

## VeriSat DVB-RCS Hub Emulator

### Introduction

The VeriSat Hub Emulator is designed to offer a complete protocol test and verification environment for development and verification of DVB-RCS terminals. The Hub emulator allows detailed monitoring and analysis of the RCS protocol signalling and exchange of protocol messages and traffic in both directions.

A Hub emulator is an essential tool for efficient and cost-effective terminal development. The emulator environment also allows the user to test scenarios and situations that are difficult to establish when using a normal Hub.

The VeriSat Hub Emulator is used in the SatLabs DVB-RCS terminal qualification programme.

### Main features:

- Emulates an operational DVB-RCS Hub; full control of the forward link signalling
- Fully compliant with DVB-RCS specifications (EN 301 790)
- Powerful and user-friendly functions for controlling forward link signalling responses as a function of incoming terminal messages
- Supports MPEG and ATM cell formats and Turbo and convolutional coding
- Monitoring and analysis of forward and return link signalling (see the VeriSat FRLA data sheet)
- Generation and editing of forward link signalling, messages and tables
- User control of the forward link tables' play-out with respect to contents and repetition rates
- Event-driven control of the forward link signalling contents as a function of the received terminal bursts
- NCR options:
  - Internal generation and playout of NCR
  - If NCR is present on the forward link, the Hub emulator uses the incoming NCR to update NCR-synchronised parameters in SCT, TIM-U logon and TBTP
- Emulation of CMT or dynamic handling of CMT based on demodulator measurements of frequency and timing offsets
- Manual configuration of TBTP, synchronised to NCR
- Management and control of test case definition, configuration, implementation and execution
- Control of response to incoming messages such as capacity requests to accommodate the test case of interest

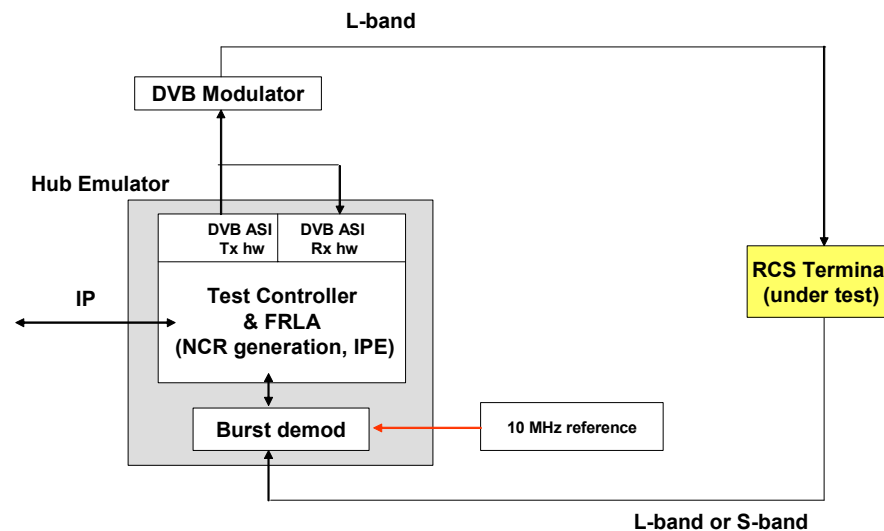


Figure 1. Block diagram of the Hub emulator with internal NCR generation and IP encapsulation.

## Hub Emulator Functional Description

### Hub emulator modules

The Hub emulator (HEM) consists of the following functional modules:

- Forward and Return Link analyser (FRLA) including a DVB-RCS burst demodulator, see data sheets for the FLA and FRLA products for detailed information.
- Forward link signalling generator and editor
  - Static tables (SI tables and broadcast TIM)
  - Dynamic handling of unicast TIM, CMT, TBTP and SCT
  - Internal NCR generation or synchronised to an externally generated NCR stream
- IP forwarding and encapsulation for forward link IP traffic to one terminal MAC address
- De-encapsulation and forwarding of return link IP traffic
- Transport stream multiplexer merging stored tables and dynamic tables in to a single ASI output stream output to the modulator
- Test cases are executed by:
  - Start sending defined forward link signalling
  - Wait for expected, defined return link message type or contents. The received message is compared with a defined filter.
  - When the received message matches the filter within a defined time-out, the next step in the test case is executed.
- Test cases may be created from scratch or edited based on existing test cases
- A test run may consist of a number of test cases that are grouped and then automatically executed

### Overview of forward link features

The main features for the forward link tables are described below. The tables are generated or edited and then played out based on user-defined test case rules.

NIT, PAT PMT, RMT Broadcast TIMs MMT, SPT	Table contents and descriptor fields may be created from scratch or loaded from a stored file and then edited. The tables are played out with configurable system parameters such as the PID and repetition rate. The PIDs are managed to maintain consistency according to the DVB SI recommendations and RCS specifications. The contents may be changed under test case execution. Transition points are user-defined (timers, counters, return link events).
SCT, FCT, TCT	These tables may be created with a user-friendly interface from scratch or edited at a parameter level by opening a stored table. The payout of the tables are repeated according to the configured repetition rate. The SCT counter value linked to the NCR is updated with in sync with the NCR (internal or external NCR depending on the configuration).
TBTP	TBTPs can be created and stored with a user-friendly interface or may be edited at a single parameter level. TBTPs are selected from a set of predefined tables depending on current test case rules and detected return link events, including reception of capacity requests or CSC bursts from the terminal (or several terminals for the multiple terminal option). The NCR-dependent counter values are updated created in sync with NCR.
Unicast TIM	Unicast TIMs can be generated from scratch or edited from stored files. Unicast TIM messages are normally sent from the hub as a result of events or messages received from a terminal such as CSC. Test case execution rules determine the TIM to be sent as a result of an input event or the absence of an input event within a defined time.
CMT	CMT messages are normally created as a result of burst demod measurements on bursts received from a terminal. Such input can be generated and transmitted on the forward link as a result of actual burst demodulator messages, or with user-defined contents independent of received terminal bursts replacing the burst demodulator measurements for special test cases.
NCR	NCR may be generated in two modes: <ul style="list-style-type: none"> <li>• External -NCR is generated by external equipment that may be present in an operational forward link. The analyser module receives the NCR. The table generator module receives the current NCR value and synchronises and updates all NCR-related counter values in the relevant tables.</li> <li>• Internal - NCR is generated internally by the NCR generator module, and all relevant table values are updated before payout in sync with the generated NCR.</li> </ul>
IP traffic	IP communication between the Hub Emulator LAN and the LAN connected to the terminal under test. If more than one IP stream to several terminals (several MAC addresses) is required, this may be offered at request.

### Overview of capabilities for handling return link signalling and traffic

The return link bursts are received, analysed and compared with the actions defined in the test case under execution. All DVB-RCS burst types and formats are supported (CSC, ACQ, SYNC and TRF). All MAC options are supported with the exception of DULM that is offered as an option at request.

CSC	When a CSC burst is received, the emulator may be configured to respond with a TIM message defined by the user.
ACQ	Support for coarse synchronisation by detecting ACQ messages and generating the corresponding FL messages
SYNC	Reception of bursts can generate CMT messages with burst demodulator measurements or simulated, user-defined contents for test purposes. The user may define actions to be taken based on the reception of signalling inside the SYNC burst (SAC fields).
TRF	The emulator receives the TRF bursts and divides the message into IP traffic and signalling messages. The IP traffic sections of the bursts are assembled into IP packets and analysed in the FRLA module. The signalling messages are handled as described in the following.
SAC	The expected SAC field contents are created and stored or retrieved prior to the test case execution. Forward link events are triggered by the reception of the pre-defined SAC message. SAC fields may be present in SYNC or TRF bursts as described in the DVB-RCS standard.
Capacity requests	RBDC, VBDC, AVBC are decoded and may trigger events and messages as defined in the test case under execution such as changing the TBTP. The capacity requests are received corresponding to the applied MAC method. DULM is not supported in the standard version but may be supported at request.
IP	The IP packets assembled from TRF bursts are forwarded onto the LAN for further routing to the destination IP address.

### Test case generation and execution

The Test Case Generator allows the user to define, store and recall test case sessions including forward link signalling, expected return link events and actions to be taken when such events are detected. Test cases are stored with information about the tables to be generated, playout time, parameter changes, repetition rate etc. Support is provided for dynamic generation of test cases, such as input on tables and/or sequences to be generated. Provision is supported to handle negative test cases, i.e. missing tables, time-outs waiting for events and wrong received parameters.

Expected responses and inputs from a terminal are specified as part of the test case. Received responses are validated with respect to expected responses, and corresponding test reports may be generated. On timeout of an expected response from terminal, an alternative test case or actions may be triggered.

Test cases may be chained in sequence, small test cases may be used as building blocks in larger test cases. This approach allows a user-friendly and well-structured test case creation also suited for creating more complex test cases by dividing the test into smaller test cases that are chained. The status of a completed test may be used as input to the next test case.

The Test Case Handler loads the selected test case or set of test cases, and starts executing the tests in the defined sequence, reporting the progress to the user.

### Report Generation

Log files are generated for the test cases that are executed stating the result of the test cases that have been run. The log stores all messages that are generated and received.

## DVB-RCS Hub Emulator Technical Specifications

---

### General

Analysis	Analysis of all forward and return link data
Forward link generator	Ability to create from scratch or modify forward link signalling data
Traffic handling	Forwarding of IP traffic to the SIT and reception of IP traffic from the SIT
Test case management	Allows for creation, modification, deletion, store, load and execution of test cases.
Communication tracing	View history of events and current status of each test case with all messages, events and errors logged.
Return link receiver	DVB-RCS burst demod (ATM or MPEG)
Target test environment	Single SIT under test (multiple terminals may supported as an option).

### Hub Emulator external interfaces

Forward link output	L-band (optionally 70 MHz or DVB ASI)
Return link input	L-band
IP Traffic interface	10/100 base T Ethernet

### Forward Link Generator

SI table generation	Creation or editing of previously captured tables
DVB-RCS Tables supported	NIT, PAT, PMT, RMT, TIM, SPT, TBTP, CMT, SCT, FCT, TCT, MMT
SPT	Provision to specify satellite co-ordinates in ASCII file
CMT	Corrections from burst demodulator are updated to the user-defined CMT template and transmitted
Unicast TIM	User defined rules govern which predefined unicast TIM is to be transmitted
TBTP	Manual configuration of TBTP. User defined rules govern which TBTP is to be transmitted in the next superframe.
NCR	NCR generation
Table play out	Updating of SCT, FCT, TCT and TBTP tables with current NCR values Play out of generated stream according to user-configured PID and repetition rates
Configuration management	Store and retrieve all tables including PID and repetition rates
IP encapsulation	Encapsulation of IP data according to DVB-S and MPE specifications

### Test Case Management

Test case generation	Test cases are created by using a set of commands applied on defined objects and events. The user selects a command from a predefined set of rules defining how each step of the test case shall be executed. Based on the result of each step, the test case flow is determined by the detected events and success or failure for a defined criteria to occur. After creation, test cases may be stored, opened, modified, deleted. Loaded test cases may be executed. Test results are used as input criteria for following tests. Test cases may be grouped as test suite and executed automatically in the defined sequence.
Test case execution	The loaded test case is executed, performing the defined actions and responding to the received events according to the applied test case rule.
Test case objects	All test case inputs and events are defined as test case objects, i.e. each FL table, each RL event. These objects are created or selected from existing objects by the user prior to test case execution. Different sets of object contents may be selected, allowing the same test case to be re-used when only the object contents change, for instance when changing from Ku-band to Ka-band. Each object is labelled with a unique label.

### Ordering information

VS510	DVB-RCS Hub Emulator
-------	----------------------

Revision 1.6

Copyright © Verisat 2006. Specifications are subject to change without notice.