



Readme Mac-hs PROCEDURE | VERSION 1.0.0

Read Me

MAC-hs Software Specification for UE Side



**Nex-G Exuberant Solutions Pvt. Ltd.
C-28 2nd floor sector-65
Noida UP-201301
INDIA**



Readme Mac-hs PROCEDURE | VERSION 1.0.0

Document Control:-

Document Reference Number	MAC Release 7.3.0
Document Name	Readme
Version	1.0.0
Issue Date	20-2-2013

	Name	Role	Signature
Authored by	Ankit Chaudhary Varun Chhangani Gagan Deep Sharma	Project Developer	
Approved by	Mr. Amit Kumar Sinha	Team Leader	
Released by	Mr. S.S Bhowmick	Manager	

Revision History:-

Name	Date	Status	Version



Readme Mac-hs PROCEDURE | VERSION 1.0.0

1.1 Purpose

This document provides the information about Medium Access Control Protocol requirements (MAC-hs) for UE side as per 3GPP release 7.3.0 specifications.

1.2 Documents and Conventions

As per Company Standards

1.3 Project Scope

To describe and demonstrate the Medium Access Control Protocol requirements for High Speed Downlink shared Channel in downlink direction on release 7.3.0 as per 3GPP specification.

1.4 Productive Perceptive

NEXP 4 is a terminal which will support 3G technologies. Our software will reside in ARM processor. The products will be supporting UMTS, HSPA, Wi-MAX and use shared memory as the software and communication Interface to store and perform procedures so as to provide a fast access and large Capacity for UE MAC layer Information Procedure. Radio Bearer Control procedure for state transition is being designed here. The product is scheduled to be release in 2013.

1.5 Product support

The product will support data rates of the order of 2 Mbps to 70 Mbps. It will be having Audio/Video interface with the peripherals like LCD screen.

1.6 Operating Environment

Currently simulating on Linux 11_04, final platform will be RTOS and ARM9 Processor.



Readme Mac-hs PROCEDURE | VERSION 1.0.0

Overview

The MAC-hs handle the HSDPA specific functions.

1)-HARQ:

The Hybrid Automatic Repeat Request entity is responsible for handling the MAC functions which are required for hybrid ARQ. It is generate ACKs or NACKs.

2)- Reordering :-The reordering queue distribution function routes the MAC-hs PDUs to the correct reordering buffer based on the Queue ID.

3)-Reordering:-The reordering entity reorders received MAC-hs PDUs according to the received TSN. MAC-hs PDUs with consecutive TSNs are delivered to the disassembly function upon reception. MAC-hs PDUs are not delivered to the disassembly function if MAC-hs PDUs with lower TSN are missing. There is one reordering entity for each QID configured at the UE.

4)-Disassembly:-

The disassembly entity is responsible for the disassembly of MAC-hs PDUs. When a MAC-hs PDU is disassembled the MAC-hs header is removed, the MAC-d PDUs are extracted and any present padding bits are removed. Then the MAC-d PDUs are delivered to higher layer.



Readme Mac-hs PROCEDURE | VERSION 1.0.0

Direction for Software usage

	Relaese 3GPP TS 25.321 version 7.3.0 Release 7	
	3GPP TS 34.123-1 version 7.0.0 Release 7	

- MAC -hs handles the HSDPA function.
- To execute this project we used Message queue for receiving and sending data.

\ STEP-1 \

- In mac_hs_IN directory there are all the files like send function, wrapper function, make file placed.
- In MAC - hs there are Eight process has been used and TSN is used which range vary between ZERO - EIGHT and ONE - SIXTYFOUR respectively.
- In mac_hs_IN there are data received file in send.c function.The main thing is that we didn't use the scanf function entire programme.
- For the input we divide the file in many parts and after that these dividing files and made a phy_pdu.txt file which all t data which we want to enter has been saved after that it's sent to send.c.

\ STEP-2 \

- send.c read the data from fread function from phy_pdu.txt.
- All the wrapper function defination has been given in same directory and declaration at mac_hs_include directory.
- send.c function received data send to Harq.c file which placed at mac_hs_stub directory.
- After that we create make file via command "make" .Compile the make file with Wall which also show hidden warnings.
- It's create executable file ./IN .



Readme Mac-hs PROCEDURE | VERSION 1.0.0

----- \ STEP-3 \ -----

- Execute of programme with. / IN data picked from logfile and send to harq function which placed at mac_hs_stub directory.
- According the received data by Harq function we received ACK it may be POSITIVE or NEGATIVE. When all the process has been completed the
- send function has been terminated.
- After that it's also checked with valgrind ./IN to check memory leakage.

- But we did not find any leakage.
- All the data which received or send in send function stored in mac_hs_logfiles directory as name of harq.txt.
- All the message queue has been also deleted after completion of all project.

----- \ STEP-4 \ -----

- To execute this project we used Message queue for receiving and sending data.
- In mac_hs_stub directory there are all the files like harq, rq, reorder, disassembly function, main file, wrapper function, make file placed.
- In mac_hs_stub all these module run one by one means in specific order which we called function it will execute.

----- \ STEP-5 \ -----

- In Harq.c function there are data has been received by send.c file which located at mac_hs_phy_stub directory,
- According to that Harq.c file match the data with received data and generate the ACK, it may be POSITIVE or NEGATIVE.
- If received data is matched then POSITIVE ACK generated else NEGATIVE ACK and this to send. mac_hs_phy_stub directory.



Readme Mac-hs PROCEDURE | VERSION 1.0.0

----- \ STEP-6 \ -----

- After that the data which received Harq.c file send to rq.c function which routes the MAC-hs PDUs to the correct reordering in buffer on QUEUE ID.
- After that data has been send to reorder.c.

----- \ STEP-7 \ -----

- Reorder.c function called after rq.c which received data from rq and it's reordering TSN in MAC-hs-PDU and reordered all the TSN.
- The data which received after POSITIVE ACK and Reordering Queue distribution then rearrange all the PDU TSN and QID.
- After that data send to Disassembly .If the MAC-hs PDU is not sent to Disassembly then lower TSN is missing.
- Final module is Disassembly.c in which MAC-hs header has been removed and Padding bit has been also removed and sent to MAC-D layer.

----- \ STEP-8 \ -----

- After that there are main file MAC_HS_OUT in which all function has been called.
- All the wrapper function definition has been given in same directory and declaration at mac_hs_include directory.
- After that we create make file via command "make".Compile the make file with make command Wall which also show hidden warnings.
- After compilation of make file new executable file ./MAC_HS_OUT has been created.

----- \ STEP-9 \ -----

- At the execution of programme with. / MAC_HS_OUT data received from send file which located at mac_hs_phy_stub directory and send to further function which placed at same directory.



Readme Mac-hs PROCEDURE | VERSION 1.0.0

According the received data from Send.c function it's received ACK it may be POSITIVE or NEGATIVE.

- When all the process has been completed the data with help of message queue send further to next module.
- After that it's also checked with valgrind ./IN to check memory leakage.
- But we did not find any leakage.
- All the data which received or mac_hs_stub directory stored in logfiles.
- All the message queue has been also deleted after completion of all project.