal network operation.

The recording files (ROP files) are collected and stored in the OSS-RC local file directory.

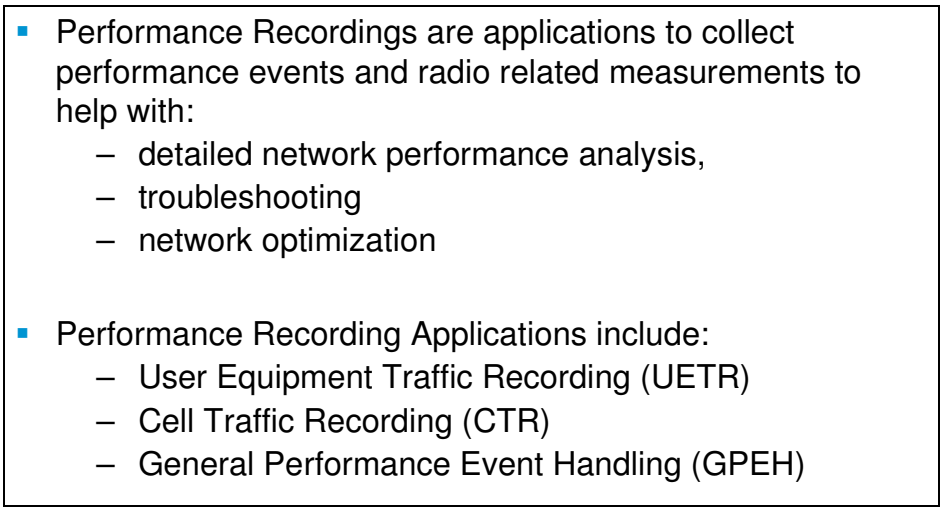


Figure 4-10: Performance recordings

Three different types of performance recordings are used- UETR, the CTR and the GPEH:

- UE traffic recording (UETR)

A recording of selected performance events, and/or radio related measurements, covering specific, operator-selected UE - using the IMSI.

- Cell Traffic Recording (CTR)

A recording of selected performance events, and/or radio related measurements, covering specific cell, usually referred to as the recording area.

These applications are based on a set of pre-defined events and periodic measurements. Each recording collects events and radio environment measurements according to the recording profile as defined by the user. The main difference between CTR and UETR is that in UETR it is the operator who decides which User Equipments (UEs) to record, while in CTR any UE for a selected cell may be recorded. The RNC adds each received measurement and event into the CTR or UETR file (depending on which one is activated). The recordings are then accumulated into files for the duration of Report Output Period (ROP). The ROP duration is 15 minutes.

The PM recording data/ ROP file is encoded in ASN.1 and converted to bit-packed binary format. The ROP file is also compressed using GZIP compression method and stored by the NE for at least 1 hour. PM recording file is not automatically decompressed when fetched by OSS-RC, it is by default stored as compressed file to save OSS-RC storage space.

2.14 UETR

- A selected UE (IMSI) is traced during live traffic to investigate signalling and UE related measurements
- Up to 16 simultaneous UETR recordings can run in parallel for one RNC.
- Possible to choose what protocol groups need to be recorded
  - Node B Application Part (NBAP),
  - Radio Access Network Application Part (RANAP),
  - Radio Network Subsystem Application Part (RNSAP) and
  - Radio Resource Control (RRC).

Figure 4-11: User Equipment Traffic Recording (UETR)

The UETR function enables the operator to record important events and measurements for a selected UE, traveling through a network. Only one UE can be selected for recording per UETR, but up to 16 simultaneous UETR recordings can run in parallel for one RNC. The selected UEs are identified by the operator using the UE's International Mobile Subscriber Identity (IMSI) number.

The operator can send out a test mobile (in particular, after changes or extensions to the network) or record a UE conducting live traffic to investigate network performance a specific area. Typically recording UE related signaling, and UE related measurement data either provided by the RBS or the UE itself.

The user may choose to record one or more of the messages within one or more of the following protocol groups: Node B Application Part (NBAP), Radio Access Network Application Part (RANAP), Radio Network Subsystem Application Part (RNSAP) and Radio Resource Control (RRC).

With this function the operator can monitor specific information that is sent to and from the UEs. For example, the operator can monitor signaling data that is used to make handover decisions. This enables the operator to identify parameters that need to be adjusted, leading to improved performance.

Although the UETR function allows for initiation of multiple UETR recordings for the same IMSI, this increases the UE load that in severe cases could result in increased dropped calls for the traced UE.

2.14.1 CTR

- To collect data for a number of UE connections within a certain recording area, which is a specific cell in which the UEs are going to be followed.
- Chosen triggering events in the selected Access Cell initiate the recording. RRC (default), NBAP, RANAP or RNSAP can be chosen
- RNC Event data, and measurement data from the RBS and UE may be selected for recording. The user can choose to record one or more messages within of the following protocol groups: NBAP, RANAP, RNSAP and RRC.

Figure 4-12: Cell Traffic Recording (CTR)

The CTR function collects data for a number of UE connections within a certain recording area. The recording area is defined as a specific cell in which the UEs are going to be followed. CTR can record up to a maximum of 16 simultaneous connections. The first 16 UEs satisfying triggering events in the recording area, composed of an Access Cell, are recorded.

A maximum of two CTRs recording subscriptions with 16 simultaneous UE connections each are allowed per RNC. A recording for a UE starts when it fulfills triggering events in the selected Access Cell. Default triggering event is `RRC Connection Setup`, this is the very first message that can be recorded by the CTR function during a connection setup.

RNC Event data, and measurement data from the RBS and UE may be selected for recording. The user can choose to record one or more messages within of the following protocol groups: NBAP, RANAP, RNSAP and RRC. The radio measurements and events that can be selected for both the uplink and the downlink are listed in RNC CPI "*Performance Recording" RNC 3810*, 72/1551-AXD 105 03/1

CTR is specifically tailored to monitor limited network areas, like individual cell performance. It is typically used for trouble shooting and verifying configuration changes such as the handover parameters.

2.15 PM Recording Administration

The Performance Management function within OSS-RC enables operator to initiate and manage (administer) Performance Recordings within the RAN network. Performance Recordings (UETR/CTR) execute as one or more Performance Monitorings on the RNC Nodes. The OSS-RC Operator controls the Performance Recordings (Monitorings) by creating GUI driven Subscription Profiles for Performance Recording data. OSS-RC allows for multiple RNC selection in the same recording profile for PM recordings UETR/CTR.

OSS-RC supports a machine-machine interface towards RNC that allows OSS-RC to administer the PM recordings. For each ROP, the recorded result is one file per recording UETR or CTR. All ROP files are automatically collected after the end of every ROP. The recording files are stored in the OSS-RC file system, where external OSS-RC users such as Network Management System users, are able to/may access them using FTP.

The PM recording data is fetched by OSS-RC from the network element every Result Output Period (ROP). Once OSS-RC has retrieved the compressed recording file, it saves the data for minimum 1 day (default value), before the data is being deleted. This OSS-RC parameter `recordingFileStorageDays` controls ROP file storage time and is user settleable, where the minimum value is 24 hours, and the maximum only limited by the available disk space.

The PM recordings when active, are adding to the data volume and to a limited extend the Module MP load. A common rule for the PM recording functions is that low traffic activity normally gives smaller recording files with less impact on Module MP load. Whereas, high traffic load increases the data volume and slightly the Module MP load.

2.16 REI, Recording and Event Interface

OSS-RC provides an optional Recording and Event Interface for retrieving ROP files (UETR/CTR) in ASCII text and or tab-delimited text format. The ROP files PM recording files are ASN.1 encoded bit-packed and compressed using GZIP. The REI parser handles the decompression and file conversion into ASCII text or tab delimited text format, allowing the end-users to follow connection related signaling in chronological order (UETR) and individual connection level.

Scanner ID information is included in the event header data for all PM recording events allowing for individual data parsing of the GPEH recordings. In addition, the "Content Based Parsing" concept has the capability to parse CTR/UETR/GPEH files related to certain profiles initiated from PMS GUI. That includes parsing of the entire recording profile, one or several ROPs in the same move.

2.17 GPEH

General Performance Event Handling (GPEH) is an optional and most flexible recording feature allowing operators to freely create subscriptions on various system levels. From basically few selective events involved in a particular traffic scenario all the way to full system recording including most or all RNC supported events and RBS internal events available in WCDMA RAN.

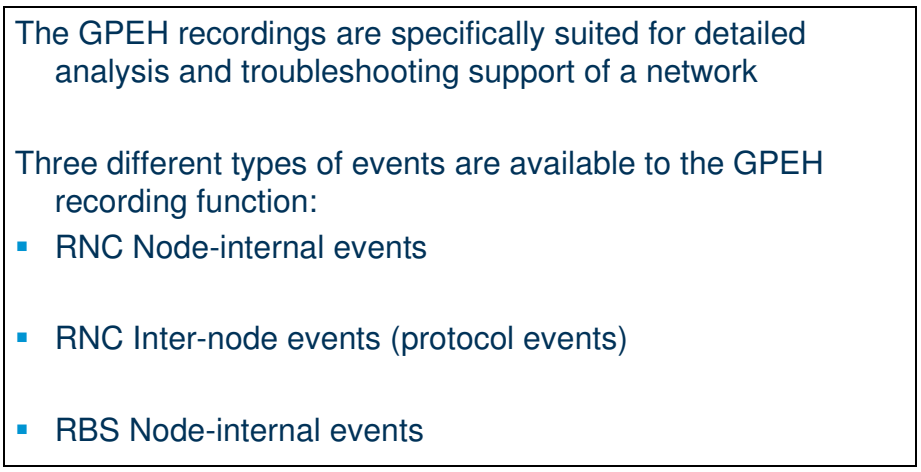


Figure 4-13: General Performance Event Handling

By analyzing the event based information, the GPEH function provides monitoring and evaluation capability of any traffic scenario taking place in a WRAN. The GPEH recordings are specifically suited for detailed analysis and troubleshooting support of a network, including:

- Detecting unacceptable performance degradation in the network, enabling the operator to take immediate actions to preserve quality of the network.
- Advanced troubleshooting of any traffic related function and it's algorithm in the network.
- Detailed performance monitoring and network optimization in order to further improve subscriber-perceived quality or a better utilization of the installed resources.
- Providing network planners with the detailed information related to dimensioning and configuration of the network.

The GPEH data is accumulated into files for the duration of Report Output Period (ROP). The ROP duration is 15 minutes. The GPEH data/ ROP files are encoded in ASN.1 and converted to bit-packed binary format. In addition, the ROP files are also compressed using GZIP compression method and stored by the NE for at least 1 hour- all depending on the file size and the storage size setting in the network element. The GPEH capacity is now boosted to handle more advanced network recordings involving more events and full network recording reducing the risk for missing data or recording interruptions. This means that the maximum supported RNC file size is increased from 60 MB/MP to 100 MB/MP uncompressed data.

The GPEH files are not automatically decompressed when fetched by OSS-RC, they are by default stored as compressed files in order to save OSS-RC storage space. In OSS-RC there is an optional feature "Recording and Event Interface (REI)" for converting the GPEH ROP files into text or tab delimited text format.

Cell ID is among common message data for a subset of GPEH messages (RNC external and RNC internal messages). Three different types of events are available to the GPEH recording function:

- RNC Node-internal events
- RNC Inter-node events (protocol events)
- RBS Node-internal events

2.18 RNC Node-internal Events

The RNC node-internal events aim to observe the performance of traffic related algorithms in the WCDMA RAN. As part of the RNC internal events there are a number of `smart events`. Smart events are events that have been specially designed to capture important error, fault or service failure cases. The RNC internal events in the GPEH recording carry massage header data (including event name, timestamp, RNC module Id, Ue context ID) as well as the full message contents. They are recordable on RNC level.

2.19 RNC Inter-node Events

The Inter-node events are based on the RNC external protocols. The events supported are the L3 messages according to the RRC, NBAP, RANAP, and RNSAP protocols. These events are recordable on RNC level and carry massage header data (including event name, timestamp, RNC module Id, Ue context ID) with or without the full message contents. The parameter `gpehDataLevel` controls the data level for protocol messages including or excluding the ASN.1 part in the recording.

2.20 RBS Node-internal Events

The RBS node-internal events aim to observe the radio link supervision/ synchronization for a specific RBS node. These events carry massage header data (including event name and timestamp) as well as the full message contents. These events are recordable on RBS level.

2.21 GPEH Administration

OSS-RC Performance Management GUI and a machine-machine interface allows users to schedule, collect and manage the PM recordings. The log files of GPEH Recording are temporarily stored by the NEs (RNC and RBS) with one file per Module Processor (MP) and ROP.

Compressed GPEH recording files are automatically fetched by the OSS-RC Performance Management function after the end of each ROP. The compressed files are locally stored in the OSS-RC file system, where OSS-RC external users have full access to the files using FTP. Collected GPEH files, by default remain in OSS-RC for at least one day before being deleted. The `recordingFileStorageDays` parameter is user controllable and can be changed in OSS-RC. The minimum value is 24 hours as ROP file storage period, and the maximum is only limited by the available disk space.

Note: In the GPEH recording it is possible to define ALL events, ALL cells, and ALL UEs for the same GPEH recording. However, in order to avoid extensive CPU load and prevent large amount of unnecessary GPEH data being recorded for each ROP, a filter containing limited number of events, and/ or limited number of cells shall be applied to exactly map particular recording need. This is in order to prevent recording interruption due to load or file size control.

2.22 PM Data Analysis Support

There are a number of optional WCDMA RAN features based on PM statistics and event data. These features are predominantly used for RN performance monitoring purposes like day-to-day operation or for trouble shooting the network.

The following WCDMA RAN analysis features are part of the Ericsson product portfolio:

- ENIQ, Ericsson Network IQ
- RES, Radio Environment Statistics - MRR-W
- NCS-W, Neighboring Cell Support
- TEMS Visualization tool.

Figure 4-14: PM data analysis tools

2.23 ENIQ, Ericsson Network IQ

Ericsson Network IQ is a proven product for performance management in multi-vendor and multi-technology environments. It can be deployed on a wide range of network technologies and data sources. Performance Management comprises of collecting network data, predominantly counter data from various network elements, including: data storage, data processing, and providing end users with performance reports.

Ericsson Network IQ product delivery comprises a generic Ericsson Network IQ system platform including: ETLC, Sybase IQ data warehouse, a set of Technology Packages, and Ericsson Network IQ Web Portal.

2.24 RES, Radio Environment Statistics - MRR-W

The Radio Environment Statistics (RES) optional feature in WCDMA RAN consists of two parts:

- RNC RES measurement function implemented in the RNC
- MRR-W is an OSS-RC application implementing all user interfaces and post-processing

RES allows operators to evaluate and supervise network performance where the subscriber-perceived quality is the key part of the feature typically used during network optimization activities. The concept is to perform statistical measurements of important radio characteristics per cell and save the results as statistical distributions.

2.25 NCS-W, Neighboring Cell Support in WCDMA

The Neighboring Cell Support feature for WCDMA (NCS-W) is an optional feature available in the Ericsson WCDMA Radio Access Network (RAN). NCS-W is part of the Radio Network Optimization (RNO) in OSS-RC.

The purpose of the Neighboring Cell Support (NCS) feature is to provide an easy and efficient way to keep the neighbor relations in the WCDMA Radio Network optimized. This means that support is provided to find missing neighboring cells that should be defined as neighbors, and to find currently defined neighbor relations that can be removed.

A properly defined neighbor cell relations is the single most common factor contributing to fewer dropped calls in a WCDMA network. Only neighbor relations between WCDMA cells using the same frequency (ARFCN), that is the neighbor relations used for soft/softer handovers, can be optimized with NCS.

2.25.1 TEMS Visualization

This is a stand-alone tool runs on a PC (Windows) used for advanced GPEH and UETR data analysis support.

The GPEH event data can be analyzed on a call by call basis with graphical information presented in map and measurement views. Statistics are also generated, for example, on cell and IMSI level. UETR recordings and MTR recordings (for GSM) can be processed into the same database which allows for advanced analysis of IRAT handovers.

3 SUMMARY

After this chapter the participants should be able to:

4 Be able to tie together the Performance Statistics and Performance Monitoring in the process of troubleshooting in WRAN Network.

4.1 Be able to initiate statistics from the OSS-RC.

4.2 Be able to initiate performance recording (e.g. UETR) from the OSS-RC.

4.3 Be able to read counter values and tie them to the situations in the network.

4.4 Explain Enhanced Health Check Filter File.

4.5 Genrate some faults and Show how to step by step resolve the faults.

Figure 4-15: Summary of Chapter 4

Intentionally Blank

5 APPENDIX A: AMOS Introduction

Intentionally Blank

1 AMOS Overview

AMOS is a text-based Operation and Maintenance (O&M) client providing access to the following services:

- Alarm Service (AS)
- Configuration Service (CS)
- File Transfer (FTP/HTTP)
- Inventory Service (IS)
- Log Service (LS)
- Notification
- OSE Shell (COLI)
- Performance Measurement Service (PM)

Figure 5-1. Services provided by AMOS

Access to all services is supported in both secure mode (secure CORBA, SSH, and SFTP) and non-secure mode (non-secure CORBA, TELNET, and FTP). The following diagram summarizes the access from AMOS to the CPP node.

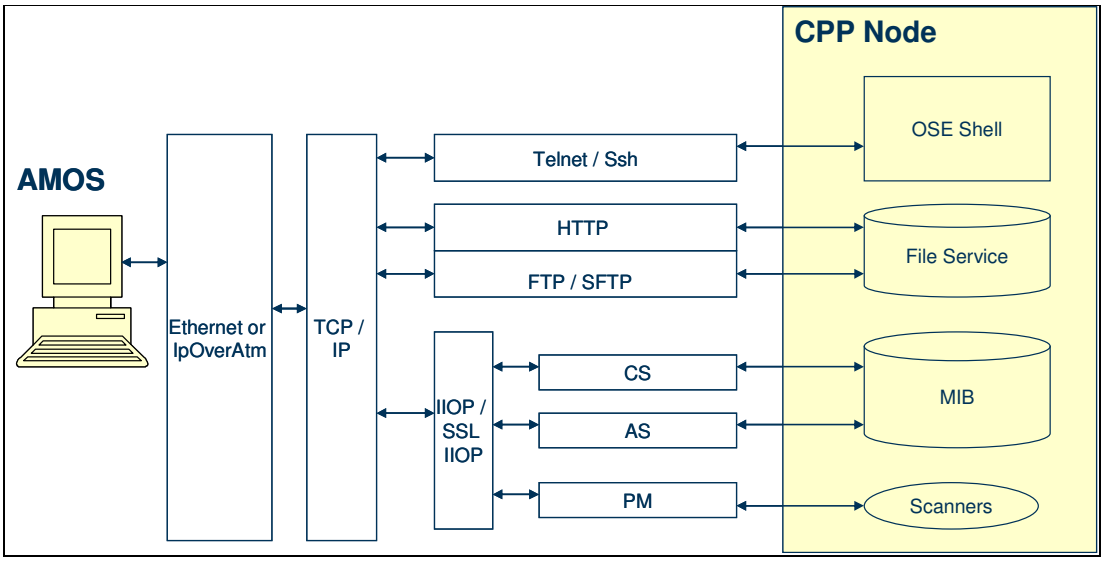


Figure 5-2: Access to the CPP node

1.1 Alarm Service

The Alarm Service can be used to retrieve the list of alarms currently active on each Managed Object (MO). The list of active alarms can be retrieved with the commands **`al`** or **`ala`**.

1.2 Configuration Service

The Configuration Service is used to read and change configuration data. Configuration data is stored in MO attributes. AMOS supports the following operations on the configuration service:

- *GetChildren* - To load all or parts of the MO-tree
- *GetAttribute* - To read the attributes of an MO
- *CallAction* - To perform an action on an MO
- *SetAttribute* - To set (change) the value of an MO attribute
- *CreateMO* - To create a new MO in the Network Element
- *DeleteMO* - To delete an MO from the Network Element

1.3 File transfer

AMOS can download and upload files and directories to and from a Network Element (NE). Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP) or Secure File Transfer Protocol (SFTP) may be used.

1.4 Inventory Service

The Inventory Service allows AMOS to get a list of all Hardware (HW) and Software (SW) defined on the NE.

1.5 Log service

The Log Service allows AMOS to save a log of certain events such as changes in the configuration data, alarms raising and ceasing, NE or board restarts, Java Virtual Machine (JVM) events and Operation and Maintenance security events. AMOS supports fetching and parsing of the following logs:

- Availability log
- System log
- Alarm / Event log

- Command log (CORBA_AUDIT)
- O&M Security log (SECURITY_EVENT)
- Command Line log (SHELL_LOG)
- HW inventory log
- Upgrade log (Trace.log)

1.6 Notification Service

The Notification Service allows AMOS to subscribe and receive notifications from the Network Elements. This informs AMOS about parameter and alarm state changes in the MOs.

1.7 OSE shell

Any OSE shell command can be typed at the AMOS prompt, and the output can be piped through external utilities if required.

1.8 Performance Measurement Service

The Performance Measurement Service is used to monitor statistics scanners or event filters. The statistics counters are stored in MO Performance Management (PM) attributes and are output to an XML file every fifteen minutes. The events are output into binary files every fifteen minutes. AMOS supports the following four operations from the performance management service:

- List Scanners and Event Filters
- Stop Scanner
- Resume Scanner
- Set Event Filter

The objective of this chapter is to list the common commands for reference while working with AMOS. There are lots of other commands and utilities that are not mentioned in the rest of the chapter. The OSS-RC CPI document *"AMOS, Advanced MO Scripting, User Guide"* could be used to consult the full AMOS usage. Alternatively, the help command **'h'** in AMOS could also be used to get the description of the commands.

It is expected that the reader performs the exercises associated with this training to get a working experience of AMOS.

1.8.1 Getting Started

AMOS is an optional tool in the OSS-RC. Provided that the license for this optional feature is installed, the following diagram explains the basics of working with AMOS:

- Start AMOS:
  Shell in OSS-RC> **moshell X** (where X=IP address or DNS name of the node)
- Exit AMOS:
  AMOS> **exit** (or **q** or **quit** or **bye**)
- Load the MO tree and build a proxy table
  AMOS> **lt xx** (xx could be 'all' or some MoTypes)
- Get help on AMOS commands
  AMOS> **h xx** (xx can be ignored, or it could be a command)
- Get description from the MOM document
  AMOS> **mom xxx** (where xxx is a MoType)

Figure 5-3. Getting started

Note that the 'help' command is a COLI (Command Line Interface) command and not a AMOS command. However, since AMOS provides accesses to the COLI interface also, the normal CLI commands can also be executed from the same AMOS window.

1.8.2 AMOS Command List

Here is a list of commands that are available in AMOS. This list is created by typing the command '**h**' in AMOS. Not all commands are possible to execute, and there might be further restrictions on who can execute which commands depending on the user rights assigned in the OSS-RC. The command list also might also be different on the various releases of AMOS.

---------- BASIC MO COMMANDS ----------

mom[tcd]	Print description of MO classes, CM/FM attributes, actions, enums and structs.
lt/ltc[1-9]	Load MO tree (full or partial) and build proxy table.
lc[1-9]/lcc	Load MO tree (full or partial) and build proxy table.
pr/lpr	Print MO LDNs and proxy ids for all or part of the MO tree currently loaded in moshell.
ma/lma	Add MO(s) to an MO group.
mr/lmr	Remove an MO group or remove MOs from an MO group (MOs will NOT be deleted, only the group).
mp	Print all defined MO groups.
get/lget	Read CM/FM attribute(s) from MO(s).
hget[c]/lhget[c]	Read CM/FM attribute(s) from MO(s), print horizontally one line per MO (instead of one line per attribute).
kget/lkget	Display CM/FM attributes in exportable printout format.
fro/lfro[m]	Read MAO data of an MO and/or froid of the MO.
st/lst	Print state of MOs (operationalState and administrativeState when applicable).
prod	Print productData of MO(s).
lk/llk	View all MO's linked to an MO, and their states (admState and opState).
lko/llko	The old lk.

set[m]/lset[m]	Set an attribute value on one or several MO's.
rset/lrset	Set attribute value on a restricted attribute or change the MOid of an MO.
bl[s]/lbl[s]	Lock or soft-lock MO(s).
deb/ldeb	Unlock MO(s).
acl/lacl	Lists available MO actions.
acc/lacc	Execute an MO action.
cr	Create an MO.
del/ldel	Delete MO(s).
rdel/lrdel	Delete MO(s) together with children and reserving MOs.
u+[s]/u-/u?/u!	Handling of undo mode (for undo of del/rdel/set commands).
Run	Run a command file in moshell format.
trun[i]	Run a command file in EMAS/MoTester format.
ctrl-z	Abort an MO command or a "for" loop. Type "touch /tmp/<stopfile>;fg" to resume the moshell session.
ol[s][h][c][u]	Poll the node until the MO service is up or until an operation has completed.
re[i]	Disconnect and reconnect to the CM service (mobrowser) and/or the PM service (pmtester).
getmom	Check the MOM version currently stored on the node.
parsemom	Parse an xml MOM file.
ld	Load one MO from the tree and add to the proxy table.
sget/lsget	Read CM/FM attributes from MO(s), one by one ("Slow" get).

---------- OTHER MO COMMANDS ----------

cvls/cvmk/cvms/cvset/cvrm/cvcu/cvget/cvput/cvls1 CV backup handling: list, make local, make remote, remove, setstartable.

inv[hr]	Complete HW/SW inventory. Includes information about RPUs, licensing, JVM, devices, XPs, ISL, etc.
cab[slxradgtme]	Display of miscellaneous COLI printouts relating to hw, sw, restarts, leds, cpu load, errors, disk/ram usage
stc[p][r]	Display state and configuration of Atm/Tdm CrossConnections.
std	Display state and configuration of devices (RNC and MGW only).
stv[b][r]	Display state, user, and bandwidth usage for ATM ports and channels.
stt[r]	Display state and user of Physical Ports and Ds0Bundles.
hc	Run a general healthcheck on the node. Obsolete ! Use dcg command instead.
diff/ldiff	Parameter auditing or MO dump comparisons.
str[12ft]	Print status of the IubLinks/AbisLinks and their associated Cells and Channels (RNC)
lki	Resource usage and configuration of IubLinks (RNC only).
remod[u][2]	Remodule an IubLink (RNC only).
tg[r][c][d]	Print Resource Object information for all MOs in LmCell (RNC only).
uer[d][t]	Print UE context data (serving or drifting) for all active calls (RNC only).
al[atkc]	Print the list of active alarms. Acknowledge/Unacknowledge an alarm.
lg[aevsyuoldhmircf]	Fetching and/or processing of node logs (alarm, event, availability, system, etc)

---------- OTHER COMMANDS ----------

uv	Print or change moshell configuration settings (also called "user variables").
pv	Print scripting variables.
!/l	Execute a unix command on the PC/workstation.
l+[m][m][s][o]/l-/l?	Open/close moshell logfiles.
ose/coli command	Send a COLI command to the node's OSE shell. Type "h ose" for syntax help and "?" to view available commands.
bo[r]/ba[swdp]/br[wd]/be[0-50]/bp	Manage board groups that can be used for running COLI commands on multiple boards.
lh	Run COLI commands on all boards of a board group.
mon/mon+/mon-/mon?	Start/stop/check the target monitor server on the node and/or start the monitor client for one or more board Group(s).
sql+/sql-/sql?	Start/stop/check the SQL client on the node (CXC1325608).
pgu[c][f][r]	Program Upgrade. For STP use only, eg, to load black LMs.
proglist/progkill	List or restart programs on boards or board groups.
ftree[f]	Recursive listing of a directory on the file system of the node or the workstation.
ftget[c]/ftput[c]	Transfer files or directories to/from the node, using ftp or sftp.
htget	Transfer files from the node using http.
edit	Edit a file on the node.
fclean[f|ff|d|e]	Removal of obsolete loadmodules OR recursive removal of a directory on the node.

hi	Print history of moshell commands entered during the current session.
lmid[c]	Print translation of loadmodule product number or T&E error codes.
p/w/pw/b	Change moshell prompt and/or window title.
prox	Toggle display of proxy identities in printout of get <mo> <attribute> command.
col	Toggle display of colors.
ul	Toggle display of userlabel in st/lst and pget/lpget printout.
conf[bld]	Toggle confirmation on various MO commands.
gs/gsg	Toggle display of old/new attribute value in set/bl/deb commands.
ip2d/d2ip	Convert an IP address into the format used in the fRO (sql database) or vice-versa.
h2d/d2h	Convert an integer to hexadecimal or viceversa.
h2b/b2h	Convert a binary to hexadecimal or viceversa.
wait	Specify a delay in hrs, mins, secs, or rops. Similar to the unix "sleep" command (scripting).
return	Exit from a command file without exiting from moshell (scripting).
print	Print a line or variable (scripting).
alias/unalias	Print or define command aliases.
q/bye/exit/quit	Exit moshell.

---------- PM COMMANDS ----------

pmom[acd]/lmom[c]	Print description of PM counters (pmom) or log attributes (lmom, CDMA only).
pget/lpget	Read PM attribute(s) from MO(s).
spget/lspget	Read PM attribute(s) one by one ("slow pget").
hpget[c]/lhpget[c]	Read PM attribute(s) from MO(s), print horizontally one line per MO (instead of one line per attribute).
pdiff/lpdiff	Print incrementation of PM attributes.
pmx[hfdn]	Display counter values, extracted from the statistics ROP files.
pmr[ag]	Produce PM KPI reports, based on counter values in statistics ROP files and formulas in CPI documentation.
pme[fd][cgu]	Fetch/decode event ROP files (RNC/RBS only).
pst	List all PM scanners and their state.
pgets[n]	Print scanner contents.
pcr[cfd]/lpcr[cfd]	Create a statistics scanner.
pbl	Suspend a scanner.
pdeb	Resume a scanner.
pdel	Delete a scanner.
emom	Display list of events available for each kind of event-based scanner.
pset[d]	Set the contents of an event-based scanner (RNC/RBS only).

1.8.3 Addressing Managed Objects in AMOS

In AMOS, when the Managed Objects need to be addressed, they could be addressed with one of the following six ways:

- **1 all or ***
  E.g. AMOS > **get all userlabel**
- **2 Proxy ID(s)**
  E.g. AMOS > **pr 0 2 5** print the proxies 0, 2 and 5
  E.g. AMOS > **pr 4-10** print proxies 4 to 10
  E.g. AMOS > **acc 10-20 restart** perform restart action on proxies 10 to 20
- **3 Link handler (for PluginUnit and Spm MOs only)**
  E.g. AMOS > **acc 001400 restart**
  E.g. AMOS > **bl 001900/sp0.lnh**
- **4 MO Group**
  MO Groups are user defined groups of MOs. All MO(s) belonging to the given MO group will be operated upon. Check the commands **ma/lma**, **bo** and **mp** on how to create/print mo group.
- **5 Board Group**
  Check the commands **bo** and **baw**
- **6 MO Filter (Regular Expressions)**
  E.g. AMOS > **pr ms-24-1**
  E.g. AMOS > **|pr ms-24-1**

Figure 5-4. Addressing MOs in AMOS

1.9 Regular Expressions

Symbol	Meaning
.	Match any single character
*	Match 0 or more occurrences of the previous character
[]	Match a character or range of characters inside the brackets
[^]	Do NOT match character or character range inside the brackets
^	Match from the beginning of the string
|	OR
$	Match from the end of the string.
!	Negation. Do not match
%	Match in reverse order
()	Group regular expressions.

Examples	Meaning
a*	Match a or aa or aaa.
.*	Wildcard that matches 0 or more occurrences of any character.
[a-z]	Matches all letters from a to z.
[abe]	Matches letters a, b and e
[^3]	Matches any character but not 3
3|5|6	Matches 3 or 5 or 6
^a.*4$	Matches a string beginning with a and finishing with 4, with any character in the middle
cell(11|23|45)	Group regular expressions together using brackets. This matches cell11 or cell23 or cell45.

Figure 5-5. Regular expressions

1.9.1 Useful Utilities

- **Command Piping**
  AMOS > **te log read | grep ERROR > lh mp te log read | grep ERROR**

- **Opening/closing a logfile**
  AMOS > **l+[m][m][s][o] [<logfile>]**
  AMOS > **l- [<logfile>]**
  AMOS > **l?**

- **Printing the history of commands in the current AMOS session**
  AMOS > **hi**

Working with COLI while in AMOS

AMOS > **?**	List all COLI command (like 'help' command in the COLI)
AMOS > **? <command name>**	List information about a particular COLI command

Working in the local (OSS/UNIX) filesystem while in AMOS

AMOS > **l UNIX command**

e.g AMOS > **l more File.txt**	Perform **'more'** on File.txt that is in pwd in the OSS-RC

Figure 5-6. Useful utilities

1.9.2 Accessing AMOS COMMANDS Help

- To show all help topics for AMOS, use the following command:
  - AMOS > **h**
- To show help related to a particular AMOS command, use the following command:
  - AMOS > **h <command name>**
- To show the old MOSHELL user guide in online mode, give the following command:
  - AMOS > **h <chapter number>**
- To show first set of AMOS commands ("basic commands"), use the following command:
  - AMOS > **m**
- To show second set of AMOS commands ("other mo commands"), use the following command:
  - AMOS > **n**
- To show third set of AMOS commands ("other commands"), use the following command:
  - AMOS > **o**
- To show all performance related AMOS commands, use the following command:
  - AMOS > **p**

Figure 5-7. Accessing AMOS commands help

1.9.3 Accessing online MOM Description

- Checking the MOM version
  AMOS > **getmom**
- Viewing the whole Managed Object tree:
  AMOS > **momt**
- Viewing all possible parents and children of a Managed Object class:
  AMOS > **momt <moclass, struct or enum>**
- Viewing the description of a Managed Object class:
  AMOS > **mom <moclass, struct or enum>**
- Viewing the description of a Managed Object class and all its children/grandchildren:
  AMOS > **momc momc <moclass, struct or enum>**
- Viewing the description of all attributes of a Managed Object class
  AMOS > **mom <moclass, struct or enum> all**
- Viewing the description of an action
  AMOS > **mom mom <action>**
- Viewing the descriptions of all attributes of type enum:admstate
  AMOS > **mom all all enumref:adms**
- Viewing the descriptions of all members of a struct type
  AMOS > **mom adminproductda all**
- Viewing all attributes of type sequence:moRef who have a flag restricted
  AMOS > **mom all all sequence:moref restricted**
- Viewing all attributes that contain a specified word in their descriptions
  AMOS > **mom all all all all <specified word>**

Figure 5-8. Viewing MOM Description

1.9.4 Loading and Unloading the Managed Objects

AMOS works on Managed Objects that are stored locally as a proxy-table. It is possible to control how much or what type of Managed Objects should be loaded in the proxy table. The command **l** with different switches (to make commands **lt**, **lc** and **ld** to load and **lu** to unload) is used for this. The following diagram explains the format:

- Load MO types/children:
  AMOS > lt/ltc[1-9] <motype-filter>|root|all [<attribute==value> AND/OR <attribute==value>]
  - The 't' is for MO type and 'c' is for children and the numbers 1-9 is for the 'depth of children' under the specified MO (or MOType)

  AMOS > lc/lcc[1-9] <moGroup>|<moFilter>|<proxy(s)>|all
- Unload MO
  AMOS > lu/llu <moGroup>|<moFilter>|<proxy(s)>
  - This is used in nodes with large configuration to decrease the load on the OSS-RC by reducing the size of the proxy table

Figure 5-9. Loading and unloading MO tables

1.9.5 Printing Managed Object Data

- Printing all MOs currently loaded
  - AMOS > **lt all**
  - AMOS > **pr**
- Printing a selection of MOs
  - AMOS > **pr <proxy identity>**
  - AMOS > **pr <RDN>**
  - AMOS > **pr !utrancel|fach** : *To print everything but NOT UtranCell or Fach*
  - AMOS > **mp** *: To check all defined MO groups, then can be followed by* ***pr Mo_group***
- (MO groups are created with **ma** or **lma** commands, and removed with **mr** command)

Figure 5-10. Printing from the MO Proxy table

- Printing State of all Managed Objects
  - AMOS > **st**
- Printing the state of all disabled Managed Objects
  - AMOS > **st all dis**
- Printing all disabled Managed Objects under a particular Managed Object
  - AMOS > **lst e1phys dis**
- Print all Managed Objects that are unlocked and disabled
  - AMOS > **st all 1.*0**
- Print all Managed Objects that are locked
  - AMOS > **st all ^0**
- Printing the state of all channels in cells based upon RDN
  - AMOS > **lst cell=cell37**
- Print all Managed Objects linked to a Managed Object and its state
  - AMOS > **lko <filter>** or **llk <filter>** or **lko <filter>**
    - The <filter> could be IubLink,UtranCell, Ranap, Rnsap, Vmgw, Mtp3bSls, Mtp3bSrs, M3uAssociation,UniSaalTp, NniSaalTp,Aal5TpVccTp, Aal0TpVccTp, Aal1TpVccTp, Aal2PathVccTp, VclTp, VplTp, VpcTp, Aal2RoutingCase, Aal2Ap,AtmPort, ImaGroup.

Figure 5-11. Printing state of MOs

- Printing Status of Links and their associated Cells and Channels (RNC only)
  AMOS > **str** or **str1** or **str2** or **strt**
- Printing Resource Usage and Configuration of IubLinks (RNC only)
  AMOS > **lki**
- Printing Resource Object information (RNC only)
  AMOS > **tg**
- Printing UE Context Data for all Active Calls
  AMOS > **uer**

Figure 5-12. Useful print commands in the RNC

1.9.6 Alarm Handling

- Printing an overview of all Alarms
  AMOS > **al**
  AMOS > **ala** (more detailed)
- Printing an overview of only critical Alarms
  AMOS > **al | grep "Crit "**
- Counting the number of Major Alarms
  AMOS > **al | grep -c "Maj "**
- Printing all Active Alarms sorted chronologically
  AMOS > **alt**
  AMOS > **alat** (with more details)
- Printing all Active Alarms with acknowledged alarms and unacknowledged alarms printed separately
  AMOS > **alk**
  AMOS > **alak** (with more details)

Figure 5-13. Alarm Handling

1.9.7 AMOS commands for Managed Objects Handling

- To read CM/FM attributes from a MO:
  AMOS > **get**
  - The get command can be combined with other options to influence how the printout should look like or if the children should be included . Examples:
    - AMOS > **hget** horizontal get
    - AMOS > **sget** slow get
    - AMOS > **hgetc** horizontol get with a comma-seperated format
    - AMOS > **lget** a get for the mo and the children underneath it
    - AMOS > **kget** similar to get but meant for export to a file
- Managed Object data is stored on the Network Element in an SQL table. Although not a normal operation, data can be directly read from the SQL database instead of using the Managed Object Service.
  AMOS > **from 0**
  AMOS > **fro plugin** Print just the resourceId for all PluginUnit

Figure 5-14. Working with MOs- printing attributes

- Change (or set) a non-restricted attribute on one or a number of MOs
  AMOS **> set[m]/lset[m] moGroup|moFilter|proxy(s) attribute [value]**

  AMOS > **set cell primarycpichpower 250**
  (set the primarycpichpower attribute to 250 for all cells)
- Locking and Unlocking Managed Objects
  AMOS > **bls <moFilter>**
  This is for soft lock. Proxy could be replaced by <moFilter> could be replaced by <proxy>

  AMOS > **bl <proxy>**
  This will perform a hard lock. <proxy> could be replaced by <moFilter>

  AMOS > **deb <proxy>** OR
  AMOS > **deb <moFilter>**

Figure 5-15. Working with MOs – set and (de)block

- AMOS > **cr LDN**

Example:

- AMOS **> cr TransportNetwork=1,AtmTrafficDescriptor=Yo**
- Attribute 1 of 3, ingressAtmQos (enumRef:AtmQos):
- Enter one of the following integers: 1:CLASS_ONE, 2:CLASS_TWO, 3:CLASS_THREE, 4:CLASS_FOUR: **1**
- Attribute 2 of 3, egressAtmQos (enumRef:AtmQos):
- Enter one of the following integers: 1:CLASS_ONE, 2:CLASS_TWO, 3:CLASS_THREE, 4:CLASS_FOUR: **1**
- Attribute 3 of 3, serviceCategory (enumRef:ServiceCategory):
- Enter one of the following integers: 1:SERVICE_CATEGORY_CBR, 2:SERVICE_CATEGORY_UBR, 3:SERVICE_CATEGORY_UBR_PLUS: **1**
- Following attributes are optional. Enter attribute value or "d" for default. Once the MO is created, these attributes cannot be changed (they are restricted).
- Attribute 1 of 5, ingressAtmPcr (long): **100 ;** Attribute 2 of 5, egressAtmPcr (long): **100**
- Attribute 3 of 5, egressAtmMcr (long): **0 ;** Attribute 4 of 5, ingressAtmMcr (long): **0**
- Attribute 5 of 5, packetDiscard (boolean): **false**
- >>> [Proxy ID = 3313] MO name :ManagedElement=1,TransportNetwork=1,AtmTrafficDescriptor=Yo

Figure 5-16. Working with MOs- create a MO

- Deleteing one or several MOs

  AMOS > **del/ldel moGroup|moFilter|proxy(s)**

  - E.g: AMOS > **ldel %ms,slot=20,plug** Deleting a Managed Object and all its children

  AMOS > **rdel/lrdel moGroup|moFilter|proxy(s)** Delete MO(s), plus their children and reserving MOs

- Actioning a Managed Object
  - AMOS > **acc/lacc moGroup|moFilter|proxy(s)|all**
    - E.g: AMOS > acc ip listroutes
- Comparing Managed Objects

  AMOS > **diff/ldiff**

  command can be used for parameter auditing or MO dump comparisons, it may be used to compare two or three Managed Objects side by side. Managed Objects must be of same Managed Object class. All attribute values that are different between the Managed Objects will be printed.

Figure 5-17. Working with MOs-delete, action and compare

1.9.8 Configuration Version Handling

▪ AMOS > **cvls** and CV	Displays both the current CV information (equivalent of **cv cu**) list (equivalent of **cv ls**).
▪ AMOS > **cvcu**	Displays the current CV information only (equivalent of **cv cu**).
▪ AMOS > **cvmk**	Creates a CV.
▪ AMOS > **cvset**	Set a CV as startable.
▪ AMOS > **cvms**	Create a CV and make it startable (combination of **cvmk** and **cvset**)
▪ AMOS > **cvrm**	Can remove many CVs in one go using pattern matching on the CV name (remove from rollback list is attempted before deletion).
▪ AMOS > **cvget**	Make a remote backup of a CV to the workstation.
▪ AMOS > **cvls1**	Gives similar output to **cvls** but accesses the CV information through OSE shell instead of the MO interface.

Figure 5-18. Configuration Version (CV) handling

1.9.9 Working with Logs

- AMOS > **lg[aevsmircdyuolhf] [-l <logdirectory | logfile>] [-m <minustime>] [-p <plustime>] [-s <startdate>] [-e <enddate>] [| unixcmds]**

  a - Parse alarm log
  e - Parse event log
  v - Parse availability log
  s - Parse system log
  u - Parse upgrade log
  y - Parse securityevent log
  l - Parse COLI SHELL_AUDITTRAIL log
  h - Parse HILI log
  m - Merge the different logs together. (Example: **lgaevm** will merge alarm/event/availability logs).
  i - Inverse chronological order.
  r - Refetch the logs from the Network Element
  c - Print the output into a file, in csv format.
  d - Show Network Element downtime figures. This option can only be combined with the r option.
- AMOS > **dcg** This is used to collect more logs by Ericsson while troubleshooting

Figure 5-19. Logs

1.9.10 Miscellaneous utilities

- **Print a variety of hardware related information**
  AMOS > **cab[slxradgtm]** t=temperature, x=AuxUnit, l=load
  r=restarts, a=abnormal restarts,
  d= disc usage, m=RAM memory usage,
  g=HW errors, s=running programs
- **Recursive print of files**
  Eg.AMOS > **ftree /c/pmd** Print all the files + directories under /c/pmd
- **FTP**
  AMOS > **ftget<source_file>** Transfer file from the node to local dir
  AMOS > **ftput<source_file>** Transfer file from local dir to the node
- **Inventory**
  AMOS > **inv[hr] <filter>** Displays the complete HW/SW inventory.

Figure 5-20. Other useful utilities

1.9.11 AMOS and Performance Handling

AMOS can be used as a tool not only for fetching and parsing the statistics counters from the CPP based node, but also as a post-processing tool to provide Key Performance Indicators (KPIs). The fact that AMOS can also be used to decoder for the performance recording files (CTR, UETR and GPEH) makes it a very useful tool while troubleshooting. The following figures illustrate the commands used for performance handling in AMOS:

- Performance related documentation
  - **pmom[acd]/lmom[c] [<moclass>] [<attribute>] [<attr-description>]**
- Getting the performance values
  - **pget/lpget [<moGroup>|<moFilter>|<proxy(s)>|all] [<attribute-filter>|all] [<value-filter>]**
  - **hpget/hlpget** displays the output in horizontal format
  - The option c at the end (e.g. **pgetc**) displays in CSV format
- Printing the PM attributes whose values have changed
  - **pdiff/lpdiff [<moGroup>|<moFilter>|<proxy(s)>|all] [<attribute-filter>|all] [<value-filter>]**

Figure 5-21. Performance Handling with AMOS ..

- Displaying counter values extracted from the statistics ROP (Report Output Period) files:
  - **pmx[hfdn] [<mofilter>|<mogroup>] [<counter-filter>] [-l <PMfiles-directory>] [-m <minushours>] [-p <plushours>] [-s <startdate>[.<starttime>]] [-e <enddate>[.<endtime>]] [-a|-d|-h] [| <unix-command>]**
- Producing KPI reports based on counter values in ROP files and formulas in CPI documents:
  - **pmr[ag] [-r <report(s)>] [-l <PMfiles-directory>] [-i <iubCellModule-file>] [-f <formulafile>] [-c <configfile>] [-m <minushours>] [-p <plushours>] [-s <startdate>[.<starttime>]] [-e <enddate>[.<endtime>]] [-o <outputFormat>]**

Figure 5-22. Performance Handling with AMOS ..

- Fetching event ROP Files and decoding:
  - **pme[fd][cgu] [<pm_logdir>] [-b <boardgroup>] [-m <minushours>] [-p <plushours>] [-s <startdate>[.<starttime>]] [-e <enddate>[.<endtime>]]**

Figure 5-23. Performance Handling with AMOS ..

6 Acronyms and Abbreviations

16 QAM	16 Quadrature Amplitude Modulation
64 QAM	64 Quadrature Amplitude Modulation
2G	Second Generation
3G	Third Generation
3GPP	Third Generation Partnership Program
AAL2	ATM Adaptation Layer type 2
AAL5	ATM Adaptation Layer type 5
ACK	Acknowledgement
A-DCH	Associated Dedicated Channel
AG	Absolute Grant
AI	Acquisition Indicator
AIUB	Antenna Interface Unit Board
AICH	Acquisition Indicator Channel
ALCAP	Access Link Control Application Part
ALB	Alarm Log Browser
ALV	Alarm List Viewer
AM	Acknowledged Mode
AMR	Adaptive Multi Rate speech codec
AP	Access Preamble
APN	Access Point Name
ARFCN	Absolute Radio Frequency Channel Number
ARNE	Add Remove Network Element
ARQ	Automatic Retransmission Request
AS	Active Set
AS	Access Stratum
ASN.1	Abstract Syntax Notation 1 Format
ASM	Alarm Status Matrix
ASC	Access Service Class
ASCII	American Standard Code for Information Interchange
ASE	Air Interface Speech Equivalent

ATM	Asynchronous Transfer Mode
AUTN	Authentication Token
BBIFB	Baseband Interface Board
BCCH	Broadcast Control Channel
BCH	Broadcast Control Channel
BCFE	Broadcast Control Functional Entity
BER	Bit Error Rate
BLER	Block Error Rate
BMC	Broadcast/Multicast Control
BP	Board Processor
BTS	Base Station System
BSS	Base Station Sub-system
BSSMAP	Base Station System Management Application Part
CBC	Cell Broadcast Centre
CBS	Cell Broadcast Service
CBU	Control Base Unit
CC	Call Control
CCCH	Common Control Channel
CCPCH	Common Control Physical Channel
CCS	Common Channel Support
CCTrCH	Coded Composite Transport Channel
CEF	Complete Exchange Failure
CEP	Connection End Point
CEX	Common Explorer
CFN	Connection Frame Number
C/I	Carrier over Interference
CK	Cipher Key
CM	Connection Management
CM	Configuration Management
C-MXB	Common Main Switch Board
CN	Core Network
CNNHO	Core Network Hard Handover
CPCH	Common Packet Channel
COLI	Node Command Line Interface
CORBA	Common Object Request Broker Architecture
CPICH	Common Pilot Channel
CPP	Connectivity Packet Platform
CRC	Cyclic Redundancy Check
CRNC	Controlling RNC
C-RNTI	Cell RNTI
CS	Circuit Switched
CSR	Customer Service Request

CTCH	Common Traffic Channel
CTR	Cell Traffic Recording
CU	Capacitor Unit
CV	Configuration Version
CQI	Channel Quality Index
dB	Decibel
DCA	Dynamic Channel Allocation
DCS	Dedicated Channel Support
DCCH	Dedicated Control Channel
DCFE	Dedicated Control Functional Entity
DCH	Dedicated Channel
DC-SAP	Dedicated Control SAP
DDM	Discrete Distributed Measurement
DL	Downlink
DPC	Destination Point Code
DPCCH	Dedicated Physical Control Channel
DPCH	Dedicated Physical Channel
DRAC	Dynamic Resource Allocation Control
DRH	Device and Resource Handling
DRNC	Drift RNC
DRNS	Drift RNS
DRX	Discontinuous Reception
DSCH	Downlink Shared Channel
DTCH	Dedicated Traffic Channel
DTX	Discontinuous Transmission
E-AGCH	E-DCH Absolute Grant Channel
EBS-GSM	Event Based Statistics for Global System for Mobile Communications
EBS-W	Event Based Statistics for Wideband CDMA
Ec	Energy per Chip
E-DCH	Enhanced Dedicated Channel
E-DPCCH	E-DCH Dedicated Physical Control Channel
E-DPDCH	E-DCH Dedicated Physical Data Channel
E-HICH	E-DCH Hybrid ARQ Indicator Channel
EL2	Enhanced Layer 2
EM	Element Manager
ENIQ	Ericsson Network IQ
EP	Elementary Procedure
E-RGCH	E-DCH Relative Grant Channel
E-RNTI	E-DCH Radio Network Temporary Identifier
ES	Extension Subrack (Cabinet)
ETB	Exchange Terminal Board

E-TFC	E-DCH Transport Format Combination
E-TFCI	E-DCH Transport Format Combination Indicator
ET-IPG	Exchange Terminal Internet Protocol Gateway
ETLC	Extract, Transfer and Load Control
EUL	Enhanced Uplink
FACH	Forward Access Channel
FAUSCH	Fast Uplink Signaling Channel
FCCH	Forward Control Channel
FDD	Frequency Division Duplex
FEC	Forward Error Correction
FER	Frame Error Rate
FFS	For Further Study
FFV	File Format Version
FTC	Fault Tolerance Call
FTP	File Transfer Protocol
FM	Fault Management
FMX	FM Expert System
FN	Frame Number
FP	Frame Protocol
FU	Fan Unit
ICF	Interface Connection Field
ID	Identifier
GAN	Generic Access Network
GPB	General Purpose Processor Board
GPEH	General Performance Even Handling
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
GTP-U	GPRS Tunneling Protocol
HARQ	Hybrid Automatic Repeat reQuest
HLR	Home Location Register
H-RNTI	HSDPA - Radio Network Temporary Identifier
HS	High Speed
HS-DSCH	High Speed Downlink Shared Channel
HS-PDSCH	High Speed Physical Downlink Shared Channel
HS-SCCH	High Speed Shared Control Channel
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IE	Information element
IMEI	International Mobile Equipment Identity

IMSI	International Mobile Subscriber Identity
IIOP	Internetwork Inter-ORB Protocol
IP	Internet Protocol
ISCP	Interference on Signal Code Power
JVM	Java Virtual Machine
KPI	Key Performance Index
KSI	Key Set Identifier
L1	Layer 1
L2	Layer 2
L3	Layer 3
LA	Location Area
LAC	Location Area Code
LAI	Location Area Identity
LDN	Local Distinguished Name
LED	Light Emitting Diode
LM	Load Modules
MAC	Medium Access Control
MAC-HS	Medium Access Control – High Speed
MAC-I	The Message Authentication Code included in AUTN, computed using f1
MB	Megabyte
MC	Multi Carrier
MC-PQ	Multi Carrier – Priority Queue
MCPA	Multicarrier Power Amplifier
MCC	Mobile Country Code
Mcps	Mega Chip Per Second
MGw	Media Gateway
MIB	Management Information Base
MIMO	Multiple Input Multiple Output
MM	Mobility Management
MNC	Mobile Network Code
MO	Mobile Originating Call
MO	Managed Object
MOM	Managed Object Model
MPE	Mechanics, Power and Environment
MP-SP	Main Processor-Signal Processor
MRR	Measurement Result Recording
MS	Main Subrack (Cabinet)
MS	Mobile Station
MSC	Mobile services Switching Centre
MSP	Multiplex Section Protection
MSPG	Multiplex Section Protection Groups

MT	Mobile Terminal
MTC	Mobile Terminated Call
MTP3b	Message Transfer Protocol 3 Broadcast
M3UA	MTP3 User Adaptation
NAS	Non Access Stratum
NBAP	Node B Application Protocol
NCLI	Node Command Line Interface
NCS	Neighboring Cell Support
NE	Network Element
NMS	Network Management System
NESW	Network Element Software
No	Noise
NSA	Node Status Analyzer
NSD	Network Status Display
Nt-SAP	Notification SAP
NTP	Network Timing Protocol
NW	Network
ODMA	Opportunity Driven Multiple Access
OMINF	Operation and Maintenance Infrastructure
OSS-RC	Operation and Support Radio Core Network
OSE	Operating System Embedded
PCAP	Positioning Calculation Application Part
PCCH	Paging Control Channel
P-CCPCH	Primary Common Control Physical Channel
PCH	Paging Channel
PCI	Protocol control Information
PDCP	Packet Data Convergence Protocol
PDP	Packet Data Protocol
PDR	Packet Data Router
PDSCH	Physical Downlink Shared Channel
PDU	Protocol Data Unit
PHY	Physical Layer
PICH	Paging Indicator Channel
PLMN	Public Land Mobile Network
PM	Performance Management
PMS	Performance Management Subsystem
PMD	Post Mortem Dump
PN	Pseudo Noise
PNFE	Paging and Notification Control Functional Entity
PRACH	Physical Random Access CHannel
PS	Packet Switched
P-SCH	Physical Synchronization Channel

PSTN	Public Switched Telephone Network
P-TMSI	Packet Temporary Mobile Subscriber Identity
P-SCH	Primary Synchronize Channel
PUSCH	Physical Uplink Shared Channel
PVC	Permanent Virtual Circuit
Q	Quintet, UMTS authentication vector
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quaternary Phase Shift Keying
RA	Routing Area
RAB	Radio Access Bearer
RACH	Random Access Channel
RAI	Routing Area Identity
RAN	Radio Access Network
RANAG	Radio Access Network Aggregator
RANAP	Radio Access Network Application Part
RAS	Remote Access Service
RAXB	Random Access Receiver Board
RB	Radio Bearer
RBS	Radio Base Station
RES	Radio Environment Statistic
RF	Radio Frequency
RFC	RAKE Finger Count
RFE	Routing Functional Entity
RFIFB	Radio Frequency Interface Board
RG	Relative Grant
RL	Radio Link
RLS	Radio Link Set
RLC	Radio Link Control
RNC	Radio Network Controller
RLIB	RNC Component Library
RNH	Radio Network Handler
RNS	Radio Network Subsystem
RNSAP	Radio Network Subsystem Application Part
RNTI	Radio Network Temporary Identifier
ROP	Result Output Period
RPU	Reliable Program Uniter
RRC	Radio Resource Control
RSCP	Received Signal Code Power
RSN	Retransmission Sequence Number
RSSI	Received Signal Strength Indicator
RT	Real Time

RTP	Real-time Transfer Protocol
RTT	Round Trip Time
RU	Radio Unit
RUIF	Radio Unit Interface
RX	Receiver
RXI	Radio Access Network Aggregator
SA	Service Area
SAAL	Signaling ATM Adaptation Layer
SAI	Service Area Identifier
SAP	Service Access Point
SAS	Stand Alone SMPC or Stand Alone Serving Mobile Positioning Center
SC	Scrambling Code
SCB	Switch Core Board
SC-PQ	Single Carrier – Priority Queue
SCCP	Signaling Connection Control Part
S-CCPCH	Secondary Common Control Physical Channel
SCFE	Shared Control Function Entity
SCH	Synchronization Channel
SCTP	Streaming Control Protocol
SDU	Service Data Unit
SF	Spreading Factor
SFN	System Frame Number
SFTP	Secure File Transfer Protocol
SG	Signaling Gateway
SGSN	Serving GPRS Support Node
SHCCH	Shared Control Channel
SHO	Soft Handover
SIB	System Information Broadcast
SID	Size Index Identifier
SIM	Subscriber Identity Module
SIR	Signal to Interference Ratio
SMS	Short Message Service
SP	Signaling Point
SPB	Special Purpose Processor Board
SRB	Signaling Radio Access Bearer
SRNC	Serving RNC Controller
SRNS	Serving RN Subsystem
S-RNTI	SRNC – RNTI
S-SCH	Secondary Synchronization Channel
SSCF	Service Specific Coordination Function
SSCOP	Service Specific Connection Oriented Protocol

SSDT	Site Selection Diversity Transmission
SSH	Secure Shell
SSN	Subsystem Number
SW	Software
SWA	Software Allocation
TAS	Timing and Synchronization
TCP	TEMS Cell Planner
TCP	Transmission Control Protocol
TCTF	Target Channel Type Field
TDD	Time Division Duplex
TE	Terminal Equipment
TEID	Tunnel Endpoint Identifier
TEMS	Test Mobile Station
TF	Transport Format
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TFCS	Transport Format Combination Set
TRFRI	Transport Format Related Indicator
TRXB	Transceiver Board
TFRI	Transport Format Resource Indicator
TFS	Transport Format Set
TMA	Tower Mounted Amplifier
TME	Transfer Mode Entity
TM	Transparent Mode
TMD	Transparent Mode Data
TMSI	Temporary Mobile Subscriber Identity
TPC	Transmit Power Control
Tr	Transparent
TrCH	Transport Channel
TRFC	Transport Format Rate Combination
TSN	Transmission Sequence Number
TTI	Transmission Time Interval
TTI	Transmission Time Instant
TUB	Timing Unit Board
Tx	Transmission
UARFCN	UMTS Absolute Radio Frequency Channel Number
UDP	User Datagram Protocol
UDI	Unrestricted Digital Information
UDP	User Datagram Protocol
UE	User Equipment
UEA	UMTS Encryption Algorithm
UEH	User Equipment Handler

UETR	User Equipment Traffic Recording
UIA	UMTS Integrity Algorithm
UL	Uplink
UM	Unacknowledged Mode
UMD	Unacknowledged Mode Data
UMTS	Universal Mobile Telecommunication System
UNACK	Unacknowledgement
UNI	User Network Interface
URA	UTRAN Registration Area
UARFCN	UMTS Absolute Radio Frequency Channel Number
U-RNTI	UTRAN-RNTI
USB	Universal Serial Bus
USCH	Uplink Shared Channel
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network
VC	Virtual Channel
VCI	Virtual Channel Identifier
VLR	Visitor Location Register
VP	Virtual Path
VPI	Virtual Path Identifier
WCDMA	Wide-band Code Division Multiple Access
WTMA	WCDMA Tower Mounted Amplifier
XML	Extensible Markup Language
XP	Fan External Processor
XRES	Expected Response

7 INDEX

8 Table of Figures