HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Communications Laboratory

Jukka Valtanen

Transport Formats in UMTS Radio Network Controller's Software Implementation

Master's Thesis

Espoo, January 7, 2008

Supervisor: Docent Timo O. Korhonen

Instructor: Timo Vesterinen, M.Sc. (Eng)

HELSINKI UNIVERSITY OF TECHNOLOGY Department of Electrical and Communications Engineering		ABSTRACT OF MASTER'S THESIS
Author Jukka Valtanen	Date 7.1.2008	Pages 78
Title of thesis Transport Formats in UMTS Radio Network Controller's Software Implementation		
Degree programme Communications Engineering	Laboratory Communications Laboratory S-72	
Supervisor Docent Timo O. Korhonen		
Instructor Timo Vesterinen, M.Sc. (Eng)		

Abstract

Radio Resource Management (RRM) is an essential topic when 3G WCDMA networks are being developed. The majority of RRM related tasks are performed in the Radio Network Controller (RNC) which is situated between the base stations and the core network.

Transport format parameters define how data is exchanged between the physical layer and the data link layer. These parameters control how much data is transferred on a transport channel and how the data is coded by the physical layer. A set of transport formats associated to a transport channel is called a Transport Format Set (TFS) and the combination of currently valid transport formats is called a Transport Format Combination Set (TFCS). The theory and the usage of these parameters is the main topic in this thesis.

After the general theory part, this paper focuses on transport format related implementation in a real RNC software. The program block under investigation performs TFS and TFCS calculation for one Radio Resource Control (RRC) connection. The essential parts of the implementation are presented and the related code is improved for better efficiency and better maintainability.

In addition to investigating and improving the current implementation, also module testing of these features was carried out as a part of this thesis. The selected testing methods and designed test cases were analysed for their efficiency and code coverage properties. After this, the implementation was considered fully functional to be delivered to higher level testing. Also the improvements made to the implementation were discovered workable.

As a result of this thesis, the observed features of the program block are now fully functional and better maintainable than before. However, development work with the observed program block will continue also after the completion of this thesis.

Keywords

transport format, radio network controller, module testing

TEKNILLINEN KORKEAKOULU
Sähkö- ja tietoliikennetekniikan osasto

DIPLOMITYÖN TIIVISTELMÄ

Tekijä	Päiväys	Sivumäärä
Jukka Valtanen	7.1.2008	78

Työn nimi

Siirtoformaatit UMTS-verkon radioverkko-ohjaimen ohjelmistototeutuksessa

Tutkinto-ohjelma	Laboratorio
Tietoliikennetekniikka	Tietoliikennelaboratorio S-72

Työn valvoja

Dosentti Timo O. Korhonen

Työn ohjaaja

Timo Vesterinen, DI

Tiivistelmä

Radioresurssien hallinta (RRM) on olennainen osa kolmannen sukupolven WCDMA-radioverkkojen kehitystä. Suurin osa radioresurssien hallinnasta toteutetaan radioverkko-ohjaimessa (RNC), joka sijoittuu verkossa tukiasemien ja runkoverkon väliin.

Siirtoformaatit (engl. transport format) määrittävät, miten tietoa välitetään fyysisen ja siirtoyhteyskerroksen välillä. Nämä parametrit kontrolloivat, kuinka paljon dataa siirretään siirtokanavalla ja miten fyysinen kerros koodaa tämän datan. Yhdelle siirtokanavalle liitettyä siirtoformaattien joukkoa kutsutaan Transport Format Setiksi (TFS) ja tietyllä ajanhetkellä voimassa olevien siirtoformaattien kombinaatiota Transport Format Combination Setiksi (TFCS). Näiden parametrien takana oleva teoria ja niiden käyttö verkossa ovat tämän diplomityön pääaiheita.

Yleisen teoriaosuuden jälkeen työ keskittyy siirtoformaatteihin liittyvään ohjelmistototeutukseen RNC:ssä. Työssä tutkittu ohjelmalohko laskee TFS- ja TFCS-parametrit yhdelle radioresursseiltaan kontrolloidulle RRC-yhteydelle. Toteutuksen oleelliset osat esitellään ja tähän liittyvän ohjelmakoodin tehokkuutta ja ylläpidettävyyttä parannetaan.

Nykyisen toteutuksen tutkimisen ja parantamisen lisäksi myös näiden toiminnallisuuksien modulitestaus toteutettiin osana tätä diplomityötä. Valitut testauskäytännöt ja suunnitellut testitapaukset analysoitiin tehokkuus- ja kattavuusominaisuuksiltaan. Tämän jälkeen nykytoteutuksen katsottiin olevan täysin toimiva lähetettäväksi ylemmän tason testausta varten. Myös toteutukseen tehtyjen parannusten todettiin olevan toimivia.

Tämän diplomityön tuloksena ohjelmalohkon tutkitut ominaisuudet ovat nykyisellään täysin toimivia ja entistä paremmin ylläpidettäviä. Kehitystyö tarkastellun ohjelmalohkon parissa jatkuu kuitenkin myös tämän työn valmistumisen jälkeen.

Avainsanat

transport format, radio network controller, module testing

Acknowledgements

I would like to thank my employer for giving me the opportunity and necessary resources for writing this thesis. The whole process of writing would not have been possible without the constant support and guidance provided me by my co-workers; thank you Petri, Harri, Anssi, Olli and my instructor Timo.

I would also like to thank my supervisor docent Timo O. Korhonen, from Helsinki University of Technology, for the guidance and advisement during the process. After the clarification of the thesis topic, he helped me in getting the process started and also thereafter gave me some valuable comments on this paper.

Most of all, I would like to thank my family, my mother, father and sister, for both financial and other support and encouragement they have given me during my studies in the past few years. I am also grateful to some of my closest friends for their support and contribution: thank you Tiina, Mika, Sampo and Lasse. Last but not least FIN/ACK to Matti and Maija for keeping me amused during the writing of this thesis.

In Espoo, 7.1.2008

Contents

Abbreviations

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

3G	3rd Generation
3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
AI	Acquisition Indicator
AICH	Acquisition Indication Channel
ALCAP	Access Link Control Application Part
AMR	Adaptive Multi-Rate (speech codec)
AMR-WB	Adaptive Multi-Rate Wideband (speech codec)
ARQ	Automatic Repeat reQuest
BCH	Broadcast Channel
CCTrCH	Coded Composite Transport Channel
CN	Core Network
CPICH	Common Pilot Channel
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CS	Circuit Switched
CTFC	Calculated Transport Format Combination
DCH	Dedicated Channel
DL	Downlink (from network to UE)
DLL	Data Link Layer
DPCCH	Dedicated Physical Control Channel
DPDCH	Dedicated Physical Data Channel
DRNC	Drift RNC
DSCH	Downlink Shared Channel
DTX	Discontinuous Transmission
E-AGCH	E-DCH - Absolute Grant Channel
E-DCH	Enhanced Dedicated Channel (=HSUPA)
E-DPCCH	E-DCH - Dedicated Physical Control Channel
E-DPDCH	E-DCH - Dedicated Physical Data Channel
E-HICH	E-DCH - Hybrid ARQ Indicator Channel
E-RGCH	E-DCH - Relative Grant Channel
F-DPCH	Fractional Dedicated Physical Channel
FACH	Forward Access Channel
FBI	Feedback Information
FDD	Frequency Division Duplex
GGSN	Gateway GPRS Support Node
GMSC	Gateway MSC
GPRS	General Packet Radio Service
GSA	Global Mobile Suppliers Association
GSM	Global System for Mobile Communications
HARQ	Hybrid Automatic Repeat reQuest

HHO	Hard Handover
HLR	Home Location Register
HLS	Higher Layer Scheduling
HS-DPCCH	High-Speed Dedicated Physical Control Channel
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
HSUPA	High-Speed Uplink Packet Access
ICH	Indicator Channel
ISDN	Integrated Services Digital Network
LoCH	Logical Channel
LTE	Long Term Evolution
MAC	Medium Access Control
MBMS	Multimedia Broadcast / Multicast Service
Mcps	Megachips Per Second
ME	Mobile Equipment
MICH	MBMS Notification Indicator Channel
MSC	Mobile Services Switching Centre
MSC	Message Sequence Chart
MUT	Module Under Test
NI	(MBMS) Notification Indicator
OSI	Open Systems Interconnection
P-CCPCH	Primary Common Control Physical Channel
PCH	Paging Channel
PDU	Protocol Data Unit
PhCH	Physical Channel
PHY	Physical Layer
PI	Page Indicator
PICH	Page Indicator Channel
PLMN	Public Land Mobile Network
PRACH	Physical Random Access Channel
PRB	Program Block (especially the one analysed in this document)
PS	Packet Switched
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RACH	Random Access Channel
RAN	Radio Access Network
RANAP	Radio Access Network Application Part
RAT	Radio Access Technology
RLC	Radio Link Control
RNC	Radio Network Controller
RNS	Radio Network Subsystem
RRC	Radio Resource Control
RRM	Radio Resource Management

S-CCPCH	Secondary Common Control Physical Channel
SCH	Synchronisation Channel
SDU	Service Data Unit
SF	Spreading Factor
SGSN	Serving GPRS Support Node
SL	Signalling Link
SRNC	Serving RNC
SW	Software
TDD	Time Division Duplex
TF	Transport Format
TFC	Transport Format Combination
TFCI	Transport Format Combination Indicator
TFCS	Transport Format Combination Set
TFI	Transport Format Indicator
TFS	Transport Format Set
TPC	Transmit Power Control
TrCH	Transport Channel
TTI	Transmission Time Interval
UE	User Equipment
UL	Uplink (from UE to network)
UMTS	Universal Mobile Telecommunications System
UPCH	Uplink Common Packet Channel
USIM	UMTS Subscriber Identity Module
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network
VLR	Visitor Location Register
WCDMA	Wideband Code Division Multiple Access

Key concepts

3rd Generation (3G)
3G is commonly referred as the third generation of mobile communication standards and technology, after second generation technologies, such as Global System for Mobile Communications (GSM). 3G technologies enable network operators to offer end-users a wide range of advanced services while simultaneously achieving greater network capacity.

Physical Layer (PHY)
Physical layer is the lowest layer within the Open Systems Interconnection (OSI) reference model, including the transmission of signals and the activation and deactivation of physical connections.

Data Link Layer (DLL)
Data link layer is the layer above the physical layer within the OSI reference model, including synchronisation and some control over the influence of errors within the physical layer.

Medium Access Control (MAC) layer
The data link layer is split into four sub-layers, one of which is called the Medium Access Control (MAC) layer. The physical layer offers transport channels as a service to MAC, which furthermore offers logical channels as a service to higher layers in the model.

Logical Channel (LoCH)
Logical channels are a service provided by the MAC layer to higher layers. They describe for which purpose and what type of data is transferred between the user and the network. Logical channels can be divided into control channels and traffic channels.

Transport Channel (TrCH)
Transport channel is a channel offered by the physical layer to the MAC layer in the data link layer for data transport between peer physical layer entities. Transport channels describe how and with what characteristics data is transferred on the physical layer over the radio interface. Transport channels can be divided into dedicated channels and common channels.

Physical Channel (PhCH)
Physical channels are the actual physical layer communication streams characterised by a specific carrier frequency, scrambling code, channelisation code and other parameters.

Transport Format (TF)
Transport format is a format offered by the physical layer to the MAC layer for the delivery of a transport block set during a transmission time interval on a transport channel.

Transport Format Set (TFS)
Transport format set is a set of transport formats which is formed when the transmission rate of a transport channel varies. For example, a variable bit rate channel has a transport format set (one transport format for each rate), whereas a fixed rate channel has only a single transport format.

Transport Format Combination (TFC)
Transport format combination is a combination of currently valid transport formats on all transport channels of the mobile station. It contains one transport format from each transport channel.

Transport Format Combination Set (TFCS)
Transport format combination set is a set of transport format combinations to be used by the mobile station.

Radio Resource Management (RRM)
Radio resource management is a set of functions of a mobile communication system used for the establishment, maintenance, and release of radio connections needed by the system.

Radio Resource Control (RRC) connection
Radio resource control connection is a point-to-point bidirectional connection between radio resource control peer entities on the mobile station and the radio access network sides, respectively. The mobile station always has either zero or one RRC connection.

1 Introduction

Third generation mobile communication systems have enabled mobile operators to offer end-users a wide range of advanced services while simultaneously achieving constantly growing network capacity, shorter delay times and higher data rates. These new services require flexibility and resource optimisation from the underlying network, especially from the network components between the core network and the user equipment. As mobile systems evolve and packet switched data connections are superseding circuit switched connections, the radio network needs constant improvements to keep up with the core network.

This work examines the implementation of a program block participating in Radio Resource Management (RRM) in the Radio Network Controller (RNC) software. The focus will be in transport format related features of this program block, and some improvements to the implementation are to be designed and implemented. The main goal of this thesis is to enhance the current implementation and to improve the maintainability of the observed program block.

In parallel, module testing of these features will be carried out, analyzed and also improved. The observed features of the program block were not tested in module level testing before starting this thesis. New functionalities in the code need be verified as soon as possible and regression testing should be executed when major changes are made in the implementation.

The viewpoint in this thesis will be academic and explorative, and the literature review in the beginning of the thesis is intended only as a short induction into the actual topic. Figure 1 illustrates the RNC's position in the 3G mobile network.

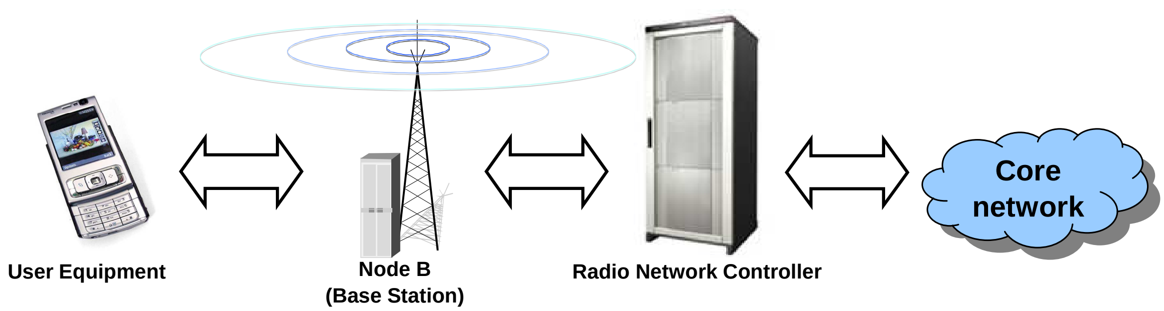

Figure 1 - Basic components in a 3G mobile network

1.1 Structure

The first part of this document, chapters 2-3, introduces the 3G radio network architecture in general, the WCDMA (Wideband Code Division Multiple Access) radio interface and its protocol architecture. This section also defines the different channel types that are used in the network: logical, transport and physical channels – all with their own specific features.

Chapter 4 introduces the theory behind transport formats and illustrates their usage in the data transmission between the physical and data link layer. This part of the thesis is heavily based on specifications and standardisation documents.

Chapter 5 focuses on transport format implementation in a real RNC software solution and points out some of the reasons why this paper was considered worth realising. This section presents the program block under investigation and the current implementation of transport formats. Some improvements to the code are implemented after the current implementation is analysed, and the analysis continues for possible further enhancements.

Chapter 6 presents the applied module testing practises in the observed environment and the set of new test cases that are used for testing transport format related functionality. Besides improving the current implementation, the main goal of this thesis is to ensure proper maintenance work and enhance module level testing of the program block in question. This part of the study is purely explorative research and it combines both technical analysis and author's own experience.

The thesis is concluded in chapter 7, references are listed in chapter 8, and some of the additional material is presented in the appendices.

The contents of this paper is written with an assumption that the reader has basic knowledge of radio networks and related terminology.

2 Third generation mobile communication networks

2.1 Evolution from 2G to 3G

Second generation (2G) mobile communication systems, such as GSM (Global System for Mobile Communications) and GPRS (General Packet Radio Service), changed profoundly the way people communicate in the 1990's and early 2000's. These systems introduced the digital world of communications and circuit switched data connections were gradually replaced by packet switched connections.

The main reason for starting the specification work for third generation cellular networks was the lack of bandwidth offered to the end-user by second generation systems. End-user requirements have increased and new Internet-based cellular services have become more and more common in the past few years. 2G air interface technologies cannot support these new requirements set by new innovative services.

UMTS (Universal Mobile Telecommunication System) networks are designed from the very beginning for flexible delivery of any type of service, where each new service does not require particular network optimisation. UMTS offers higher bit rates to end-users, lower delays, seamless mobility for both circuit-based and packet-based connections, simultaneous voice and data capability and interworking with the existing GSM/GPRS networks. (Holma & Toskala 2007, p. 9)

The evolution from 2G to 3G is still ongoing and these systems will co-exist yet a long time, but 3G has already established a very strong position in the market while new technologies are still emerging. New value added services are constantly being developed and they are brought aggressively to the customers by operators all over the world.

The new radio access technology introduced by 3G is utilising the frequency spectrum more efficiently than pre-existing technologies: the data flow and its bit rate are not dependent on time slots any more and packet type of traffic has been considered from the very beginning of the planning of the radio access method.

In this chapter, the very basics of third generation telecommunication systems and UMTS technology are presented. The starting point is the WCDMA air interface technology and the network structure in third generation mobile systems. We also introduce the UTRA (UMTS Terrestrial Radio Access) protocol model as an introduction to the following chapters, where the physical layer of the protocol model is taken under more specific analysis.

2.2 WCDMA technology and 3GPP

WCDMA technology has established itself as the most widely adopted third generation air interface technology and operators are constantly launching new UMTS networks based on this technology[1]. WCDMA specification has been created in 3GPP (the 3rd Generation Partnership Project), which is a joint standardisation project of standardisation bodies from Europe, Japan, Korea, the USA and China. 3GPP was established in December 1998 and nowadays it also has responsibility for second generation mobile systems.

Figure 2 shows the different release phases of 3GPP's standardisations. Release '99 specified the first UMTS 3G networks incorporating WCDMA air interface technology. Each new release has thereafter incorporated hundreds of new specification documents and future releases beyond the latest Release 7 are in the development under the title Long Term Evolution (LTE). Overview of the new functionality included in each release is available through 3GPP web pages[2] where all documents and specifications are also published.

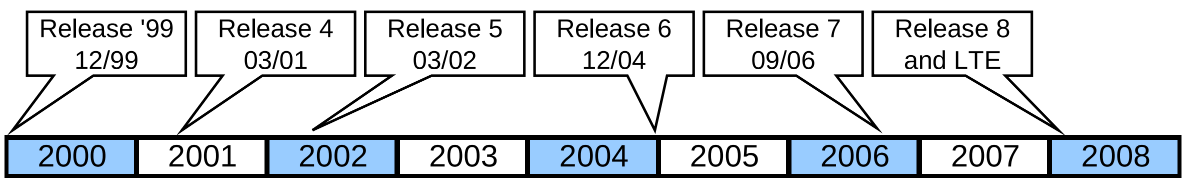

Figure 2 - 3GPP releases

In WCDMA, all users use the same frequency band simultaneously and users are assigned a specific spreading code or codes varying per transaction. If the originating bit rate is low, the signal can be spread well and the power required for transmission is low. If the originating bit rate is high, the signal cannot be spread that well and the power required for transmission will be higher. The information to be transferred is spread all over the defined frequency band as a function of time and in time-frequency graph the signal pretty much looks like noise.

[1] http://en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System#Realworld_implementations

[2] http://www.3gpp.org/specs/releases-contents.htm

There are two different WCDMA variants that are both being developed in parallel. These variants differ from each other in how the different transmission directions, i.e. uplink and downlink, are separated. In Frequency Division Duplex (FDD), the air interface transmission directions are separated by different frequencies and in Time Division Duplex (TDD) variant they are separated by time intervals. In this document, only the FDD variant will be considered if not otherwise mentioned.

Compared to second generation technologies, such as GSM, WCDMA offers high and variable bit rates to end-user, multiplexing of services with different quality requirements (delay, jitter etc.) on a single connection and the support for asymmetric uplink and downlink traffic. In WCDMA, these requirements are realised with high spectrum efficiency, larger carrier spacing, fast power control and transmit diversity. (Holma & Toskala 2007, pp. 3-4)

2.3 Radio access network architecture

2.3.1 Logical network elements and interfaces

In this chapter, a wide overview of the UMTS system architecture, including the logical network elements and interfaces, is presented. The UMTS system mainly utilises the same well-known architecture that has been used by all main second generation systems, but there are some significant differences introduced as well.

The UMTS system consists of a number of logical network elements, which can be grouped based on similar functionality or based on which sub-network they belong to. In this context, the first one of these approaches is preferred: each network element has its own logical functionality and the interfaces between the elements are well defined. The high-level system architecture of UMTS is presented in Figure 3.

The Radio Access Network (RAN, UMTS Terrestrial RAN = UTRAN) handles all radio-related functionality and interfaces the Core Network (CN) through the Iu interface. The CN is responsible for switching and routing calls and data connections to externals networks. The User Equipment (UE) completes the systems and creates the Uu interface between the end-user and the radio access network.

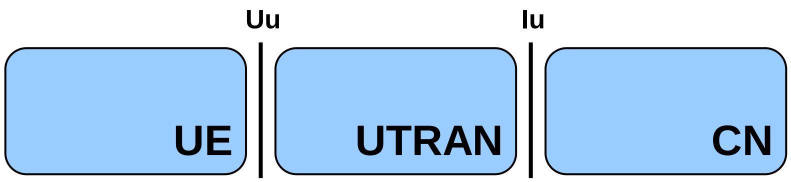

Figure 3 - UMTS high-level system architecture

Now when we go deeper in the system, we can define the different logical network elements. Figure 4 shows these elements in a simple UMTS Public Land Mobile Network (PLMN), a network operated by a single operator. This PLMN is then connected to other networks, such as ISDN, PSTN and the Internet.

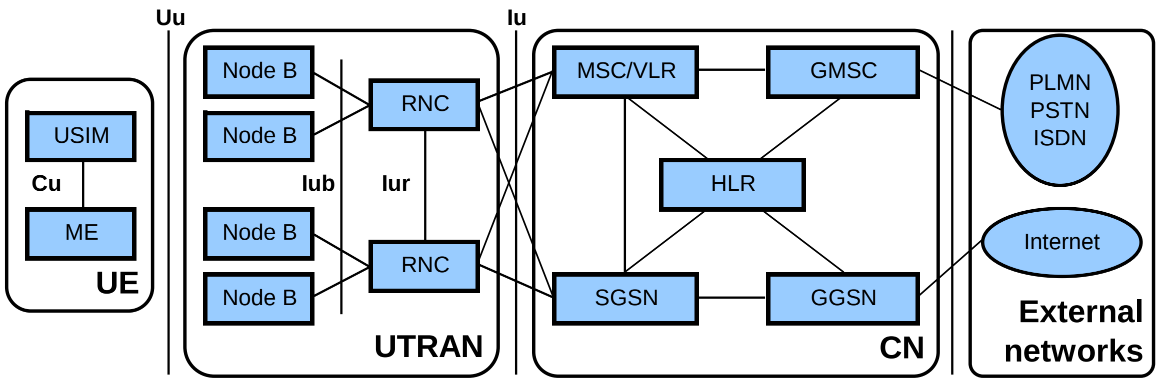

Figure 4 - Network elements in a PLMN

The UE consists of two logical entities: the Mobile Equipment (ME) is the actual radio terminal used for radio communication over the Uu interface, whereas the UMTS Subscriber Identity Module (USIM) is a smartcard that contains subscriber identity information and performs authentication algorithms. The interface between the USIM and ME is called the Cu interface.

The UTRAN consists of two logical entities and interfaces between them. The base stations, which are also called Node Bs, convert the data stream from the Uu interface to the Iub interface. In Release ’99 the only radio resource management related task of the Node B was the inner loop power control, but some of the latest features of WCDMA have introduced several new functionalities for the Node B to handle. These include for example packet scheduling, resource allocation, congestion control and retransmission handling in some cases.

The Radio Network Controller (RNC) is a network element that owns and controls radio resources in its domain, i.e. the base stations connected to it. Because this thesis will mainly focus on this particular network element, chapter 2.3.2 is dedicated for introducing the RNC and its different roles in more detail.

In the core network, the main elements are Home Location Register (HLR), Mobile Services Switching Centre / Visitor Location Register (MSC/VLR), Gateway MSC (GMSC), Serving GPRS Support Node (SGSN) and Gateway GPRS Support Node (GGSN), which are not further investigated in this context.

In Figure 4, the external networks are divided into two groups: Circuit Switched (CS) networks provide circuit switched connections such as telephony services, whereas Packet Switched (PS) networks provide packet switched connections for packet data services such as the Internet. These external networks are not either in the scope of this document.

As mentioned by Holma and Toskala (2007, p. 67), the UMTS standards are structured so that the internal functionality of each network element is not specified in detail. Instead, only the interfaces between the logical network elements have been defined, which allows manufacturer specific implementations in all network elements. Furthermore, open interfaces give UMTS operators the possibility of acquiring different network elements from different manufacturers.

2.3.2 RNC

The RNC operates as a service access point for all services that UTRAN provides the core network – for example management of connections to the UE. One RNC and all base stations connected to it via the Iub interface form a Radio Network Subsystem (RNS). The UTRAN then consists of one or more RNSs, where the RNCs may be connected to each other via the Iur interface. The RNC can be connected to multiple core domains via the Iu interface as shown in Figure 4. These can be further divided into Iu-CS and Iu-PS interfaces, depending on the core network type.

In order to understand the behaviour and functionality of the RNC in different situations, three different RNC roles have been defined:

- the Controlling RNC (CRNC)
- the Serving RNC (SRNC)
- the Drifting DRNC (DRNC)

The RNC that has control over one Node B is called the Controlling RNC (CRNC) of that Node B. The Controlling RNC is responsible for the load and congestion control of its own cells, and also executes admission control and code allocation for new radio links that are established in those cells. (Holma & Toskala 2007, p. 71)

If one mobile-UTRAN connection uses resources from more than one RNC, the RNCs involved can have two different logical roles. The Serving RNC (SRNC) is the RNC that terminates the Iu link for the transport of user data and the corresponding Radio Access Network Application Part (RANAP) signalling to and from the core network. The SRNC also terminates the Radio Resource Control (RRC) signalling on Layer 3 between the UE and UTRAN.

One UE connected to UTRAN has one and only one SRNC. All basic radio resource management functionalities are executed in the SRNC, but the ever-increasing delay and bit rate requirements are constantly moving RRM related procedures towards the base stations.

The Drift RNC (DRNC) is any RNC, other than the SRNC, that controls cells used by the mobile. DRNC is responsible for supporting the SRNC with radio resource management. DRNC may perform macrodiversity combining and splitting, but it routes the user plane data transparently to the SRNC between the Iub and Iur interfaces. One UE may have zero, one or possibly even more DRNCs. (Holma & Toskala 2007, p. 71)

When a new cell is added to the active set (radio links simultaneously involved in a connection) of a UE and that cell is connected to the network through an RNC other than the SRNC, that new RNC takes the “drift” role. Later on, when the user equipment moves around and the active set gets updated accordingly, it is also possible to relocate the SRNC so that the DRNC changes to act as the new SRNC.

3 UTRA radio interface protocol architecture

3.1 Introduction to the protocol architecture

3GPP technical specification TS 25.201, “Physical layer – general description”, (2007, p. 7) illustrates the radio interface protocol architecture around the physical layer with a figure similar to Figure 5 below. The radio interface means the Uu point between the User Equipment (UE) and Radio Access Network (RAN).

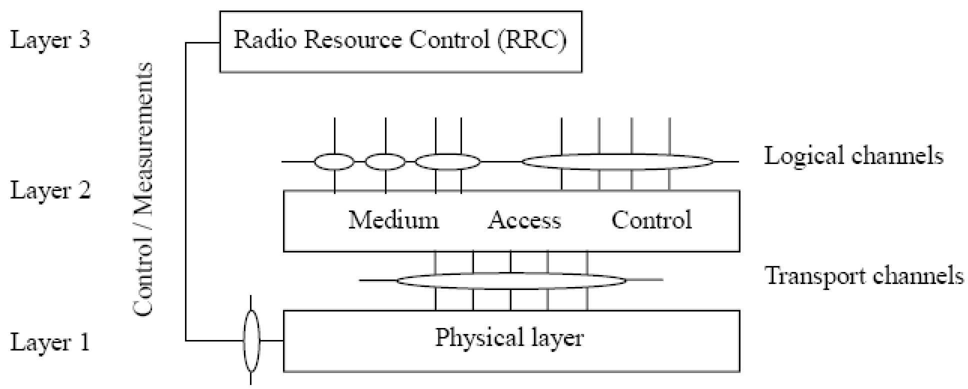

Figure 5 - Radio interface protocol architecture

The radio interface is composed of Layers 1, 2 and 3. The lowest layer, Layer 1, is the physical layer, which is based on WCDMA technology and described in 3GPP specification series TS 25.200. Layer 2 above it is split into four sub-layers: Medium Access Control (MAC), Radio Link Control (RLC), Packet Data Convergence Protocol (PDCP) and Broadcast/Multicast Control (BMC). Furthermore, Layer 3 is divided into control plane and user plane and, together with Layer 2, described in 3GPP specification series TS 25.300.

Layer 1 in the protocol model takes care of the actual transmission of data across the radio path. In the well-known OSI (Open Systems Interconnection) reference model, the physical layer is the lowest data transmission layer and it only provides the means of transmitting raw bits over the physical data link. The physical layer also includes tasks like forward error correction (channel coding), interleaving, error detection (CRC), closed loop power control and synchronisation (3GPP TS 25.201 2007, p. 7). The physical layer and physical channels are introduced in chapters 3.2 and 3.5, respectively.

The physical layer interfaces the MAC sub-layer of Layer 2 (the data link layer) and offers so called transport channels (TrCH) as a service to MAC. A transport channel is characterised by how the information is transferred over the radio interface and these are the main topic in chapter 3.4. Furthermore, the MAC layer offers logical channels as a service to the RLC sub-layer of Layer 2. A logical channel is characterised by the type of information transferred and these are shortly covered in chapter 3.3.

The physical layer also interfaces the Radio Resource Control (RRC) layer of Layer 3 (the network layer), which can be used for controlling the physical layer. The RRC terminates in the UTRAN and this protocol contains all procedures to control, modify and release Radio Bearers (RB). The RRC messages use radio bearer services offered by Layer 2 for transport.

3.2 WCDMA physical layer and air interface

When comparing different cellular systems with each other, the physical layer of the radio interface typically contains most of the differences and is therefore the most interesting part of the study. In the OSI reference model, the physical layer is the lowest layer and it includes the transmission of signals and the activation and deactivation of physical connections.

The physical layer has a major impact on equipment complexity with respect to the required baseband processing power in the terminal and in the base station equipment. WCDMA technology also introduces new challenges to the implementation of the physical layer. As third generation systems are wideband from the service point of view as well, the physical layer needs to be designed to support various different services. More flexibility is needed for future service introduction, too. (Holma & Toskala 2007, p. 91)

The physical layer offers data transport services to higher layers and it is designed to support variable bit rate transport channels, to offer so-called bandwidth-on-demand[3] services and to be able to multiplex several services within the same Radio Resource Control (RRC) connection. (Soldani & Li & Cuny 2006, p. 113)

According to 3GPP TS 25.201 (2007, pp. 7-8), the physical layer is expected to perform (among others) the following functions:

- Macrodiversity distribution / combining and soft handover execution
- Error detection on transport channels and error indication to higher layers
- Multiplexing of transport channels and demultiplexing of Coded Composite Transport Channels (CCTrCHs)
- Rate matching and mapping of CCTrCHs onto physical channels
- Modulation and spreading / demodulation and despreading of physical channels
- Frequency and time (chip, bit, slot, frame) synchronisation

[3] Bandwidth-on-demand is a data communication technique for providing additional capacity on a link as necessary to accommodate bursts in data traffic or other special requirements. The technique is commonly used in different networks to temporarily boost the capacity of a link.

The basic idea in WCDMA is that the signal to be transferred over the radio path is formed by multiplying the original baseband digital signal with another signal, which has a much greater bit rate. This operation is called channelisation and the number of chips per data symbol is called the Spreading Factor (SF).

We need to make a clear separation between the different kinds of bits in WCDMA. One bit of baseband digital signal, the actual information, is called a symbol. On the other hand, one bit of code signal used for signal multiplying is called a chip. The code signal bit rate, i.e. the chip rate, is fixed in WCDMA being 3.84 million chips per second (3.84 Mcps). The symbol rate indicates how many data symbols are transferred over the radio path and it is expressed as kilosymbols per seconds (ks/s).

In scrambling operation, a scrambling code is applied to the signal, which makes signals from different sources separable from each other. Scrambling is used on top of spreading and it separates terminals or base stations from each other. The symbol rate is not affected by the scrambling operation. (Holma & Toskala 2007, p. 41; 96)

Figure 6 below shows the relationship of these terms and operations.

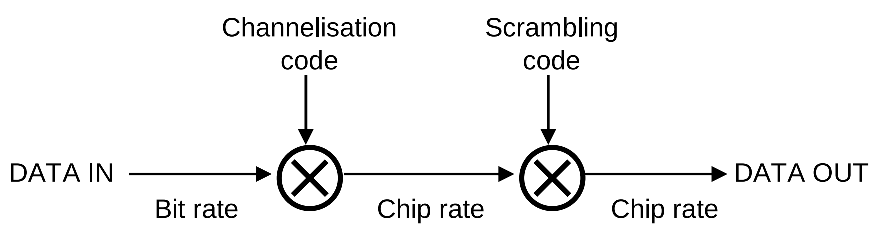

Figure 6 - Spreading and scrambling

The FDD variant of WCDMA employs the frequency range of 1920–1980 MHz for the uplink and 2110–2170 MHz for the downlink. Recently some other frequencies have been adopted for WCDMA usage, too. Both uplink and downlink transmission are organised in the time domain in *radio frames*, which have a duration of 10 ms. As the chip rate is 3.84 Mcps, a radio frame contains 38 400 chips. Each frame is further divided into 15 *slots* and the length of a slot corresponds to 2 560 chips. A *sub-frame* is the basic time interval for E-DCH and HS-DSCH transmission and related signalling at the physical layer. The length of a sub-frame corresponds to 3 slots (7680 chips) or 2 ms. (Tanner & Woodard, 2004, p. 69)

3.3 Logical channels

Logical channels (LoCH) are a service provided by the MAC layer to higher layers. They describe for which purpose and what type of data is transferred between the user and the network. Logical channels can be divided into control channels and traffic channels. Based on descriptions by Holma & Toskala (2007, pp. 143-144) and Soldani & Li & Cuny (2006, pp. 109-111), the logical channels are shortly presented in Tables 1 and 2.

Table 1 - Logical control channels and their descriptions

Logical control channel	***Description***
Broadcast Control Channel (BCCH)	Downlink channel for broadcasting system control information.
Paging Control Channel (PCCH)	Downlink channel for transmitting paging information.
Dedicated Control Channel (DCCH)	Point-to-point bidirectional channel for transmitting dedicated control information between the network and UEs.
Common Control Channel (CCCH)	Bidirectional channel for transmitting control information between the network and UEs.
MBMS point-to-multipoint Control Channel (MCCH)	Point-to-multipoint downlink channel for transmitting control information from the network to the UEs that receive the MBMS.
MBMS point-to-multipoint Scheduling Channel (MSCH)	Point-to-multipoint downlink channel for transmitting scheduling control information from the network to the UEs that receive the MBMS, for one or several MTCHs carried on a CCTrCH.

Table 2 – Logical traffic channels and their descriptions

Logical traffic channel	***Description***
Dedicated Traffic Channel (DTCH)	Point-to-point uplink or downlink channel, dedicated to one UE, for transmitting user information.
Common Traffic Channel (CTCH)	Point-to-multipoint downlink channel for transmitting dedicated user information for all or a group of specified UEs.
MBMS point-to-multipoint Traffic Channel (MTCH)	Point-to-multipoint downlink channel for transmitting traffic data from the network to the UEs that receive the MBMS.

Logical channels include user data and different tasks the network and the terminal should perform in different moments of time. In the MAC layer, logical channels are mapped to transport channels and the MAC layer is also responsible for selecting an appropriate transport format (TF) for each transport channel depending on the instantaneous source rates of the logical channels (Holma & Toskala 2007, p. 142). This topic is further investigated later, but in this context, logical channels are not covered in more detail.

3.4 Transport channels

3.4.1 General on transport channels

Transport channels (TrCH) are a service offered by the physical layer to the MAC layer in the data link layer. As logical channels describe what is transferred, transport channels describe how and with what characteristics data is transferred on the physical layer over the radio interface. One or more logical channels can be mapped to a given transport channel by the MAC.

The RNC basically deals only with transport channels and thus the following chapters are essential for the reader to understand before moving to chapters 4, 5 and 6. The mapping of logical channels onto transport channels is not in the scope of this thesis but the mapping of transport channels onto physical channels is described in chapter 3.6.

There are two types of transport channels: dedicated channels and common channels. A dedicated channel is a resource reserved for a single UE only, whereas a common channel is a resource divided between all or a specific group of users in a cell.

The following channel descriptions are based on references Holma & Toskala (2007, pp 92-95), Soldani & Li & Cuny (2006, pp. 113-115), and 3GPP TS 25.211 (2007, pp. 8-10).

3.4.2 Dedicated transport channels

There are currently two types of dedicated transport channels described in the 3GPP specifications and these are shortly presented in the following.

3.4.2.1 DCH – Dedicated Channel

The Dedicated Channel (DCH) was originally in Release ‘99 the only dedicated transport channel in the specifications. The DCH is a downlink or uplink transport channel, which is transmitted over the entire cell or over only a part of the cell using e.g. beam-forcing antennas and varying antenna weights with adaptive antenna systems.

The DCH carries all the information, excluding the HSPA traffic on HS-DSCH and E-DCH, intended for a given user, coming from layers above the physical layer. This information includes both the actual service data, such as speech frames, as well as higher layer control information, such as measurement reports and control commands from the terminal. The DCH supports fast data rate change on a frame-by-frame basis.

From this thesis' point of view, the DCH is probably the most important transport channel, because the transport format related calculation introduced later in the document is heavily based on currently active DCH channels, their properties and analysis.

3.4.2.2 E-DCH – Enhanced Dedicated Channel

The Enhanced Dedicated Channel (E-DCH) is an uplink transport channel introduced together with High Speed Uplink Packet Access (HSUPA) and it is available in 3GPP Release 6 and later releases. HSUPA is a new feature of WCDMA that provides data speeds of up to 5.8 Mbps in the uplink direction. HSUPA achieves its high performance through more efficient scheduling in the base station and faster retransmission control.

The E-DCH operates in the uplink direction only, with the possibility of changing data rate each Transmission Time Interval (TTI). Because of strict delay requirements, the transport format related calculation for the E-DCH transport channel is performed in the Node B instead of the RNC.

3.4.3 Common transport channels

There are currently five types of common transport channels described in the 3GPP specifications and these are shortly presented in the following.

Two of the former seven transport channels, Uplink Common Packet Channel (CPCH) and Downlink Shared Channel (DSCH), have been removed from the specifications by 3GPP RAN WG1 in early 2005 from Release 5 onwards. The use of these two transport channels was in any case optional and could have been decided by the network. In practice, the CPCH was not implemented in any of the networks and the DSCH was replaced with HSDPA. (3GPP RP-050248 2005; 3GPP RP-050250 2005; Holma & Toskala 2007, p. 95)

3.4.3.1 BCH – Broadcast Channel

The Broadcast Channel (BCH) is used to transmit information specific to the whole UTRA network or for a given cell. Typical data on the broadcast channel includes the available random access codes, cell access slots, and cell-type transmit diversity methods. The BCH is intended for downlink traffic only.

All terminals in the cell area need to decode the BCH correctly in order to register to the cell. Thus the information rate on the BCH is limited by the ability of old and other low-end terminals to decode the data flow, resulting in a low and fixed data rate for this channel. The spreading factor for the BCH is fixed to 256 and also the slot format is fixed. Furthermore, the BCH needs to be transmitted with a relatively high power.

3.4.3.2 FACH – Forward Access Channel

The Forward Access Channel (FACH) is used to carry downlink control information to a terminal known to be located in the cell area. The FACH is used, for example, after a random access message has been received by the BTS and control information needs to be sent to the terminal. Furthermore, it can be used to carry small amounts of downlink packet data with the possibility of changing data rate every TTI.

The FACH is always transmitted over the entire cell area and there can be several FACH channels in a cell. One of these channels must have such a low bit rate that it can be received by all terminals in the cell area.

3.4.3.3 PCH – Paging Channel

The Paging Channel (PCH) is a downlink transport channel that carries paging procedure related data, which is used when the network needs to initiate communication with a specific terminal, for example when a speech call is coming to the terminal. The paging message can be transmitted in a single cell or in up to a few hundred cells, depending on the system configuration and location area size.

The PCH is always transmitted over the entire cell and all terminals in the cell area must be able to receive the paging information. The transmission of the PCH is associated with the transmission of physical-layer generated Paging Indicators, see chapter 3.5.4.

3.4.3.4 RACH – Random Access Channel

The Random Access Channel (RACH) is an uplink transport channel used to carry control information from the terminals in the cell area. This is used, for example, when a terminal requests to set up a connection to the network.

The RACH is always received from the entire cell and thus in practice the data rates need to be rather low. The RACH is characterised by a collision risk and by being transmitted using open loop power control. As with the FACH presented earlier, it is also possible to transmit small amounts of packet data on the RACH.

3.4.3.5 HS-DSCH – High Speed Downlink Shared Channel

The High Speed Downlink Shared Channel (HS-DSCH) is a downlink transport channel introduced together with High Speed Downlink Packet Access (HSDPA) and it is available in 3GPP Release 5 and later releases. HSDPA is a key feature of WCDMA that provides data speeds of up to 14.4 Mbps in the downlink direction. HSDPA achieves its high performance through adaptive modulation and coding, fast packet scheduling at the base station, and fast retransmissions known as Hybrid Automatic Repeat Request (HARQ).

Unlike the E-DCH, the HS-DSCH is shared by several UEs in the cell area. It is always associated with one downlink Dedicated Physical Channel (DPCH) and one or several High Speed Shared Control Channels (HS-SCCH). The HS-DSCH is transmitted over the entire cell or over only a part of the cell using e.g. beam-forming antennas.

3.5 Physical channels

3.5.1 General on physical channels

Physical channels (PhCH) are the actual physical layer communication streams characterised by a specific carrier frequency, scrambling code, channelisation code (optional), start and stop time (duration) and, on the uplink, relative phase (0 or $\pi/2$).

In the following, physical channels are introduced only by their names but more comprehensive descriptions can be found in Appendix 1, Tables 1-3, and in the author's special assignment (Valtanen 2007). Physical channels are described here only to the extent that is necessary for understanding the theory and analysis related to transport formats. The theory is largely based on 3GPP TS 25.211 (2007, pp. 9-40).

3.5.2 Uplink physical channels

3.5.2.1 Dedicated uplink physical channels

There are five types of dedicated uplink physical channels:

- the uplink Dedicated Physical Data Channel (uplink DPDCH)
- the uplink Dedicated Physical Control Channel (uplink DPCCH)
- the uplink E-DCH Dedicated Physical Data Channel (uplink E-DPDCH)
- the uplink E-DCH Dedicated Physical Control Channel (uplink E-DPCCH)
- the uplink Dedicated Physical Control Channel associated with HS-DSCH transmission (uplink HS-DPCCH)

These uplink dedicated physical channels can be divided in two parts: those that include TFCI (e.g. for several simultaneous services) and those that do not include TFCI (e.g. for fixed rate services). The case is the same with the downlink direction. The UTRAN determines if a TFCI should be transmitted and it is mandatory for all UEs to support the use of TFCI both in the uplink and in the downlink.

3.5.2.2 Common uplink physical channels

Currently there is only one common uplink physical channel:

- the Physical Random Access Channel (PRACH)

This channel is used to carry the RACH transport channel. The random-access transmission is based on a Slotted ALOHA approach with fast acquisition indication. The random-access transmission consists of one or several preambles of length 4096 chips and a message of length 10 ms or 20 ms.

3.5.3 Downlink physical channels

3.5.3.1 Dedicated downlink physical channels

There are four types of dedicated downlink physical channels:

- the downlink Dedicated Physical Channel (downlink DPCH; basically a time multiplex of a downlink DPDCH and a downlink DPCCH, compare to the uplink direction in chapter 3.5.2.1)
- the Fractional Dedicated Physical Channel (F-DPCH)
- the E-DCH Relative Grant Channel (E-RGCH)
- the E-DCH Hybrid ARQ Indicator Channel (E-HICH)

These downlink dedicated physical channels can be divided into two parts: those that include TFCI (e.g. for several simultaneous services) and those that do not include TFCI (e.g. for fixed rate services). The case is the same with the uplink direction. The UTRAN determines if a TFCI should be transmitted and it is mandatory for all UEs to support the use of TFCI both in the uplink and in the downlink.

3.5.3.2 Common downlink physical channels

There are ten types of common downlink physical channels:

- the Common Pilot Channel (CPICH)
- the Primary Common Control Physical Channel (P-CCPCH)
- the Secondary Common Control Physical Channel (S-CCPCH)
- the Synchronisation Channel (SCH)
- the Acquisition Indicator Channel (AICH)
- the Paging Indicator Channel (PICH)
- the High Speed Shared Control Channel (HS-SCCH)
- the High Speed Physical Downlink Shared Channel (HS-PDSCH)
- the E-DCH Absolute Grant Channel (E-AGCH)
- the MBMS Indicator Channel (MICH)

The main difference between the Primary and Secondary CCPCH is that the transport channel mapped to the P-CCPCH (BCH) can only have a fixed predefined transport format combination, whereas the S-CCPCH supports multiple transport format combinations using the TFCI field in its frame structure (see Appendix 2).

3.5.4 Indicators

In addition to these channel types, there are also so-called indicators specified by 3GPP. Indicators are means of fast low-level signalling entities which are transmitted without using information blocks sent over transport channels. The indicators defined are Acquisition Indicator (AI), Page Indicator (PI) and MBMS Notification Indicator (NI). Their mapping to indicator channels is channel specific and they are transmitted on those physical channels that are indicator channels. (3GPP TS 25.211, 2007, p. 9)

3.6 Mapping of transport channels onto physical channels

The mapping of transport channels onto physical channels is summarised in Figure 7.

Transport channels	Physical Channels
DCH	Dedicated Physical Data Channel (DPDCH)
	Dedicated Physical Control Channel (DPCCH)
	Fractional Dedicated Physical Channel (F-DPCH)
E-DCH	E-DCH Dedicated Physical Data Channel (E-DPDCH)
	E-DCH Dedicated Physical Control Channel (E-DPCCH)
	E-DCH Absolute Grant Channel (E-AGCH)
	E-DCH Relative Grant Channel (E-RGCH)
	E-DCH Hybrid ARQ Indicator Channel (E-HICH)
RACH	Physical Random Access Channel (PRACH)
	Common Pilot Channel (CPICH)
BCH	Primary Common Control Physical Channel (P-CCPCH)
FACH	Secondary Common Control Physical Channel (S-CCPCH)
PCH	Synchronisation Channel (SCH)
	Acquisition Indicator Channel (AICH)
	Paging Indicator Channel (PICH)
	MBMS Notification Indicator Channel (MICH)
HS-DSCH	High Speed Physical Downlink Shared Channel (HS-PDSCH)
	HS-DSCH-related Shared Control Channel (HS-SCCH)
	Dedicated Physical Control Channel (uplink) for HS-DSCH (HS-DPCCH)

Figure 7 - The mapping of transport channels onto physical channels (from 3GPP TS 25.211, 2007)

3.7 UE operational modes

In conclusion of this third chapter, the operational modes of a UE are presented shortly in the following. These modes are essential to understand before §the transport format related analysis is presented in chapters 4 and 5.

There are two basic operational modes of a UE: *idle mode* and *connected mode*. The connected mode can be further divided into four different service states, so-called RRC states, which define what kind of physical channels the mobile is using. These operational modes and RRC states (with their allowed transitions) are presented in Figure 8.

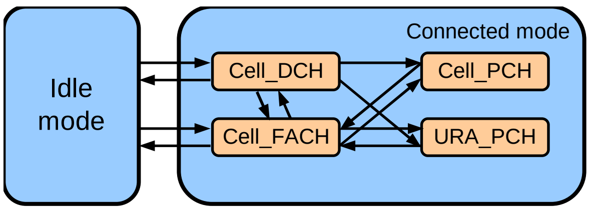

Figure 8 – Operational modes of a UE and RRC states in connected mode

After the mobile station is switched on, it selects the network to contact, looks for a suitable cell of the chosen PLMN and tunes to its control channel. Now the UE is in idle mode and the procedure is called "camping on a cell". In idle mode, the UE is able to receive system information and cell broadcast messages. The UE stays in idle mode until it needs to establish an RRC connection for more advanced services. (Holma & Toskala 2007, p. 154)

In Cell_DCH state, a dedicated physical channel is allocated to the UE in the uplink and downlink. In Cell_FACH state, no dedicated physical channel is allocated to the UE, but transport channels RACH and FACH are used instead for transmitting both signalling messages and small amounts of user-plane data. In Cell_PCH state, the UE can be reached only via the PCH transport channel. The UE also listens to system information on BCH in this state. The URA_PCH state is almost similar to Cell_PCH state, but there are some differences in cell reselection procedures. (Holma & Toskala 2007, p. 154-155)

From this thesis' point of view the Cell_DCH state is the most important RRC state. These states and terms will be referred in chapter 5.3, where the implementation of a RNC software program block is investigated in a more detailed level.

4 Transport formats

4.1 Data transmission between PHY and MAC

As already mentioned in earlier chapters, the physical layer offers its services to higher layers by transport channels, which describe with which properties (data rate, channel coding etc.) bits are transmitted over the air. These transport channels are accessed by the MAC protocol, which belongs to Layer 2 in the UTRA protocol model. The characteristics of a transport channel are defined by its transport format or transport format set, specifying the physical layer processing to be applied to the transport channel in question and any service-specific rate matching as needed. (3GPP TS 25.302 2007, p. 11)

The data transfer between MAC and PHY is organised by the transmission of so-called transport blocks. A *transport block* is the basic unit of data exchanged between MAC and PHY and every transport block belongs to one and only one transport channel. In RRC signalling, a transport block corresponds to RLC PDU (Protocol Data Unit). Each transport block can be given its own Cyclic Redundancy Check (CRC) for error detection by Layer 1. (Tanner & Woodard 2004, pp. 125-126)

To have an optimised size of transmitted data unit e.g. for error checks and different higher layer protocols, several transport blocks can be transferred at the same time on the same transport channel between MAC and PHY. The set of all transport blocks exchanged at the same time on one transport channel is called a *transport block set*.

Transport blocks and transport block sets can have several characteristics, which are described by the following attributes:

- transport block size
- transport block set size
- transmission time interval (TTI)
- error protection scheme (coding type and coding rate)
- size of CRC error detection/protection method
- rate matching parameters

All transport blocks within one transport block set have a fixed transport block size, but the size can vary between different transport block sets. The transport block set size indicates the total number of bits in that particular transport block set, i.e. the number of transport blocks in the set multiplied by the transport block size. (Tanner & Woodard 2004, p. 126)

The TTI value defines the time interval between two subsequent transport block set transfers between MAC and PHY. In earlier 3GPP releases, the TTI value was always an integer multiple of the radio frame length (10 ms), the possible values being 10, 20, 40 or 80 ms. In recent releases, together with HSDPA and HSUPA, a value of 2 ms TTI has also been introduced to increase the data rates and to satisfy the tighter delay requirements. With a shorter TTI value the data transmission can be controlled more efficiently.

The rate matching parameters indicate how the transmission process is carried out so that the block size matches the radio frames. Rate matching will either repeat bits to increase the rate or puncture bits to decrease the rate. (Tanner & Woodard 2004, p. 142)

The concept of transport blocks and transport block sets is illustrated below in Figure 9.

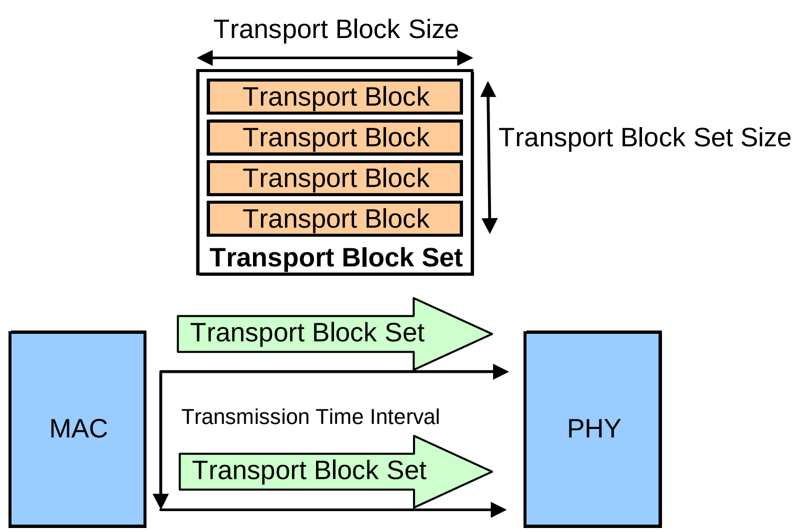

Figure 9 - Transport block, transport block set and transmission time interval (TTI)

4.2 Transport format sets

4.2.1 General on transport format sets

For an established transport channel different transport block and transport block set attributes can be used. When the MAC layer sends a transport block to the physical layer it has to indicate with which attributes the transport block shall be sent forwards. The same applies when the physical layer receives data and sends it up to the MAC layer for further processing.

For this purpose a so-called Transport Format (TF) parameter set is used. Transport format is a format applied to a transport block set on a given transport channel for a given TTI. This parameter controls how much data is transferred on the transport channel in that particular transmission time interval and how the data is coded etc. by the physical layer. (Tanner & Woodard 2004, p. 126)

The transport format constitutes of two parts. The *semi-static part* includes parameters that are configured by the RRC layer of Layer 3 in the protocol model. These parameters are fixed for all transport block sets of the corresponding transport channel until the RRC parameters are reconfigured. The *dynamic part* includes parameters whose value can change from one transport block set to another. It is the MAC layer's responsibility to choose an appropriate set of parameters from the allowed values for the transmission. Also the dynamic part of the transport format is configured by the RRC layer.

A Transport Format Set (TFS) is defined as the set of transport formats associated to a transport channel. The representation of a specific transport format within a TFS is called a Transport Format Indicator (TFI). A TFS is formed when the transmission rate of the transport channel varies and thus it includes multiple parameter sets for the dynamic part of the transport format. For example, a variable rate DCH has a transport format set (one transport format for each rate), whereas a fixed rate DCH has only a single transport format. The semi-static part of all transport formats are the same within a TFS.

The framework of a transport format set is illustrated on the next page in Figure 10.

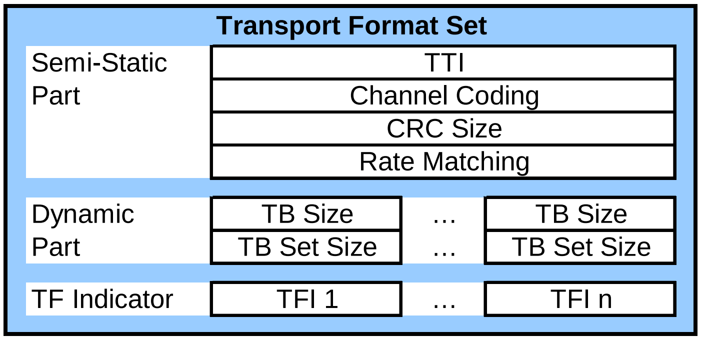

Figure 10 - The framework of a transport format set

Effectively the transport block size and transport block set size together with the TTI form the instantaneous bit rate of a transport channel. Variable bit rate services may be achieved e.g. by changing the transport block size and the transport block set size between each TTI. (3GPP TS 25.302 2007, p. 31)

All of these parameters actually control the Quality of Service (QoS) of the transport channel, as the transport format has an effect on the maximum data rate, transmission delay, error protection and error detection. However, extensive switching between TFSs will not automatically grant an effective user data throughput and higher layer protocols have a major impact on the performance (Heier & Malkowski 2002).

4.2.2 HSPA

The above-mentioned theory applies only for transport channels other than HS-DSCH and E-DCH. For these two high speed channels the transport format consists of three parts – one *dynamic* part, one *semi-static* part and one *static* part. The transport format for HS-DSCH and E-DCH is always explicitly signalled. For HS-DSCH the dynamic part of the transport format includes the modulation scheme used and the TTI value is fixed to 2 ms. For E-DCH both TTI of 2 ms and 10 ms are supported. (3GPP TS 25.302, 2007, pp. 28-31)

These new transport channel types do not affect the transport format related analysis introduced later in this document because the scheduling and transport format calculation for HSDPA and HSUPA takes place in the Node B instead of the RNC. Thus faster scheduling algorithms are possible, shorter frame sizes can be used and the so-called Adaptive Modulation and Coding (AMC) scheme can be effectively utilised according to the variations in rapidly changing channel conditions.

4.3 Transport format combination sets

4.3.1 General on transport format combination sets

As earlier presented in chapter 3, Layer 1 can multiplex several transport channels together for transmitting these channels simultaneously. For each transport channel, there is a list of possible transport formats defined in the TFS, but at a given time instant not all combinations of these may be submitted to Layer 1.

3GPP technical specification TS 25.301, "Radio interface protocol architecture", (2007) defines a Transport Format Combination (TFC) as "an authorised combination of the combination of currently valid transport formats that can be submitted simultaneously to Layer 1 for transmission on a Coded Composite Transport Channel (CCTrCH) of a UE, i.e. containing one transport format from each transport channel".

The CCTrCH is a data stream given by encoding and multiplexing one or several transport channels together to be transmitted on one or a set of physical channels. This is done in order to use the physical resources as efficiently as possible. There are, however, some restrictions on the types of transport channels that can be multiplexed onto a single CCTrCH. (Tanner & Woodard 2004, p. 127)

Furthermore, a Transport Format Combination Set (TFCS) is defined as a set of transport format combinations on a CCTrCH. The TFCS is what is actually given to the MAC layer for controlling the transmission. The assignment of a suitable TFCS is done in Layer 3 in the RNC – or correspondingly in the UE. When scheduling and mapping the data transmission onto the physical layer, the MAC layer chooses between the different TFCs given in the TFCS and thereby has control over the dynamic part of the transport format.

The representation of the current transport format combination used on the CCTrCH is called a Transport Format Combination Indicator (TFCI). The TFCI is transmitted in the physical control channel to inform the receiver which TFC was transmitted and which transport channels are active for the current frame. (Tanner & Woodard 2004, p. 127)

4.3.2 HSPA

As with transport format sets, also transport format combination sets have a different approach when it comes to high-speed data channels, i.e. HSDPA and HSUPA. As mentioned earlier, the packet scheduling with these channels is moved from the RNC to the Node B and also new protocols to the MAC layer have been introduced.

With the introduction of HSDPA, an additional MAC-hs layer was installed in the Node B to take care of faster retransmissions. The MAC-es layer was introduced together with HSUPA and it is terminated in the RNC to cover for packet re-orderings. The MAC-e layer, also introduced together with HSUPA, is terminated in the Node B and it is to handle the ARQ functionality, packet scheduling and priority handling. (Holma & Toskala 2007, p. 406)

In HSUPA, the E-TFC (Enhanced Uplink Transport Format Combination) is the counterpart for the TFC with regular Release ’99 and HSDPA channels. The TFC selection for these normal channels is made prior to E-TFC selection after which the UE builds up its own E-TFC restriction list. Based on transmission power resources and any bit rate limitations coming from the Node B, the UE selects the best E-TFC from the restriction list, trying to maximise the data throughput. (Björninen 2006)

The E-TFC contains only one transport block, but the new MAC layers introduce some additional overhead bits to the transport block. The HSUPA functionality and E-TFC related theory is further discussed by e.g. Holma & Toskala (2007, pp. 403-421) and Björninen (2006).

4.4 Example of transport formats

For the conclusion of the theory part of this thesis, an example of transport formats and their actual usage is presented (adapted from Tanner & Woodard 2004, pp. 127-129). Let us consider a situation where a mobile user is assigned two transport channels (DCHs) that are multiplexed onto a CCTrCH for transmission. This data stream is then transmitted on a physical channel - in this case on the DPDCH. The interface between higher layers and the physical layer is presented in Figure 11.

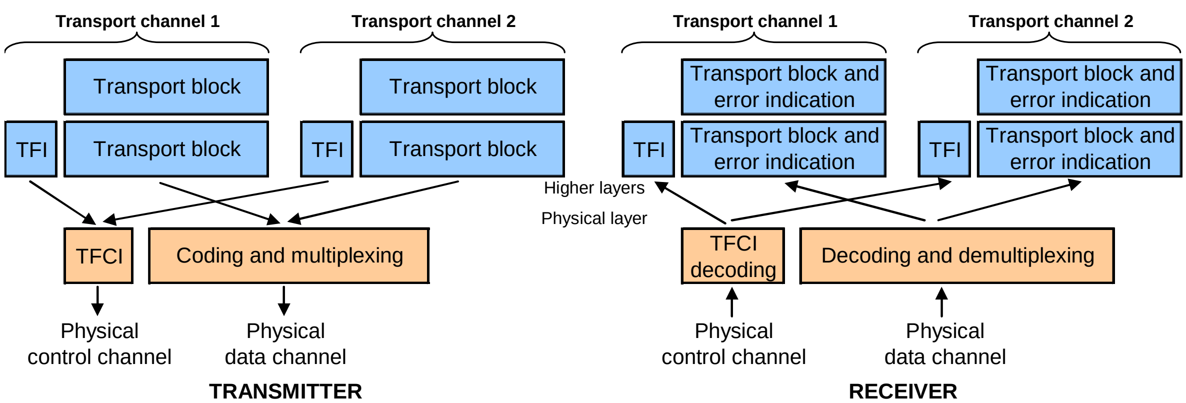

Figure 11 - Two transport channels mapped onto a single physical channel

A transport format set can be defined for each transport channel. This is done in Table 3.

Table 3 - Example transport format sets associated with two TrCHs

Transport format	Dynamic part (# bits, # TrBks)	Semi-static part
$TrCH_1$		TTI = 40 ms
$TF_{1,1}$	40,0	1/2 conv. coding
$TF_{1,2}$	40,1	12 bit CRC
$TF_{1,3}$	40,4	RM_1 = 100
$TrCH_2$		TTI = 10 ms
$TF_{2,1}$	0,1	1/3 conv. coding
$TF_{2,2}$	150,1	24 bit CRC
		RM_2 = 120

The first transport channel, denoted as $TrCH_1$, has three different TFs allowing three different data rates (off, low rate, high rate). All transport formats carry 40 bits in each transport block, but the number of transport blocks per transport block set varies. $TF_{1,1}$ carries no transport blocks at all and no data is actually transmitted when this TF is selected. $TF_{1,2}$ and $TF_{1,3}$ carry 40 and 160 bits per transport block set, respectively. The semi-static part of the transport format set is the same for all three transport formats.

The second transport channel, denoted as $TrCH_2$, has only two different transport formats allowing two different data rates (off, high rate). $TF_{2,1}$ sends one transport block every TTI but since the size of this block is zero, no data bits are actually transmitted. $TF_{2,2}$ carries one transport block of size 150 bits and the semi-static part is again the same for both of these transport formats.

The difference between sending no transport block at all (as with $TF_{1,1}$) and sending an empty transport block (as with $TF_{2,1}$) is that with the first one no allocation on the DCH is actually made but with the second one a transport block of zero bits is transmitted, a CRC is included in Layer 1 processing and a physical channel is reserved for the transmission.

We can now calculate that the maximum bit rate for $TrCH_1$ is 160 bits / 40 ms = 4 kbits/s and for $TrCH_2$ 150 bits / 10 ms = 15 kbits/s. In addition to different bit rates and TTI values, given the different channel coding types and CRC sizes, we would be much more likely to detect any transport blocks containing errors on $TrCH_2$ than on $TrCH_1$. Thus by defining the transport formats we also affect the Quality of Service (QoS) provided to the connection. Also the choice of the rate matching attribute gives a means of controlling the different protection given to different transport channels. (Tanner & Woodard 2007, p. 129)

Finally, Table 4 on the next page presents the transport format combination set (TFCS) for this example CCTrCH. Only these combinations of transport formats can be transmitted on the CCTrCH. There are a total of six possible unique combinations of transport formats to be applied, but not all of these are necessarily included in the TFCS. In this example, we have restricted the transmission so that both transport channels can't use their maximum bit rates simultaneously.

The RNC controls the use of TFCS in the network and it can thereby set its own limitations to the transmission. This can be achieved by limiting the allowed TFCs to those values for which the included TFs do not correspond to high bit rates simultaneously. This topic will be further investigated in the next chapter.

Table 4 - Example transport format combination set associated with a CCTrCH

Transport format combination (TFC)	TFC indicator (TFCI)	TF for $TrCH_1$	TF for $TrCH_2$
TFC_1	0	$TF_{1,1}$	$TF_{2,1}$
TFC_2	1	$TF_{1,2}$	$TF_{2,1}$
TFC_3	2	$TF_{1,3}$	$TF_{2,1}$
TFC_4	3	$TF_{1,1}$	$TF_{2,2}$
TFC_5	4	$TF_{1,2}$	$TF_{2,2}$

The means for calculating the TFCI values is presented later in chapter 5.3.5. To give an illustration of how the TFCI information is actually transferred on the physical layer and in a physical channel, see the different frame structures presented in Appendix 2. The different physical channels have somewhat similar structure as regards to transport format related information exchange.

This chapter has introduced the interface between the physical layer and the MAC and the RRC layer. The data transmission is organised by transport blocks and all necessary transmission attributes are indicated by the use of transport formats, transport format sets and transport format combination sets. As the theory part of this thesis is now completely covered and all key concepts related to transport formats are presented, it is time to move on to the next chapter and introduce the actual implementation of TFS and TFCS calculation in the Radio Network Controller (RNC) software.

5 Transport format implementation in the RNC

5.1 RNC architecture

This fifth chapter will focus on a transport format related software implementation in a real RNC solution. Before diving into the realisation of the program block in question, a wide overview of the RNC architecture is presented. The logical role of the RNC is already covered in chapter 2.3.2, so this chapter will focus only on the RNC's internal structure.

The RNC is a fairly complicated network element that consists of several different hardware and software layers. The observed RNC logically consists of four parts: network interface functions, switching and multiplexing functions, user plane functions and control functions. All functionalities are distributed to a set of functional units that are entities of hardware and/or software, capable of accomplishing a special purpose or function.

The whole system can be seen as a network of responsibilities and services. In this model, all software can be divided into three levels: system blocks, service blocks and program blocks. System blocks are hierarchically on the top of the model and they offer a defined number of services and perform a defined number of functions. Service blocks are in the middle of the model and they are part of the implementation of a system block. They offer certain system block services to other system blocks and a number of services used within the same system block. Program blocks are in the lowest level of the model and they are the actual implementations of service blocks.

This kind of software model is used widely in various kinds of software development projects. In this model, the different levels can be used to restrict the visibility of services – a term called *information hiding* is often used in this context. Services can be hidden so that program blocks belonging to a different service do not see each other's service definitions.

The RNC application software is included into one system block in the RNC, and all radio resource management related functionalities are located in a single service block. In this thesis, we are focusing only on one specific program block in the RRM area, and this program block is introduced in the next chapter.

5.2 Program block under investigation

A program block is a loadable package unit that, in runtime, corresponds to a process family. It is part of the implementation of a service block which offers a number of services to the outside and a number of services to be used within the service block. In this chapter, one program block in the radio resource management area of the RNC software is taken under more specific analysis. After introducing the program block shortly in the following, the transport format related functionality is covered in the next few chapters.

The observed program block is a completely new program block in the RNC software and it is a result of a major architectural change made recently in the RNC Layer 3 software development area. These changes are meant to improve Layer 3 signalling inside the RNC, software quality and especially software maintainability. This particular program block is meant to hide communication between certain other program blocks and to enable RRM related software to make radio resources related decisions independently in only one service block. The architecture change also benefits the module testing of the software.

The program block has interfaces to other program blocks as illustrated in Figure 12. The other program blocks #2 and #3 are responsible for basic RRM related features such as base station resource management and handovers. The other program block #1 is responsible for Layer 3 and other signalling towards the base stations and core network. The other program block #4 is used for some internal functions and resource allocation.

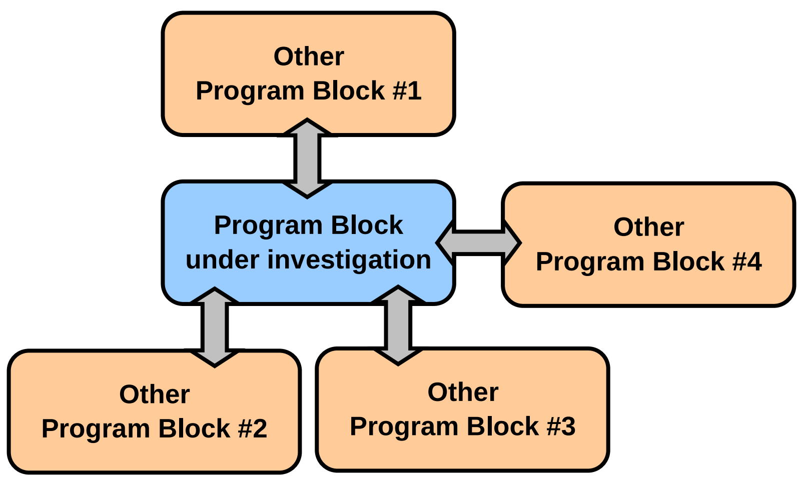

Figure 12 - Interfaces of the program block under investigation

In addition to the interfaces presented in Figure 12, the program block also has connections to RNC's internal databases e.g. for retrieving some global parameters. As this program block resides in the middle of these other program blocks, it is responsible for all kinds of radio resource management related task for one RRC-connection between the mobile and the UTRAN. The program block exists only when the mobile is in Cell_DCH state, i.e. when the UE is receiving and transmitting data on the dedicated traffic channel (DCH), see chapters 3.4 and 3.7 for a quick recap of these topics.

The program block consists of a number of code modules. The Free On-line Dictionary of Computing[4] defines a code module "as an independent piece of software that forms a part of one (or possibly more) larger program entity". Generally a module is the smallest code entity that can be compiled, deployed and run separately of other code entities implementing the software product. Modules provide an abstraction or an information hiding mechanism so that the internal implementation of a module can be changed without requiring any change to other modules.

One of the many tasks this program block performs is the TFS and TFCS calculation for an active connection. The next chapter will focus on this topic.

[4] See http://foldoc.org/

5.3 Current TFS and TFCS implementation

5.3.1 Scenarios for TFS and TFCS calculation

The program block performs TFS and TFCS calculation in the following situations:

- RRC connection setup: TFS and TFCS calculation for the signalling link (SL)
- State transition from Cell_FACH to Cell_DCH
- Inter-RNC hard handover (target RNC side)
- RAB establishment / PS NRT user plane creation (from 0/0 to X/Y bit rate)
- RAB modification (upgrade or downgrade)
- RAB release / PS NRT user plane release (only new TFCS calculation)
- Changing DL code (as a result from code tree re-allocation etc.)
- TFCS subset limitation
- Compressed mode activation and stopping (depending on CM method)
- SRNC relocation (in target side source side values must be used)

The calculation is performed using the same procedures in the code in all of these scenarios – only calling of these procedures and some minor details may vary depending on the scenario. These procedures are located in two distinct code modules that are presented in chapter 5.3.3.

The program block stores the current as well as the old TFS/TFCS values during the time period when these values are re-calculated. The reason for this is that if something goes wrong during the procedure, the old values can be restored easily.

TFS and TFCS calculation is never done in the DRNC but always in the SRNC side counterpart of the program block.

5.3.2 Data types

The program block stores the current DCH-list of the active connection and the TFS/TFCS calculation is largely based on this list. Information about one specific DCH is stored in data type dch_t, which includes (among others) the DCH-type, maximum bit rates for downlink and uplink and the id-number for that DCH.

The connection-specific DCH-list is stored in a separate DCH-container (data type dch_container_t), which simply consists of individual dch_t data types. The combined structure of these two data types is therefore as follows:

- dch_container_t
    - dch_t (max 8)
        - dch_id (byte)
        - dch_type (byte)
        - maximum_br_dl (dword)
        - maximum_br_ul (dword)

The DCH-type is always one of the following:

- Signalling link (always present in Cell_DCH state)
- AMR voice (consisting of three different subflows, each of different DCH-type)
- CS-T (circuit-switched transparent data channel, e.g. video telephony)
- CS-NT (circuit-switched non-transparent data channel, e.g. IP-traffic)
- PS-RT (packet-switched real-time channel, e.g. streaming applications)
- PS-NRT (packet-switched non-real-time channel, e.g. general web-browsing)
- E-DCH / HS-DSCH (not relevant in TFS/TFCS calculation)

Other transport channel types (BCH, FACH, PCH and RACH, see chapter 3.4) are not visible to the program block, because it only exists in Cell_DCH state and is aware of only dedicated transport channels (i.e. DCH and E-DCH) and the HS-DSCH common transport channel.

The Adaptive Multi Rate (AMR) speech codec and the related voice channel is fundamentally different from other DCH-types because it consists of three different subflows that are all treated as separate DCHs. Also the TFS calculation is performed separately for these subflows. The AMR codec has eight different source rates from 4.75 kbps to 12.2 kbps and a low rate background noise encoding mode (3GPP TS 26.071 2007, p. 7). In addition, a so-called wideband AMR (AMR-WB) has been defined by 3GPP to enhance voice quality with nine additional source rates from 6.60 kbps to 23.85 kbps (3GPP TS 26.171 2007, p. 7). All these different source rates affect the TFS calculation by each having different transport format parameters.

The maximum amount of 8 simultaneous DCHs is based on the restriction that in addition to the signalling link, there can only be one active connection to the circuit-switched core network (CS data channel requiring only 1 DCH or AMR voice call consisting of 3 subflows, i.e. 3 DCHs), one real-time packet-switched connection and three non-real-time packet switched connections – forming together 8 simultaneous DCHs.

The data types used for TFS and TFCS information exchange are split into a DCH-specific tfs_info_t data type and a connection-specific tfcs_info_t data type. The tfs_info_t data type is based on the TFS related theory presented in chapter 4.2 and its subfields are structured as following:

- tfs_info (tfs_info_t)
    - changed_dl (boolean)
    - tfs_dl (tfs_t)
        - dynamic_tf_info_list (max 10)
            - nbr_of_transport_blocks (word)
            - rlc_size (word)
        - semi_static_tf_info
            - transmission_time_interval (byte)
            - channel_coding_type (byte)
            - coding rate (byte)
            - rate_matching_attribute (word)
            - crc_size (byte)
    - changed_ul (boolean)
    - tfs_ul (tfs_t)
        - dynamic_tf_info_list (max 10)
            - nbr_of_transport_blocks (word)
            - rlc_size (word)
        - semi_static_tf_info
            - transmission_time_interval (byte)
            - channel_coding_type (byte)
            - coding rate (byte)
            - rate_matching_attribute (word)
            - crc_size (byte)
    - is_ul_and_dl_same (boolean)

The tfcs_info_t data type is based on the TFCS related theory presented in chapter 4.3 and its subfields are structured as following:

- tfcs_info (tfcs_info_t)
    - dl_changed (boolean)
    - tfcs_dl (tfcs_t)
        - ctfc_size (byte)
        - tfcs (word, max 1024)
    - ul_changed (boolean)
    - tfcs_ul (tfcs_t)
        - ctfc_size (byte)
        - tfcs (word, max 1024)
    - is_ul_and_dl_same (boolean)

The is_ul_and_dl_same parameter in both of these data types is used to inform the network that parameters for only one direction can be sent to the UE, which will minimise the required signalling between the UE and the network.

5.3.3 General on TFS/TFCS calculation

The TFS and TFCS functionalities are realised in code modules 6 and 7, respectively. The calculation with interfaces to other program blocks is illustrated in Figure 13. Each active connection has its own DCH-container (data type dch_container_t), which consists of one or several (maximum 8) DCHs (data type dch_t) as explained in the previous chapter.

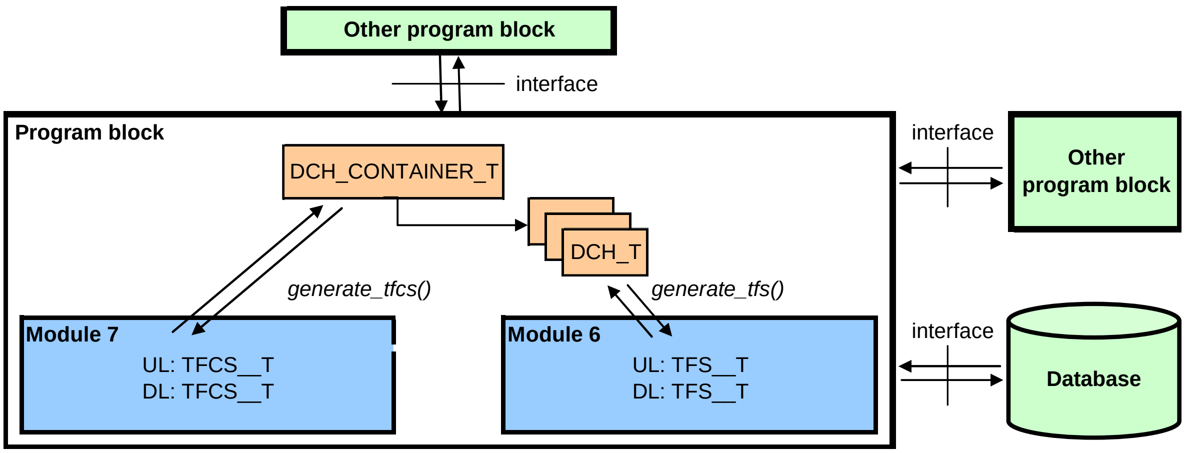

Figure 13 - Modules and interfaces related to TFS/TFCS calculation

When the program block needs to perform TFS and TFCS calculation, it first calls the generate_tfs() routine in Module 6 for each DCH separately. Transport format sets are calculated for both uplink and downlink, stored into tfs_t data type and combined into a DCH-specific tfs_info_t data type.

After the TFS calculation is completed, the program block calls the generate_tfcs() routine in Module 7. Transport format combination sets are calculated for both uplink and downlink, stored into tfcs_t data type and combined into a connection-specific tfcs_info_t data type. Finally all this information is passed to another program block, which is responsible for signalling the data towards the base stations and the UE.

In inter-system handover, the program block will not calculate TFS/TFCS at first. Instead, default TFS/TFCS values according to chosen AMR mode will be delivered to the UE. These default parameters are defined in 3GPP TS 25.331 (2007, pp. 1148-1215) and every UE must be able to handle them. Only after successful resource setup, the program block will calculate the TFS/TFCS based on the actual bitrates and other connection-specific parameters.

5.3.4 TFS calculation

The TFS calculation for individual DCHs is performed in Module 6, which contains a large number of internal procedures e.g. for different DCH types and filling the different parts of the TFS. Depending on the scenario, some other code module calls the generate_tfs() procedure in Module 6, which then proceeds with the calculation and calls its own internal procedures depending on DCH type, transmission direction et cetera.

In general, TFS parameters are derived in the program block as follows:

- Semi-static part
  - TTI: according to DCH type, bit rates and values read from a database
  - Channel-coding type: hard-coded values
  - Coding rate: hard-coded values
  - Rate matching attribute: received from another program block
  - CRC-size: hard-coded values
- Dynamic part (1-10 items)
  - RLC PDU size: according to bitrates and values read from a database
  - Number of transport blocks: derived from RLC PDU size and TTI

The hard-coded parts of the TFS semi-static part are pretty straightforward from the code implementation point of view. The most common value for channel-coding type and coding rate is one third rate convolutional coding, but depending on the UE capability and DCH-type also one third rate turbo encoding may be used. Formerly an option of no channel coding was also specified, but this has now been removed from 3GPP specifications. A brief overview of these encoding options is included in Tanner & Woodard (2004, p. 137), but they are not discussed in more detail in this context.

The CRC-size parameter of the TFS semi-static part depends on the DCH-type. The possible values are constants (0, 8, 12, 16 or 24 bits) and these values are simply hard-coded to the tfs_t data type presented earlier in chapter 5.3.2. Note that the program block does not actually perform any CRC calculation but only informs the size of this error detection code in the TFS.

As indicated earlier, neither the rate matching attribute of the TFS semi-static part is calculated by the program block itself but the value is received from another program block instead. The TFS module only sets this value to the tfs_t data type for the delivery towards the base stations and the UE.

The most interesting part of the TFS calculation is the dynamic part of the TFS and the TTI value in the semi-static part of the TFS. These parameters together define the maximum and possible intermediate bit rates for the transport channel. After all resources have been reserved successfully and maximum bit rates are stored in the dch_t data type, the dynamic part of the TFS can be calculated. Apart from the simple hard-coded filling of values for the signalling link, this calculation is separated into different procedures depending on the DCH type.

Generally, the TTI value is first read from a data base according to the DCH type and allocated maximum bit rate. For AMR calls and all three AMR subflows, the TTI value is always 20 ms as the speech encoder is based on speech frames of this size (3GPP TS 26.071, p. 7), but for other DCH-types the TTI value can be either 10, 20 or 40 ms. Usually the higher the bit rate, the lower the TTI value – an obvious result to keep the other parameters better under control.

As regards to CS-T and CS-NT type of DCHs, the TTI value is also affected by the Maximum SDU Size and Transfer Delay parameters. The first one of these indicates the maximum allowed Service Data Unit (SDU) size for the radio bearer and the second one the maximum delay for 95th percentile of the distribution of delay for all delivered SDUs during the lifetime of a RAB (3GPP TS 25.413 2007, p. 137). If the transfer delay is too critical, the TTI value can be decreased to meet the requirements, but in some cases the RAB allocation may fail because of too small a value of this parameter.

After the TTI value has been decided and approved, the rest of filling the TFS dynamic part is just laborious “if – elseif – elseif – … – else” -based decision making, which is affected by the bit rates and TTI value. This part of the code consists of hundreds and hundreds of rather similar looking lines. Every possible RLC PDU size and number of transport blocks -combination must be filled for that particular DCH and, depending on the DCH type and TTI, a different number of these dynamic part parameters is included in the TFS.

The most demanding DCH type for the calculation is the AMR speech call, which requires not only one but three DCHs, and TFS parameters need to be filled for all these subflows. This part of the implementation consists of several different loops and decision branches, where all supported AMR source rates are checked and their parameters are filled to the dynamic part of the TFS. Furthermore, the parameters are usually different in the downlink and uplink direction.

As mentioned earlier, TFS calculation is not performed for HS-DSCH (HSDPA) and E-DCH (HSUPA) transport channels. However, the HSDPA uplink return channel is handled as any other DCH, and therefore the TFS calculation is done for that as described above.

5.3.5 TFCS calculation

The TFCS calculation for the active connection is performed in Module 7. A prerequisite for the calculation is that transport format sets are first calculated successfully for each DCH separately. Another code module then calls the generate_tfcs() procedure in Module 7, which proceeds with the calculation and calls its own internal procedures. Generally the TFCS calculation is a bit more complex task than the TFS calculation, but the essential parts of the procedure are presented in the following.

The TFCS for one transmission direction consists of two parts as indicated in chapter 5.3.2:

- ctfc_size (byte)
- tfcs (word, max 1024)

The ctfc_size parameter informs the Calculated Transport Format Combination (CTFC) size in bits. The CTFC is a tool for efficient signalling of transport format combinations and the topic is discussed in full detail in 3GPP TS 25.331 (2007, pp. 1276-1277). The possible values for this parameter are 2, 4, 6, 8, 12, 16 and 24 bits.

The CTFC vector, named only as tfcs in the list above, is the actual representation of the combination of currently valid transport formats. The calculation of this vector is based on the number of transport channels and the size of their transport format sets. UE-capability information may also set limitations which need to be taken into account in the calculation. The program block stores this capability information and checks the limitations when any DCH is added or modified. These limitations may cause RAB related operation to fail.

The actual calculation of the CTFC vector involves some simple mathematics, which is adapted from 3GPP TS 25.331 (2007, p. 1276). Let I be the number of transport channels (in our case DCHs) and each transport channel $TrCH_i$, i=1,2,…,I, has L_i transport formats, i.e. the transport format indicator TFI_i can take L_i different values, $TFI_i \in \{0,1,2,\ldots L_i-1\}$.

Now we define a coefficient vector P, $P_i = \prod_{j=0}^{i-1} L_j$, where i=1,2,…,I and L_0=1.

Next, let the $TFC(TFI_1, TFI_2, \ldots, TFI_I)$ be the transport format combination for which $TrCH_1$ has transport format TFI_1, $TrCH_2$ has transport format TFI_2, and so on. The corresponding $CTFC(TFI_1, TFI_2, \ldots, TFI_I)$ value is computed as:

$$CTFC(TFI_1, TFI_2, \ldots, TFI_I) = \sum_{i=1}^{I} TFI_i \cdot P_i$$

The algorithm produces a unique CTFC value for all combinations of $TFI_1, TFI_2, \ldots, TFI_I$.

How this all is done in practice is that we form the coefficient vector P based on the currently allocated DCHs and their transport format set sizes. Before this, the DCHs are arranged based on the dch_id: $TrCH_1$ corresponds to the DCH having the lowest dch_id, $TrCH_2$ corresponds to the DCH having the next lowest dch_id, and so on.

Then we create a TFCS matrix, where each DCH represents a column, the possible TFIs of those DCHs are repeated on the rows, and thus each row in the matrix represents a unique transport format combination. The CTFC value can then be calculated by summing the rows together and weighting the columns with the coefficient vector P.

Finally, if the number of transport format combinations is larger than the allowed maximum, we need to call a procedure which reduces the TFCs in the TFCS list. Some of the intermediate bit rates are dropped out from the TFSs and the calculation is tried again. If the calculation succeeds, the values in the CTFC vector are arranged in an increasing order and returned for the delivery towards the UE.

The maximum index number of the CTFC vector determines the value of the ctfc_size parameter in the tfcs_t data type. The required amount of bits to represent the TFCI must be rounded up to the next possible value; 2, 4, 6, 8, 12, 16 or 24 bits as mentioned before.

A simple example will illustrate the calculation. Assume that three transport channels (i.e. DCHs) are allocated with $TFI_1 \in \{0,1,2\}$, $TFI_2 \in \{0,1,2\}$ and $TFI_3 \in \{0,1\}$. To reduce the combinations, assume furthermore that when $TFI_1=0$, any combination of TFI_2 and TFI_3 is allowed, while when $TFI_1 \neq 0$ then TFI_2 and TFI_3 must both be 0.

This will produce the following coefficient vector P:

$P_1 = L_0 = 1$
$P_2 = L_0 \cdot L_1 = 1 \cdot 3 = 3$
$P_3 = L_0 \cdot L_1 \cdot L_2 = 1 \cdot 3 \cdot 3 = 9$

The valid combinations for data transmission, and the computed CTFC values, are presented in Table 5.

Table 5 - CTFC calculation for the example case

TFI_1	TFI_2	TFI_3	CTFC	TFCI
0	0	0	$0{\cdot}1 + 0{\cdot}3 + 0{\cdot}9 = 0$	0
0	1	0	$0{\cdot}1 + 1{\cdot}3 + 0{\cdot}9 = 3$	1
0	2	0	$0{\cdot}1 + 2{\cdot}3 + 0{\cdot}9 = 6$	2
0	0	1	$0{\cdot}1 + 0{\cdot}3 + 1{\cdot}9 = 9$	3
0	1	1	$0{\cdot}1 + 1{\cdot}3 + 1{\cdot}9 = 12$	4
0	2	1	$0{\cdot}1 + 2{\cdot}3 + 1{\cdot}9 = 15$	5
1	0	0	$1{\cdot}1 + 0{\cdot}3 + 0{\cdot}9 = 1$	6
2	0	0	$2{\cdot}1 + 0{\cdot}3 + 0{\cdot}9 = 2$	7

Now we have a unique CTFC value for each transport format combination. To indicate the allowed combinations, the sequence of CTFCs {0, 1, 2, 3, 6, 9, 15} is signalled to the Node B and further to the UE. In this example, signalling each CTFC requires 4 bits, and thus the ctfc_size parameter value is set to 4.

Another two examples, involving AMR speech calls and its peculiarities, are presented by Ghadialy (2004). The AMR codec with its three subflows induce the TFCS matrix to be slightly different and more complex because the selected AMR source rates force certain transport formats to be used on all three subflows.

In RRC connection setup, the TFCS calculation is always performed for the signalling link only. This RAB configuration is the most straightforward of all and the implementation is therefore hard-coded for efficiency purposes. The size of the signalling link TFS is two (data is either transmitted at full speed or not at all) and the size of the TFCS is also two, the CTFC vector being simply {0, 1}. The ctfc_size parameter gets value 2 bits.

5.4 Analysis of the current implementation

The original implementation of TFS and TFCS calculation in the observed RNC SW dates back to year 2002 and 2003. As a result of the software architecture change project, the current implementation is realised by porting the related procedures quite directly from the old architectural solution. Therefore the implementation contains e.g. some global data structures that are copied and mapped to different local variables for proper TFS and TFCS calculation. The implementation itself is fully working and even quite efficient, because the underlying theory comes directly from 3GPP specifications, but the problems are mainly in data visibility and other internal operations.

Because of the procedure porting and other known issues, the implementation is definitely not optimal at present. Also from the code maintenance point of view, the current implementation is rather vulnerable. Code commentary would certainly need some improvements as well. This topic was already discussed in the author's special assignment (Valtanen 2007) preceding this thesis work. The resolution in that paper was that a more profound analysis is needed and a complete re-design of the TFS/TFCS calculation may be inevitable.

Now that the current implementation was investigated in a more detailed level and the issues were analysed and discussed with a larger group of people, it was decided that the TFS module (Module 6) can be edited and improved more freely than the TFCS module (Module 7). The TFCS module is a rather complex structure, where any modifications to the code may result in serious consequences from the whole functionality point of view. There were also some new functionalities still being developed to the TFCS module, which would require the implementation to remain otherwise relatively stable for the time being.

As both functionalities are currently working as intended, any possible modifications to the code need to be well-founded, reasonable and verified with proper testing methods. No modifications to the external or internal interfaces of the program block were allowed to be implemented within the limits of this thesis. However, the improvements to the current implementation are presented in the next chapter.

5.5 Improvements to the implementation

Based on the author's special assignment (Valtanen 2007), it was first considered that the whole TFS and TFCS implementation should be re-designed and possibly re-implemented from scratch. When the issues were discussed in more detail and when this thesis work evolved, it became clear that testing the current implementation would still require a significant amount of time and the author would be the main responsible of this task.

Therefore, testing of TFS and TFCS, the whole contents of Modules 6 and 7, was raised as an essential part of this thesis work. This entity, including the presentation of the testing process and the applied testing methods, is presented in chapter 6.

It also became clear that a more profound change in the design of these features was not possible in the time frame of this thesis, but only small corrections, modifications to clear weaknesses, removal of obsolete code, and, of course, better commentary of the code could be carried out during this work.

When this thesis work was started, Module 6 (TFS implementation) consisted of about 3350 lines of code and Module 7 (TFCS implementation) of 4100 lines. The committed changes decreased the size of both modules a couple of hundreds of code lines. After careful consideration, some obsolete parts of the code were removed, some loops and algorithms were optimised, but, in the meanwhile, a large number of commentary lines were added to both modules.

There were also some small bug fixes to the implementation that came up during the writing of this thesis, but the author was mainly not responsible for these corrections. These changes are, however, discussed in chapter 6.6 where the committed changes to TFS/TFCS test cases are presented.

After all, the implementation still deploys some global data structures, clumsy data conversions and unnecessary gimmicks that could well be optimised in the future. However, the implementation is now fully functional, seemingly flawless at the time, and the improved commentary of the code certainly enhances the code maintainability.

5.6 Possible changes and further analysis

The RNC software observed in this thesis is under constant state of change. Totally new functionalities are introduced with regular intervals and also old features are being improved and optimised all the time. These changes usually affect numerous service blocks and program blocks underneath simultaneously, which adds the complexity and work required by each change.

The improvements made in the implementation were presented in the previous chapter. Depending on the progress of the whole architecture change project, a more profound change in the TFS/TFCS design and implementation may be realised after the completion of this thesis.

One possible target for a larger improvement process is the TFCS module. For example, in the current implementation the sizes of transport format sets are re-calculated based on DCH type, transmission direction, maximum bit rate, TTI value and possible restriction classes. It would certainly be easier to just copy these values straight from the TFS data structure, which is anyway updated just before the TFCS calculation.

One of the difficult features from the TFS and TFCS point of view is certainly the AMR speech coded with its different source rates. The original AMR codec consists of eight different source rates yielding up to eight different transport formats for all three AMR subflows as described in chapter 5.3.2. These source rate modes force certain transport formats to be used on all three subflows, which affect both TFS and TFCS calculation as presented earlier.

Now when we add the AMR-WB feature with its nine additional source rates, the TFS/TFCS calculation becomes quite laborious and the code implementation grows accordingly. There is no simple solution for handling these AMR scenarios, but they always require a special handling in the code.

If more AMR source rate codecs are introduced by 3GPP in the future, this affects the TFS and TFCS implementation in some level, but the implications can be minimised already now by taking the possible changes into account when the code is modified and by keeping the code well-maintainable. Also testing of any new feature obviously requires some effect.

Although HSPA, MBMS and other new features of WCDMA constantly bring more and more issues to the development process, the basic functionality of the underlying network must remain untouched. Especially all transport format parameters analysed in this thesis need to be kept unchanged regardless of any new functionality introduced in the network. The basic requirement is that all UEs based on Release ’99 (published now over eight years ago) must be supported by all WCDMA networks.

The radio resource management related functionality in the network is slowly moving towards the base stations since the constantly growing bit rates have tighter delay requirements and they require faster response times from the network. However, the software in the network is changing and developing as well. With modular implementation in the RNC and in the Node B the same software can be deployed in any network element if only the underlying functionality remains relatively unchanged. This reduces the work amount required for new services and technologies substantially.

The conclusion is that the implementation observed in this thesis can still be improved with some re-design of the procedures and also with other optimisations. At the same time, the code needs to be kept as easily maintainable as possible.

Also module and regression testing of these features must be well designed and correctly managed and conducted. The testing process, different testing methods and the test environment used for testing TFS and TFCS functionalities in the RNC are taken under more specific analysis in the next chapter.

6 Testing of transport formats in RNC software

6.1 Testing process

As described earlier in this document, the observed program block is only a small portion of RNC software. The whole product is implemented incrementally in projects and programs with a large number of new features realised in each phase. The product and all program blocks of it are also tested in separate phases. These different testing phases of RNC software are illustrated below in Figure 14.

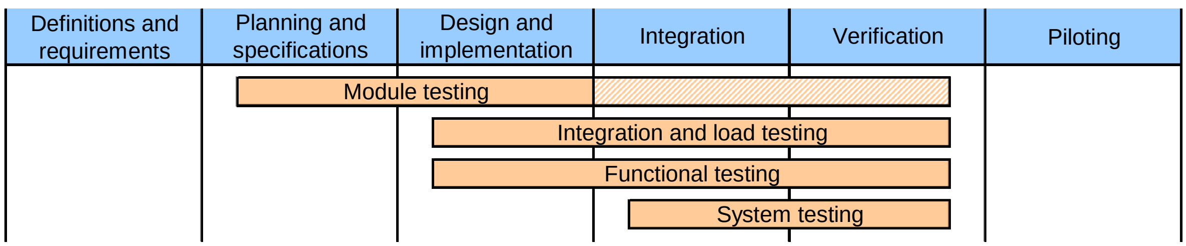

Figure 14 - Testing process phases

6.1.1 Module testing

Module testing (also called *unit testing* in some references) is the lowest level of testing right after a specific part of the code is implemented and this is the level we are focusing on in this thesis. Module testing can be seen as an integrated part of the design and implementation phase. It is planned and executed during and after the design and implementation, and new functionalities in the code should be verified as soon as possible. Ideally, module testing should be complete right after the implementation is ready, but usually this target is not achieved. Module testing continues after the implementation has been completed and so-called *regression testing* is executed when major corrections or changes are done in later phases of the development process (Pressman 2005, p. 401).

Module testing has some requirements, which have been discussed e.g. by Pressman (2005), Kaner (2003) and Virkki (2007). Good module testing should be at least thorough (good coverage, ability to test any internal routine or service), repeatable (independency from other program blocks or projects), well maintainable and readily automatic. The result of a module test case should be either “success” or “failure”, so that the test engineer does not need to check any print-outs with special knowledge to confirm the result.

In module testing level, there is usually a very close collaboration between the test engineer and the people working with the code. This is the case also with the observed program block and our working environment. The collaboration enables an immediate mutual information flow between the test engineer and the code developer. All design documents and code modules are available for the test engineer so that he or she can select proper inputs for testing and thus obtain better test coverage for test cases.

Module test cases are logically grouped into test case sets and documented to a module test plan. The module test plan describes what is being tested with which test cases and it also contains information about the test environment and targeted test coverage. This plan is inspected and reviewed by a large group of people working with the program block. The scope of module testing should not be too far from higher level testing so that a smooth transition from module testing to higher level testing is enabled.

There are also a number of other requirements and practices for good module testing, but these may be company internal or program-specific requirements which can’t be included in the scope of this thesis. Also general project management practices (e.g. Kujala & Martinsuo & Artto 2006; Vartiainen et al. 1999) can be applied in software development and module testing.

6.1.2 Integration and load testing

In module testing, there is usually only one program block or feature to be tested and other processes or program blocks are simulated or faked. After the code implementation has been verified in module level testing, modules and units are integrated to a larger entity for higher level testing, where each program block runs as a real process against each other.

This *integration testing* starts already in the specification phase, but the actual testing takes place in the unit integration phase. Now the focus is on the design and the construction of the software architecture. Integration testing usually reveals some specification problems, which cannot be found in module testing phase. This phase also contains *load testing*, where the system’s robustness is tested with intense traffic or features requiring high processing capacity. (Pressman 2005, p. 390)

6.1.3 Functional testing

After the integration testing phase, the complete product is tested in the *functional testing* level - sometimes also called *validation testing*. This phase verifies the correct behaviour of the whole product, in this case the RNC, and it is done with real network hardware in a real environment. The planning of functional testing is based on the system specifications and it can be started as soon as the requirements for the system have been approved. In functional testing, all software modules work together, their cooperation is ensured, all new functionalities are tested and also all old functionalities are guaranteed to be working correctly.

6.1.4 System testing

System testing is the final stage in the product software development process. In this phase, testing is done from not only the product manufacturer's point of view but also the customer's requirements and demands are taken care of. Simulators, such as traffic generators, can be used when necessary. The purpose of system testing is to ensure that the whole product works as it is meant to work in the circumstances and procedures that correspond to the intended use of the product. System testing ensures the development team that the product is ready for (pilot) delivery to the customer.

6.2 Testing methods

Before introducing the test environment used in the observed RNC SW development team, let us take a brief look at the general testing methods in the software development world. If we ignore some static techniques, such as walkthroughs, inspections and technical reviews (IEEE Std. 1028-1997, 1998; Freedman & Weinberg 1990) that are not based on the execution of the software product itself, there is a general division of different approaches between black box testing and white box testing.

Black box testing (sometimes also *behavioural testing*) is a specification method, which doesn't have any insight into the code itself. Black box testing is based on the requirements set for the system. It takes an external perspective to the test object to derive test cases and it can be done in total ignorance of how the object is constructed. This method is mainly based on inputs and outputs, so the test designer selects suitable inputs and determines and verifies the correct output. The system and its state cannot be observed during the testing and there is no knowledge of the test object's internal structure or implementation. (Meyers 1979; Beizer 1995, p. 8)

White box testing takes a different aspect. In white box testing, the testing strategies are derived from the structure of the tested object. Now the code architecture, branches and statements are all available for the test engineer. The system and its state can be observed during the test procedure and the test flow can be designed to meet system characteristics. This approach is generally more intense, more complicated and usually also more expensive. Synonyms for white box testing are *structural testing*, *clearbox testing* and *glass-box testing*, which quite well reflect the full visibility of the internal implementation of the software product. (Beizer 1995, p. 8)

Hybrid test strategies combine both black box and white box testing strategies. None of these approaches can be said to be superior to any other, and different testing strategies may be used in different testing phases. Hybrid strategies are often useful at all testing levels. (Beizer 1995, p. 9)

In the observed environment and with the observed program block, module level testing is mainly based on black box testing. The test environment is introduced later in chapter 6.3. However, black box testing is not all testing there should be done. According to Beizer (1995, p. 11), black box testing should only represent 35 to 65 percent of all testing – with object oriented programming a bit more, though.

Black box type of testing is quite well suitable for testing the TFS/TFCS features introduced in this document, but certain properties of white box testing are certainly of value. The knowledge of the program block's internal state helps in analysing the results and finding any possible bugs in the code. Therefore the implementation includes some switches and enhancements with which we can e.g. check the internal DCH and radio link container of the active connection by sending a test message to the program block. The results can then be checked from the internal log writings and we can actually in some level observe the internal state of the program block. These test messages and internal log writings do not have effect on the final result of the test case, but they can be applied in numerous ways to improve module testing and they are especially handy when undesired bugs need to be located in the code.

6.3 Testing environment

There are several different kinds of module testing systems used in the observed product line and as the execution of test cases is mostly automated, the analysis of them is not. Test cases used for testing the TFS/TFCS functionality were implemented using a widely exploited testing environment illustrated in Figure 15. The environment consists of a Windows workstation (user interface and test case design/execution) connected to an execution environment (RNC software emulator running in a standard PC) and a definition database (network directory). The interconnection of these components is handled simply by a Local Area Network (LAN).

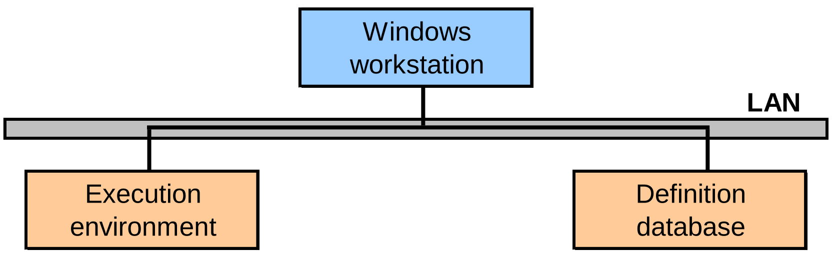

Figure 15 – The testing environment framework

Individual test cases consist of one Module Under Test (MUT) and several simulated counterpart processes. The message exchange between the MUT master process, MUT hand process and related counterpart processes is described using a message sequence chart (MSC) illustrated in Figure 16.

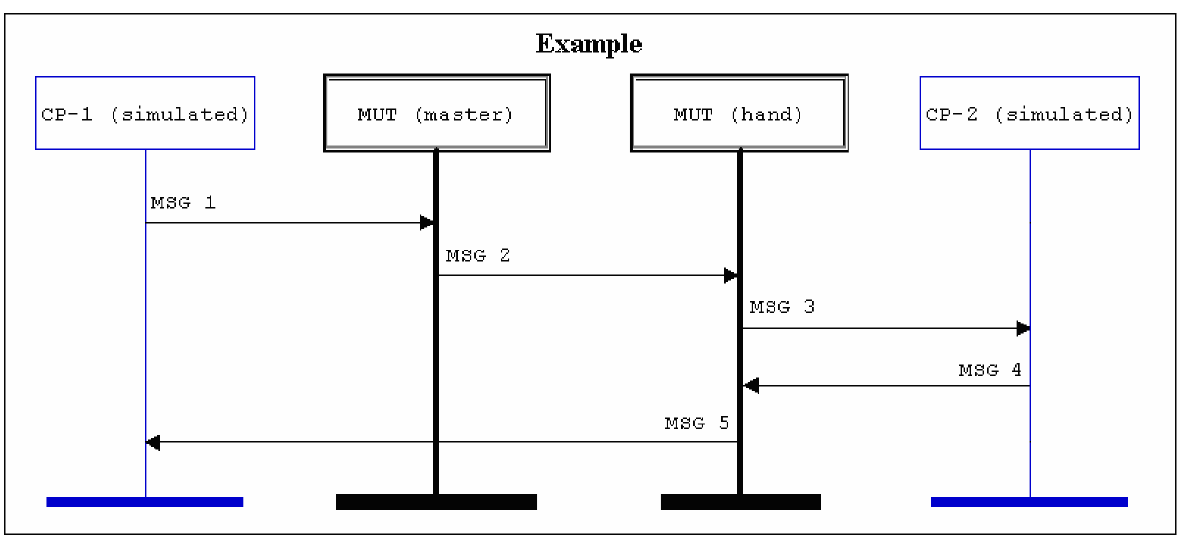

Figure 16 - The framework of a test case

All messages include a lot of parameters that can be controlled by the test engineer. The test engineer can freely determine what input values are included in the messages that are sent to the program block, and he or she also has responsibility for what output parameters sent by the program block are verified. Logically related test cases and common configuration data are collected to a module test project and the state of each test case (not tested / ok / not ok) can be monitored separately.

Monitoring individual messages in a test case is possible in the Windows-based testing program, either in a sophisticated graphical view or in a brute, but often surprisingly useful, hexadecimal view. These monitoring alternatives are illustrated in Figure 17 with a message carrying TFS information as described earlier in chapter 5.3.2.

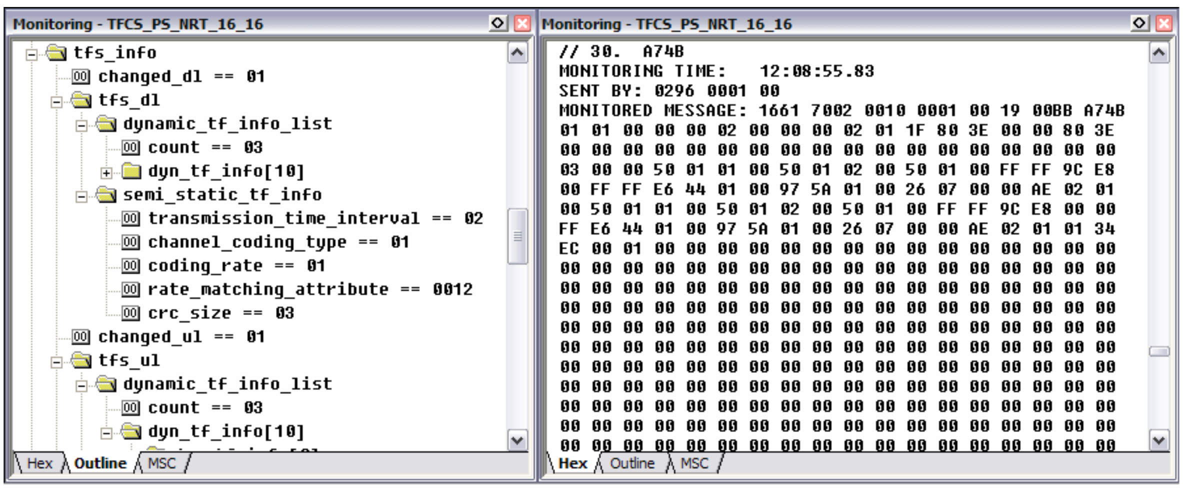

Figure 17 - Graphical (on the left) and hexadecimal (on the right) view of the message data

Executing any test case successfully requires both the message sequence to be exactly as defined in the test case MSC and the message data in each message to be as required in the test case. Not all data fields in the message are necessarily filled or monitored, but usually only the essential parts for that particular test case are of interest. This guarantees that unnecessary dependencies between different test cases are avoided and the testing process can be intensified substantially.

6.4 Initial test cases

For the purposes of this thesis, a TFS/TFCS related test case set was designed. These features were already implemented but not yet tested in module level testing when this thesis work was started. The test case set initially consisted of 47 individual test cases, where in each test case a different scenario was separately configured and tested. The test cases together with their TFS/TFCS properties are listed in Table 6.

Table 6 - Initial test cases for testing the TFS/TFCS feature

Test case	Scenario	DCH configuration (bit rates DL/UL)	TFS size (DL/UL)	TFCS size (DL/UL)
1	RRC connection setup	SL (3.7/3.7)	2/2	2/2
2	RAB setup	SL + CS-T data (28.8/28.8)	2/2 + 2/2	4/4
3	RAB setup	SL + CS-T data (32.0/32.0)	2/2 + 2/2	4/4
4	RAB setup	SL + CS-T data (33.6/33.6)	2/2 + 2/2	4/4
5	RAB setup	SL + CS-T data (64.0/64.0)	2/2 + 2/2	4/4
6	RAB setup	SL + CS-NT data (14.4/14.4)	2/2 + 2/2	4/4
7	RAB setup	SL + CS-NT data (14.4/28.8)	2/2 + 2/3	4/6
8	RAB setup	SL + CS-NT data (14.4/57.6)	2/2 + 2/5	4/10
9	RAB setup	SL + CS-NT data (28.8/14.4)	2/2 + 3/2	6/4
10	RAB setup	SL + CS-NT data (28.8/28.8)	2/2 + 3/3	6/6
11	RAB setup	SL + CS-NT data (28.8/57.6)	2/2 + 3/5	6/10
12	RAB setup	SL + CS-NT data (57.6/14.4)	2/2 + 5/2	10/4
13	RAB setup	SL + CS-NT data (57.6/28.8)	2/2 + 5/3	10/6
14	RAB setup	SL + CS-NT data (57.6/57.6, TD=200 ms)	2/2 + 5/5	10/10
15	RAB setup	SL + CS-NT data (57.6/57.6, TD=150 ms)	2/2 + 3/3	6/6
16	RAB setup	SL + CS-NT data (57.6/57.6, TD=90 ms)	2/2 + 2/2	4/4
17	RAB setup	SL + PS-RT (8/8)	2/2 + 2/2	4/4
18	RAB setup	SL + PS-RT (16/16)	2/2 + 3/2	6/4
19	RAB setup	SL + PS-RT (32/32)	2/2 + 5/3	10/6
20	RAB setup	SL + PS-RT (64/64)	2/2 + 5/5	10/10
21	RAB setup	SL + PS-RT (128/128)	2/2 + 5/5	10/10
22	RAB setup	SL + PS-RT (256/128)	2/2 + 5/5	10/10
23	RAB setup	SL + NB-AMR (4.75/4.75)	2/2 + 4/4	8/8
24	RAB setup	SL + NB-AMR (5.90/5.90)	2/2 + 4/4	8/8
25	RAB setup	SL + NB-AMR (7.95/7.95)	2/2 + 6/6	12/12
26	RAB setup	SL + NB-AMR (12.20/12.20)	2/2 + 3/3	6/6
27	RAB setup	SL + WB-AMR (6.60/6.60)	2/2 + 5/5	10/10
28	RAB setup	SL + WB-AMR (8.85/8.85)	2/2 + 5/5	10/10
29	RAB setup	SL + WB-AMR (12.65/12.65)	2/2 + 5/5	10/10
30	RAB setup	SL + PS-NRT (0/0)	2/2 + 1/1	2/2
31	RAB setup	SL + PS-NRT (8/8)	2/2 + 2/2	4/4
32	RAB setup	SL + PS-NRT (16/16)	2/2 + 3/3	6/6
33	RAB setup	SL + PS-NRT (32/32)	2/2 + 3/3	6/6
34	RAB setup	SL + PS-NRT (64/64)	2/2 + 5/5	10/10
35	RAB setup	SL + PS-NRT (128/128)	2/2 + 5/4	10/8
36	RAB setup	SL + PS-NRT (256/256)	2/2 + 5/5	10/10
37	RAB setup	SL + PS-NRT (384/384)	2/2 + 6/6	12/12
38	Multi-RAB setup	SL + NB-AMR (12.20/12.20) + PS-NRT (128/128) + PS-RT (64/64)	2/2 + 3/3 + 5/4 + 5/5	150/120
39	Multi-RAB setup	SL + PS-RT (32/32) + CS-T (32.0/32.0) + PS-NRT (64/64)	2/2 + 5/3 + 2/2 + 5/5	100/60
40	Multi-RAB setup	SL + WB-AMR (12.65/12.65) + PS-RT (32/32) + PS-NRT (64/64)	2/2 + 5/5 + 5/3 + 5/5	250/150
41	Multi-RAB setup	SL + CS-NT (14.4/14.4) + PS-RT (16/16) + PS-NRT (128/128) + PS-NRT (128/128)	2/2 + 2/2 + 3/2 + 5/4 + 5/4	300/128
42	RAB upgrade	SL + PS-NRT (128/128 → 128/256)	2/2 + 5/4 → 2/2 + 5/5	10/8 → 10/10
43	RAB downgrade	SL + PS-NRT (128/128 → 128/64)	2/2 + 5/4 → 2/2 + 5/5	10/8 → 10/10
44	RAB release	SL + PS-NRT → SL	2/2 + 5/4 → 2/2	10/8 → 2/2
45	SRNC relocation	SL	2/2	2/2
46	SRNC relocation	SL + NB-AMR	2/2 + 3/3	6/6
47	Inter-RNC HHO	SL + NB-AMR	2/2 + 3/3	6/6

As can be seen from the table, the focus in the test case set is clearly on different kind of RAB setup scenarios (cases 2-41). The TFS to be formed is different for each DCH type and it also depends on the defined maximum bit rate as described earlier in this document. The other scenarios (cases 42-47) use the same procedures in Module 6 as all RAB setup scenarios and these are designed only for checking that the TFS procedure calls are working correctly. The same applies to TFCS calculation in Module 7. Therefore there is no need to define separate test cases e.g. for all DCH combinations possible in SRNC relocation. From the code coverage point of view this kind of a test case set covers all possible code branches fairly extensively.

The scenario for all RAB related test cases follows the same pattern: after RRC connection setup (including the signalling link), a RAB is created with one radio link and the message going from the program block under investigation to another program block responsible for signalling the data forwards is verified for all TFS and TFCS data. The procedures for calculating this data are described earlier in chapters 5.3.4 and 5.3.5, respectively. Note that in addition to the RAB to be configured, there is always the signalling link (SL) present in the DCH list in each scenario.

What is also worth noticing is that with some DCH allocations, e.g. PS-NRT 128/128 kbps, the size of the TFS can be different on downlink and uplink even though the bit rate is the same. This can be explained by different TTI values and thus different transport block set sizes on the different transmission directions. For this particular PS-NRT 128/128 kbps case, on the downlink the TTI is 20 ms with possible transport block set sizes being 0, 1, 2, 4 and 8. On the uplink the TTI is only 10 ms with transport block set sizes 0, 1, 2 and 4. As the transport block size is 320+16 bits (user data + RLC overhead) in both cases, the maximum user bit rate yields up to 128 kbps.

For CS-T cases (e.g. video conference calls) these RABs are always symmetrical from the maximum bit rate point of view. For all other DCH types the bit rates can be asymmetrical, but it was agreed that there is no need to implement test cases for all different bit rate combinations. There are, however, different procedures for calculating these TFSs on the uplink and downlink, and the current test cases are therefore sufficient.

These test cases were prepared and executed successfully before the improvements described in chapter 5.5 were designed or implemented. At this phase, there were still some bugs and misconceptions found in the implementation, but these were corrected by other people responsible for the code at the time. After this, it was confirmed in higher level testing that all TFS and TFCS related functionalities of the program block were working properly without any apparent errors.

Throughout the improvement process it was considered necessary that all implemented test cases should be able to be executed successfully whenever the code would be committed to the common version handling system. By doing so, it could be guaranteed that the code image to be distributed from the module testing team to higher level testing was functioning without any problems in this area.

Should there be any new test cases that were considered necessary for TFS/TFCS testing, these new test cases were agreed to be implemented during the writing of this thesis and possibly even after its completion. These issues are discussed in chapter 6.6.

6.5 Analysis of the testing methods used

Virkki (2007) has considered in his thesis the risk handling in module testing environment. The results show that risks can be reduced to some extent by certain activities, such as proper inspections, co-operation with later testing phases and using coverage measurement tools revealing code branches that are left untested.

With the feature examined and improved in this thesis, inspections with other coders and designers were arranged at regular intervals. The module test cases were inspected before the improvements to the implementation were started and any new test cases produced during the writing of this thesis were agreed to be approved by at least some of the experts in the area.

Co-operation with later testing phases was realised by the author being the contact person in this area for later phase testing teams. No special test coverage tools were used since isolating this particular feature of the program block for this purpose was not seen worth realising or in any way necessary. Coverage tools are, however, utilised in the program-block-wide module testing of the program block.

The problem of the developer and the tester being the same person was not considered a bad situation in this particular case. It can be seen that as a tester the developer knows the code perfectly and can therefore make efficient test cases with high coverage of the feature. In his thesis, Virkki (2007, p. 82) suggested that only one developer and tester may prove suitable in some smaller implementations – just like the one now in question.

According to author's own experience and also Virkki's observations (2007, p. 83), the most important thing that reduces the risk of insufficient quality in module testing is the competence of the testing engineer him/herself. When the tester knows his or her working area well enough, he or she can design good test cases regardless of the feature or the environment.

6.6 Improvements and changes to test cases and testing methods

Kaner (2003) has discussed in his research what really is a good test case. Designing good test cases is a complex art, but the initial test cases prepared in the starting phase of this thesis were considered quite efficient. From the code coverage point of view they were covering all branches fairly extensively and even some new bugs were found when this test case set was designed – although the implementation was already considered quite functional and flawless at that time.

Nevertheless, as the work with this thesis evolved, a need for some changes in the designed test case set was discovered. First of all, there was a small misconception regarding PS-RT calls and hence the new software architecture was not giving the same TFS values as the old architecture did. Even though the values were looking alright from one point of view, a small correction to the TFS calculation algorithm was needed for the whole system to work properly. This correction affected almost all PS-RT and multi-RAB cases as the transport block sizes were doubled and the transport block set sizes were halved – the user bit rates, however, remaining the same.

When the initial test case set was designed, it was intentionally left as an open issue whether new test cases regarding compressed mode and its TFS/TFCS functionality were needed. Compressed mode, further discussed e.g. by Tanner & Woodard (2004, pp. 112-121), is a radio path feature which is needed when the UE makes measurements in another frequency or Radio Access Technology (RAT, basically GSM) in a WCDMA system. It consists of stopping the transmission during a certain amount of time to create gaps in the DCH transmission to allow the UE to make these measurements. Depending on the selected compressed mode method, this may have effect on the TFSs and the TFCS of the connection.

As a result of this theory and after some discussion with the code developers, it was decided that a few new test cases were needed. These three new test cases are testing compressed mode start and stop scenarios with some TFS changes and TFCS recalculation taking place in the connection.

One additional test case was designed for inter-RNC hard handover scenario. This test case uses a different RAB combination than the previously designed inter-RNC HHO scenario and thus the TFS and TFCS are also different. There was a doubt that this particular SL + PS-NRT + CS-T combination with certain bit rates and other parameter values would result into false TFS values, but eventually this could not be verified at least in module testing level.

Furthermore, two additional test cases were prepared to ensure that the TFCS calculation does not get confused when the dch_id numbers for AMR subflows are not consecutive – although they usually are. This scenario can happen, for example, when PS-NRT DCHs are first configured on dch_ids 1 and 2, the first one of these gets removed and AMR is then allocated on dch_ids 1, 3 and 4. However, AMR subflows A, B and C are always allocated on dch_ids in an increasing order.

All in all, the current test case set for testing the TFS/TFCS functionality in the program block includes 53 test cases, which are now intended to be run as a part of normal regression testing of the program block. The PS-RT related changes to the initial test cases (cases 17-22 and 38-41, compare to Table 6) with the 6 new test cases (cases 48-53) for inter-RNC HHO, compressed mode and AMR multi-RAB, are presented in Table 7.

Table 7 - Changes to initial test cases and new test cases

Test case	Scenario	DCH configuration (bit rates DL/UL)	TFS size (DL/UL)	TFCS size (DL/UL)
17	RAB setup	SL + PS-RT (8/8)	2/2 + 2/2	4/4
18	RAB setup	SL + PS-RT (16/16)	2/2 + 2/2	4/4
19	RAB setup	SL + PS-RT (32/32)	2/2 + 3/3	6/6
20	RAB setup	SL + PS-RT (64/64)	2/2 + 5/5	10/10
21	RAB setup	SL + PS-RT (128/128)	2/2 + 5/5	10/10
22	RAB setup	SL + PS-RT (256/128)	2/2 + 5/5	10/10
38	Multi-RAB setup	SL + NB-AMR (12.20/12.20) + PS-NRT (128/128) + PS-RT (64/64)	2/2 + 3/3 + 5/4 + 5/5	150/120
39	Multi-RAB setup	SL + PS-RT (32/32) + CS-T (32.0/32.0) + PS-NRT (64/64)	2/2 + 3/3 + 2/2 + 5/5	60/60
40	Multi-RAB setup	SL + WB-AMR (12.65/12.65) + PS-RT (32/32) + PS-NRT (64/64)	2/2 + 5/5 + 3/3 + 5/5	150/150
41	Multi-RAB setup	SL + CS-NT (14.4/14.4) + PS-RT (16/16) + PS-NRT (128/128) + PS-NRT (128/128)	2/2 + 2/2 + 2/2 + 5/4 + 5/4	200/128
48	Inter-RNC HHO	SL + PS-NRT (8/8) + CS-T (64/64)	2/2 + 2/2 + 2/2	8/8
49	Compressed mode	SL + PS-NRT (64/32 → 32/16)	2/2 +5/3 → 2/2 + 5/3	10/6 → 10/6
50	Compressed mode	SL + PS-NRT (32/32 → 8/8)	2/2 + 3/3 → 2/2 + 5/5	6/6 → 10/10
51	Compressed mode	SL + PS-NRT (32/32 → 8/8) + PS-NRT (32/32 → 8/8)	2/2 + 3/3 + 3/3 → 2/2 + 5/5 + 5/5	18/18 → 50/50
52	Multi-RAB setup	SL + PS-NRT (32/32) + NB-AMR (12.20/12.20)	2/2 + 3/3 + 3/3	18/18
53	Multi-RAB setup	SL + PS-NRT (32/32) + PS-NRT (32/32) + NB-AMR (12.20/12.20)	2/2 + 3/3 + 3/3 + 3/3	54/54

6.7 Results of module testing

In the initial phase of this thesis, the test case set presented in chapter 6.4 was executed successfully with all 47 test cases showing a yellow happy face in the test program. Based on these results, it was considered that all TFS and TFCS related features were working acceptably and well enough without any visible flaws or errors.

During the writing of this thesis, one problem described in the previous chapter was discovered regarding PS-RT calls and their TFS calculation. The problem was that the new software architecture, although working correctly from its own point of view, was not working similar way as the old architecture. The analogous functionality had been one of the project requirements throughout the software architecture change process, so the problem had to be corrected. However, this problem could not have been detected in module level testing as the program block was still giving reasonable values in its TFS calculation.

After the PS-RT related corrections, inter-RNC HHO, compressed mode and AMR multi-RAB additions described in chapter 6.6 were completed, the test case set consisted of 53 individual test cases. These new cases were successfully passed right after they were designed and the corresponding changes in the code were implemented. Also after the code improvements described in chapter 5.5 were committed, all test cases were executed successfully.

After all, we can conclude that all TFS and TFCS features of the program block are now tested fairly extensively and thoroughly. These features are also proven to be fully functional in higher level integration testing.

Any possible modifications made into the implementation in the future are now easy to verify in module level testing, thanks to the extensive and well-maintainable test case set designed during this thesis work. The work done for this thesis will definitely pay itself back at some point during the continuous development phase of the observed program block.

7 Discussion and conclusion

This thesis has introduced transport formats as an essential part of 3G WCDMA networks. Transport formats are applied in the data exchange between the physical and data link layer and they define the characteristics of a transport channel. Transport formats control the instantaneous data rate of a transport channel and affect on the physical layer coding applied to the channel. The use of transport formats enable some of the main characteristics of 3rd generation mobile communication systems: flexible delivery of any type of service, variable bit rates to end-user and multiplexing of services on a single connection.

Even though the transport format related theory presented in chapter 4 is rather directly based on 3GPP specifications, manufacturer specific implementations have a significant role in the efficient usage of transport channels and radio resources in general.

The Radio Network Controller, the RNC, performs a large majority of radio resource management related tasks in a WCDMA network. In chapter 5, this thesis presented one of the program blocks participating in radio resource management and transport format related calculation in the RNC. The program code was carefully inspected and the essential parts of the implementation were presented in chapter 5.3.

In the introduction chapter of this thesis, some improvements to the program code were planned to be designed and implemented. The current implementation was analysed in chapter 5.4 and the improvements to the implementation were presented in chapter 5.5. Although these improvements were eventually smaller than originally intended, they have certainly enhanced the code maintainability and corrected some minor bugs in the present implementation. The code efficiency has also been enhanced in some level. Moreover, the author can now be considered to be an expert in this area, which will hopefully benefit the program in the future.

The analysis of the implementation continued in chapter 5.6. The whole RNC software is under constant state of change, which requires the code to be as well maintainable as possible. New functionalities are introduced constantly and old functionalities are improved all the time. Even though the role of Node Bs in radio resource management is significantly increasing, the RNC still performs a large majority of these tasks in the network. Moving RRM related functionalities to a totally different network element requires big changes in the software, but these changes can be minimised by modular implementation and proper code maintenance.

Module testing of TFS and TFCS functionalities was also carried out during the writing of this thesis. This was the main topic in chapter 6. Some minor bugs were found and corrected in the code, and the designed test case set was improved for better code coverage. The analysis of the module testing process has shown that not all bugs can be found in module testing level but later testing phases usually reveal some defects as well. However, the earlier the bugs are found, the cheaper and easier the correction is. Overall, the features covered in this thesis are now fully tested and the designed test case set will remain as an essential part of regression testing in the future.

This thesis has also analysed module testing methods that are used in the observed environment. The designed test case set was found to be rather efficient from the code coverage point of view even though no actual code coverage tools were used. Black box type of testing was found to be applicable for testing TFS and TFCS features of the program block. White box type of testing could be employed more than at present, but these methods are generally more complex and more expensive. All in all, no major improvements were seen necessary in the testing process, but ensuring that all test engineers have good enough skills in their work area will certainly enhance the testing process in general.

The observed features of the program block are now fully implemented and completely tested in module level testing. The implementation has also been improved to some extent, which was one of the main goals of this thesis. The work with the observed program block and with the software architecture change project will continue after the completion of this thesis, and the obtained results will act as a reference for any future improvements possibly to be made in the program block.

8 References

3GPP (2005), *"Technical Report TR 25.808, FDD Enhanced Uplink; Physical layer aspects"*, version 6.0.0 Release 6.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25808.htm

3GPP (2007), *"Technical Report TR 25.848, Physical layer aspects of UTRA High Speed Downlink Packet Access"*, version 4.0.0 Release 4.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25848.htm

3GPP (2007), *"Technical Report TR 25.896, Feasibility study for enhanced uplink for UTRA (FDD)"*, version 6.0.0 Release 6.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25896.htm

3GPP (2004), *"Technical Report TR 25.899, High Speed Download Packet Access (HSDPA) enhancements"*, version 6.1.0 Release 6.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25899.htm

3GPP (2007), *"Technical Specification TS 25.201, Physical layer – general description"*, version 7.3.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25201.htm

3GPP (2007), *"Technical Specification TS 25.211, Physical channels and mapping of transport channels onto physical channels (FDD)"*, version 7.2.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25211.htm

3GPP (2007), *"Technical Specification TS 25.212, Multiplexing and channel coding (FDD)"*, version 7.5.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25212.htm

3GPP (2007), *"Technical Specification TS 25.213, Spreading and modulation (FDD)"*, version 7.2.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25213.htm

3GPP (2007), *"Technical Specification TS 25.214, Physical layer procedures (FDD)"*, version 7.5.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25214.htm

3GPP (2007), *"Technical Specification TS 25.301, Radio interface protocol architecture"*, version 7.2.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25301.htm

3GPP (2007), *"Technical Specification TS 25.302, Services provided by the physical layer"*, version 7.4.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25302.htm

3GPP (2007), *"Technical Specification TS 25.306, UE radio access capabilities"*, version 7.4.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25306.htm

3GPP (2007), "*Technical Specification TS 25.321, Medium Access Control (MAC) protocol specification"*, version 7.5.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25321.htm

3GPP (2007), *"Technical Specification TS 25.322, Radio Link Control (RLC) protocol specification"*, version 7.3.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25322.htm

3GPP (2007), *"Technical Specification TS 25.331, Radio Resource Control (RRC); Protocol specification"*, version 7.5.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25331.htm

3GPP (2007), *"Technical Specification TS 25.401, UTRAN overall description"*, version 7.4.0 Release 7.
Retrieved August 1st, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25401.htm

3GPP (2007), *"Technical Specification TS 25.413, UTRAN Iu interface Radio Access Network Application Part (RANAP) signalling"*, version 7.7.0 Release 7.
Retrieved November 6th, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/25413.htm

3GPP (2007), *"Technical Specification TS 26.071, AMR speech codec; General Description"*, version 7.0.1 Release 7.
Retrieved October 30th, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/26071.htm

3GPP (2007), *"Technical Specification TS 26.171, Adaptive Multi-Rate - Wideband (AMR-WB) speech codec; General description"*, version 7.0.0 Release 7.
Retrieved October 30th, 2007, from
http://www.3gpp.org/ftp/Specs/html-info/26171.htm

3GPP (2005), *"TSG RAN Meeting #28, RP-050248"*.
Retrieved June 4th, 2007, from
http://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_28/Docs/PDF/RP-050248.pdf

3GPP (2005), *"TSG RAN Meeting #28, RP-050250"*.
Retrieved June 4th, 2007, from
http://www.3gpp.org/ftp/tsg_ran/TSG_RAN/TSGR_28/Docs/PDF/RP-050250.pdf

3GPP (1999), *"TSG RAN Working Group 1 Meeting #7, TFCI mapping"*
Retrieved November 21st, 2007, from
http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_07/Docs/Pdfs/R1-99b33.pdf

Beizer, B (1995), *Black Box Testing: Techniques for Functional Testing of Software and Systems*
John Wiley & Sons Ltd, New York. ISBN: 0-471-12094-4

Björninen, T (2006), *"Performance Analysis of Transport Format Combination Selection in WCDMA Enhanced Uplink"*
Retrieved November 12th, 2007, from
http://www.netlab.hut.fi/opetus/s383310/05-06/Kalvot%2005-06/Bjorninen_310106.ppt

Cornett, S (1996), *"Code Coverage Analysis"*
Bullseye Testing Technology
Retrieved October 9th, 2007, from
http://www.bullseye.com/coverage.html

Freedman D. P. & Weinberg G. M. (1990), *Handbook of Walkthroughs, Inspections, and Technical Reviews*, 3rd edition,
Dorset House Publishing, New York. ISBN: 0-932633-19-6

Ghadialy, Z (2004), *"Calculating CTFC (Calculated Transport Format Combination)"*
Retrieved November 21st, 2007, from
http://www.3g4g.co.uk/Tutorial/ZG/zg_ctfc.html

Heier, S & Malkowski, M (2002), *"UMTS Radio Resource Management by Transport Format Assignment and Selection"*, Proceedings of WPMC 2002 - The 5th International Symposium on Wireless Personal Multimedia Communications
Retrieved August 1st, 2007, from
http://comnets.rwth-aachen.de/437+M535fa143fe0.pdf

Holma, H & Toskala, A (2006), *HSDPA / HSUPA for UMTS*,
John Wiley & Sons Ltd, Chichester. ISBN: 978-0-470-01884-2

Holma, H & Toskala, A (2007), *WCDMA for UMTS, Radio Access for Third Generation Mobile Communications*, 4th edition,
John Wiley & Sons Ltd, Chichester. ISBN: 978-0-470-31933-8

IEEE (1998), *"Std 1028-1997 IEEE Standard for Software Reviews"*,
Institute of Electrical and Electronics Engineers.

Kaner, C (2004), *"What Is a Good Test Case"*
Retrieved October 16th, 2007, from
http://testingeducation.org/articles/what_is_a_good_test_case_star_2003_paper.pdf

Kujala, J & Martinsuo, M & Artto, K (2006), *Projektiliiketoiminta*,
WSOY, Helsinki. ISBN: 951031482X

Laiho, J & Wacker, A & Novosad, T (2002), *Radio Network Planning and Optimisation for UMTS*,
John Wiley & Sons Ltd, Chichester. ISBN: 0-471-48653-1

Meyers, G (1979), *The Art of Software Testing*,
John Wiley & Sons Ltd, New York. ISBN: 0-471-04328-1

Ojanperä, T & Prasad R (1998), *Wideband CDMA for Third Generation Mobile Communications*,
Artech House Publishers, Norwood, MA. ISBN:0-89006-735-X

Pressmann, R. S. (2005), *Software Engineering: A Practitioner's Approach*, 6th edition,
McGraw-Hill, New York. ISBN: 007-123840-9

Proakis, J (2007), *Digital Communications*, 5th edition,
McGraw-Hill, New York. ISBN: 0072957166

Soldiani, D & Li, M & Cuny, R (2006), *QoS and QoE Management in UMTS Cellular Systems*,
John Wiley & Sons Ltd, Chichester. ISBN: 978-0-470-01639-8

Tanner, R & Woodard, J (2004), *WCDMA Requirements and Practical Design*,
John Wiley & Sons Ltd, Chichester. ISBN: 0-470-86177-0

Valtanen, J (2007), "*Transport formats in WCDMA physical layer operation*", Special Assignment,
Helsinki University of Technology, Department of Elect. and Comm. Engineering.

Vartiainen, M, Rantamäki, T, Hakonen, M, Simola, A (1999), *Tuotekehityksen palkitseminen*,
Metalliteollisuuden kustannus, Helsinki. ISBN: 951-817-702-3

Virkki, O (2007), "*High Speed Uplink Packet Access Effects in Radio Network Controller's Software Module Testing*", M.Sc. Thesis,
Helsinki University of Technology, Department of Computer Science and Engineering.

Williams, L (2004), *"White-Box Testing"*
Open Seminar in Software Engineering
Retrieved October 16th, 2007, from
http://agile.csc.ncsu.edu/SEMaterials/WhiteBox.pdf

Appendix 1

Physical channel descriptions

Table I - Dedicated uplink physical channels

Physical channel	***Description***
DPDCH	Carries the DCH transport channel. There may be zero, one or several DPDCHs on each radio link.
DPCCH	Carries control information generated at L1 (e.g. known pilot bits, TPC commands and the TFCI). The TFCI informs the receiver about the instantaneous TF of the simultaneously transmitted uplink DPDCH radio frame. There is always only one uplink DPCCH on each radio link.
E-DPDCH	Carries the E-DCH transport channel. There may be zero, one or several E-DPDCH on each radio link.
E-DPCCH	Carries control information associated with the E-DCH. There is at most one E-DPCCH on each radio link. E-DPDCH and E-DPCCH are always transmitted simultaneously, except some discontinuous transmission (DTX) related features.
HS-DPCCH	Carries uplink feedback signalling related to downlink HS-DSCH transmission. This consists of Hybrid-ARQ Acknowledgements (HARQ-ACK) and Channel Quality Indications (CQI). The spreading factor of the HS-DPCCH is fixed to 256.

Table II - Dedicated downlink physical channels

Physical channel	***Description***
DPCH	Carries the DCH transport channel time-multiplexed with control information generated at Layer 1 (e.g. known pilot bits, TPC commands and the TFCI). Can be considered as a time multiplex of a downlink DPDCH and a downlink DPCCH.
F-DPCH	Carries the control information generated at Layer 1 (TPC commands). This is a special case of downlink DPCCH.
E-RGCH	Carries the uplink E-DCH relative grants and indicates to the UE whether to increase, decrease or keep unchanged the transmit power level of the E-DCH. The spreading factor of the E-RGCH is fixed to 128.
E-HICH	Carries the uplink E-DCH hybrid ARQ acknowledgement indicators from the Node B to the UE. The spreading factor of the E-HICH is fixed to 128.

Table III - Common downlink physical channels

Physical channel	***Description***
CPICH	Carries a pre-defined bit sequence to aid the channel estimation at the terminal. This unmodulated channel, scrambled with a cell-specific scrambing code, provides a reference signal e.g. for coherent detection, cell acquisition and handovers. The spreading factor of the CPICH is fixed to 256.
P-CCPCH	Carries the BCH transport channel. The parameters of this channel contain no flexibility and they need to be known by all terminals made since the publication of Release '99 specifications. P-CCPCH alternates with the Synchronisation Channel (SCH). The spreading factor of the P-CCPCH is fixed to 256.
S-CCPCH	Carries the FACH and PCH transport channels. These two can share a single S-CCPCH or can use different physical channels. All terminals in the network need to be able to decode this channel. The spreading factor of the S-CCPCH is fixed according to the maximum data rate.
SCH	Used for cell search purposes. Consists of two sub channels, the Primary and Secondary SCH. No transport channels are mapped onto the SCH. The SCH is time multiplexed with the P-CCPCH.
AICH	Carries Acquisition Indicators (AI) and is used to indicate from the base station the reception of the random access channel signature sequence. The spreading factor of the AICH is fixed to 256.
PICH	Carries Paging Indicators (PI) and is always associated with an S-CCPCH to which a PCH transport channel is mapped. The PICH is operated together with PCH to provide terminals with efficient sleep mode operation. The spreading factor of the PICH is fixed to 256.
HS-SCCH	Carries signalling related to HS-DSCH transmission and enables the demodulation of data on the HS-DSCH. In the case of retransmission or an erroneous packet, it performs the possible physical layer combining of the data sent on the HS-DSCH. The spreading factor of the HS-SCCH is fixed to 256.
HS-PDSCH	Carries the HS-DSCH transport channel. It corresponds to one channelisation code of fixed spreading factor 16 from the set of channelisation codes reserved for HS-DSCH transmission. Multi-code transmission is allowed, which translates to the UE being assigned multiple channelisation codes in the same HS-PDSCH subframe, depending on its UE capability.
E-AGCH	Carries the uplink E-DCH absolute grants and provides an absolute power level above the DPDCH power level that the UE should adopt. The spreading factor of the E-AGCH is fixed to 256.
MICH	Carries MBMS notification indicators and is always associated with an S-CCPCH to which a FACH transport channel is mapped. The spreading factor of the MICH is fixed to 256.

Appendix 2

Physical channel frame types

In the following, frame structures for those five physical channels that include TFCI information are presented (Figure I - Figure IV): the uplink DPDCH, the uplink DPCCH, the downlink DPCH, the PRACH and the S-CCPCH. Frame structures for other physical channels are presented in 3GPP TS 25.211, "Physical channels and mapping of transport channels onto physical channels (FDD)", (2007).

With these physical channels, each radio length of length 10 ms is split into five subframes, each of 3 slots, each of length 2560 chips, corresponding to one power-control period. The exact mapping of TFCI bits onto slots is described in 3GPP TS 25.212, "Multiplexing and coding (FDD)", (2007).

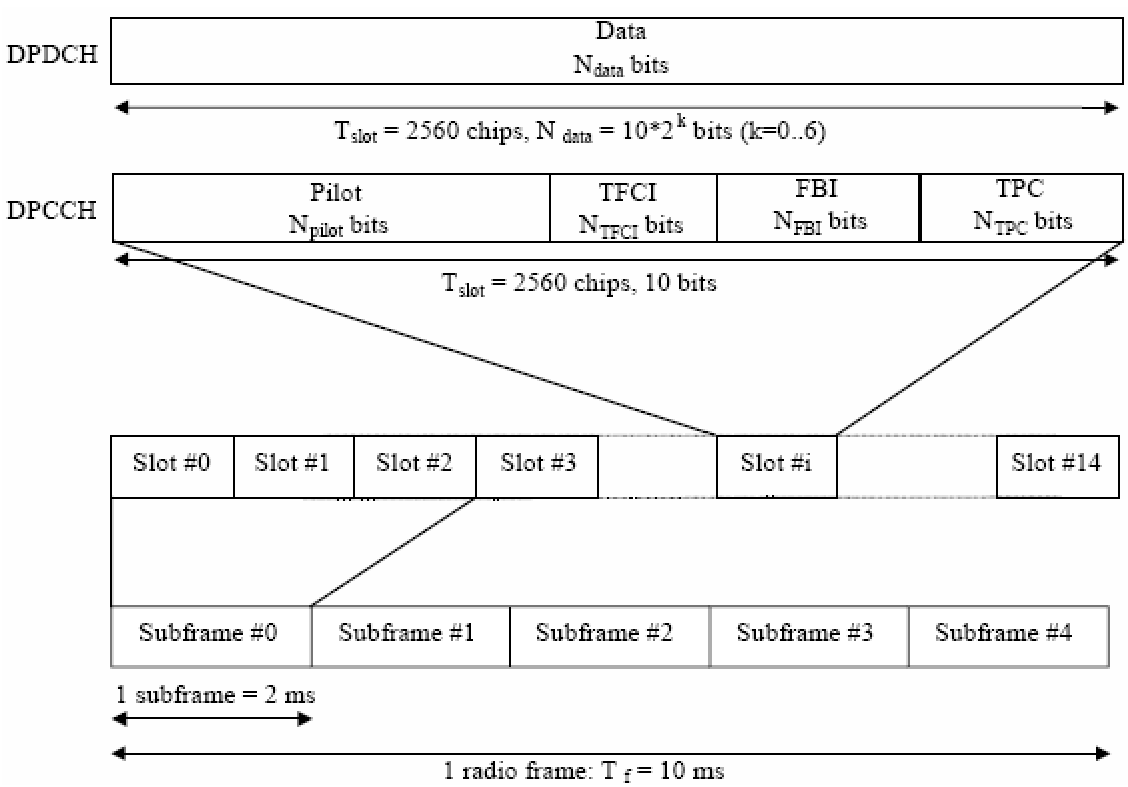

Figure I - Frame structure for the uplink DPDCH and DPCCH (from 3GPP TS 25.211, 2007)

As can be seen from the figure above, the uplink DPDCH and the uplink DPCCH are always frame aligned with each other.

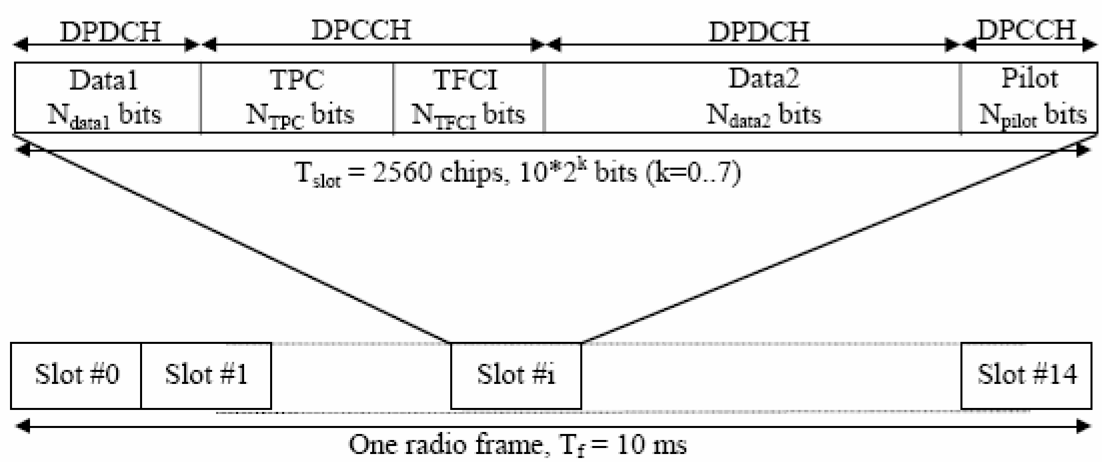

Figure II - Frame structure for the downlink DPCH (i.e. the downlink DPDCH and the downlink DPCCH, from 3GPP TS 25.211, 2007)

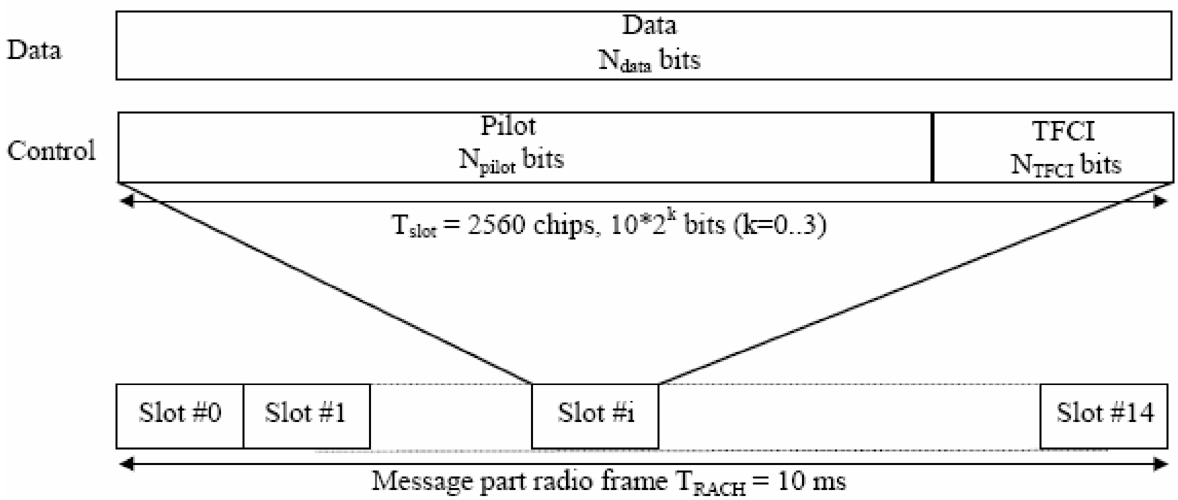

Figure III - Frame structure for the PRACH message part (from 3GPP TS 25.211, 2007)

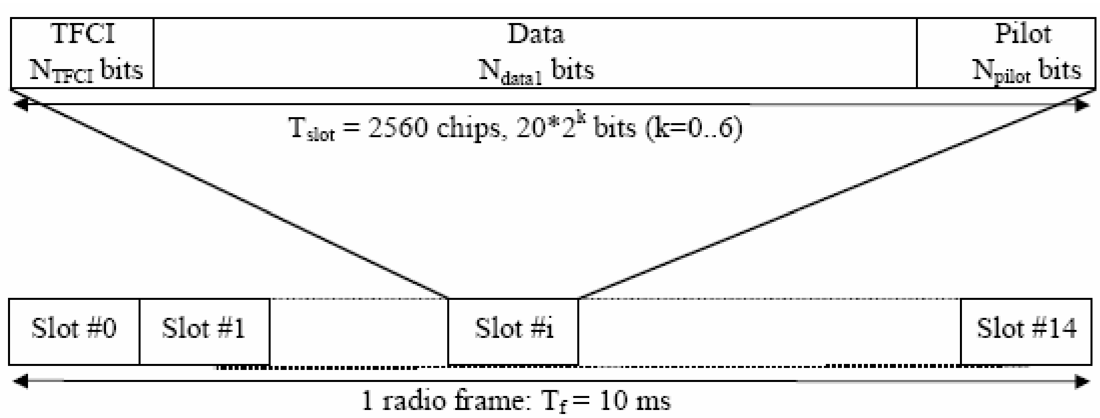

Figure IV - Frame structure for the S-CCPCH (from 3GPP TS 25.211, 2007)

Multicode transmission may be employed in the downlink. This means that the CCTrCH is mapped onto several parallel downlink DPCHs using the same spreading factor. In this case, the Layer 1 control information is transmitted only on the first downlink DPCH. DTX bits are transmitted during the corresponding time period for the additional downlink DPCHs, see Figure V below. (3GPP TS 25.211, 2007, p. 23)

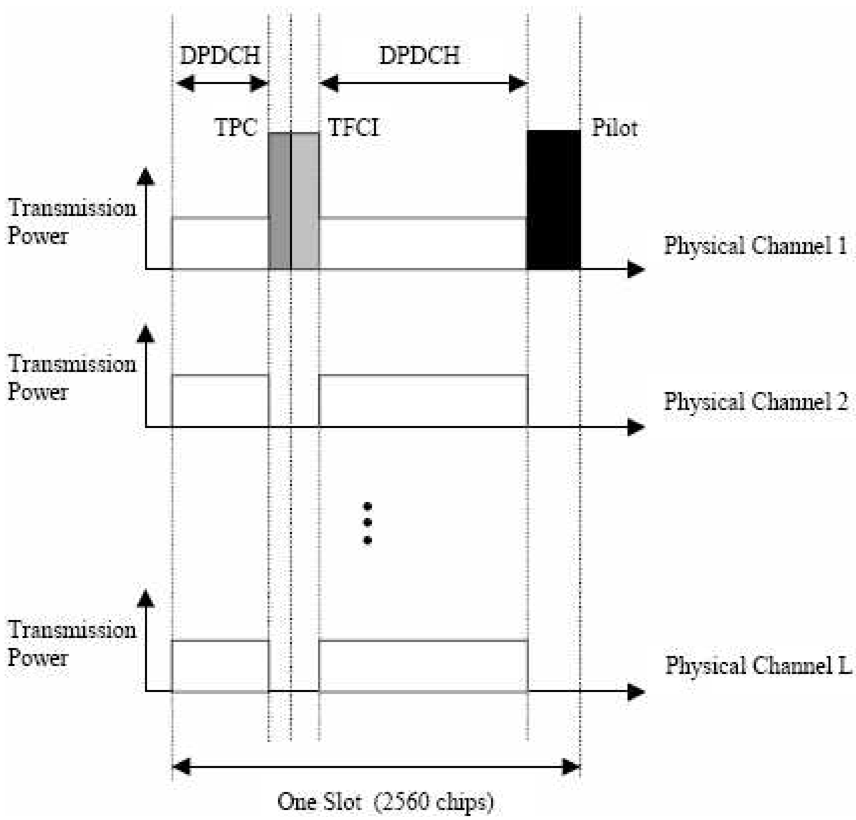

Figure V - Downlink slot format in case of multi-code transmission (from 3GPP TS 25.211, 2007)