

## IDD L3-Server

ICT Embedded B.V.			
<b>Project</b>		<b>Document ID</b>	ICT2001050304.2.17-2
<b>Author(s)</b>	G.J. Steinfeldter, J.F. Meijer	<b>Status</b>	Final
		<b>Date</b>	27 Aug. 2001
		<b>Classification</b>	Restricted
	Prepared for double-side printing	<b>Version</b>	2.0
Printed	08/03/2015 21:31:00		
Filename	IDD_L3server		

## Abstract

This document describes the interfacing details for the Layer-3 work package within the LTS system. The Interface Design Document (IDD) is used to supplement the Architectural Design Document (ADD) and Detailed Design Document (DDD). The Interface Design Document serves to communicate and control interface design decisions.

Obsolete

### History

Version	Date	Author	Description
0.1	9 Aug. 2001	J.F. Meijer, G.J. Steinfelder	Initial version
0.2	20 Aug. 2001	J.F. Meijer	Update after meeting with Condat
0.3	24 Aug. 2001	G.J. Steinfelder	Processed comments from Ronny Kießling
1.0	27 Aug. 2001	G.J. Steinfelder	Processed additional comments from Ronny Kießling and made document final.
2.0	13 Sep. 2001	G.J. Steinfelder	Added L3SRV_IS_LOGGING_ENABLED and L3SRV_LOGGING_ENABLED.

### Distribution

Version	Date	To	Company
0.1	9 Aug. 2001	D. Kreuer; R. Kießling	Condat
0.2	21 Aug. 2001	D. Kreuer; R. Kießling	Condat
0.3	24 Aug. 2001	D. Kreuer, R. Keißling	Condat
1.0	27 Aug. 2001	D. Kreuer, R. Keißling	Condat
2.0	13 Sep. 2001	D. Kreuer, R. Keißling	Condat

This document is property of ICT Embedded B.V. No part of it may be reproduced or used in any form or by any means without written permission of the owner.

If the ICT logo is not printed in green print, this is an unofficial copy.

© 2001 ICT Embedded B.V. All rights reserved.

# Table of contents

<b>1. Introduction</b> .....	<b>6</b>
1.1 Purpose .....	6
1.2 Terms and abbreviations .....	6
1.3 References .....	6
<b>2. System overview</b> .....	<b>7</b>
2.1 Layer-3, Logging and presenting .....	7
2.2 External interfaces of L3Server .....	7
<b>3. Interface L3Server - L3Viewer</b> .....	<b>8</b>
3.1 Interface description.....	8
3.2 Interface messages .....	8
3.3 Scenario's .....	9
3.3.1 Show information of a running test case.....	9
3.3.2 Show information of a logged test case.....	9
3.3.3 Save filtered information in a new file .....	10
3.4 Detailed description of the messages.....	11
3.4.1 L3SRV_CONNECT .....	11
3.4.2 L3SRV_DISCONNECT.....	11
3.4.3 L3SRV_SUBSCRIBE .....	12
3.4.4 L3SRV_UNSUBSCRIBE .....	12
3.4.5 L3SRV_LOAD_LOG_FILE .....	13
3.4.6 L3SRV_SET_SERVER_FILTER.....	13
3.4.7 L3SRV_COPY_LOG_FILE .....	14
3.4.8 L3SRV_EXIT.....	14
3.4.9 L3SRV_DATA.....	15
3.4.10 L3SRV_CONNECTED.....	15
3.4.11 L3SRV_OK.....	15
3.4.12 L3SRV_ERROR.....	16
<b>4. Interface L3Server – LTS</b> .....	<b>17</b>
4.1 Interface description.....	17
4.2 Interface messages .....	17
4.3 Scenario's .....	17
4.3.1 Set filters into a mobile.....	17
4.4 Detailed description of the messages.....	18
4.4.1 L3SRV_SET_MOBILE_FILTER .....	18
4.4.2 L3SRV_GET_MOBILE_FILTER .....	19
4.4.3 L3SRV_ENABLE_LOGGING .....	19
4.4.4 L3SRV_IS_LOGGING_ENABLED .....	20
4.4.5 L3SRV_LOG_FILE .....	20
4.4.6 L3SRV_MOBILE_FILTER .....	21
4.4.7 L3SRV_LOGGING_ENABLED .....	21
4.4.8 TEXE_STARTED_TC.....	22
4.4.9 TEXE_RESULT .....	22
<b>5. Interface L3Server – L3Driver</b> .....	<b>24</b>
5.1 Interface description.....	24
5.2 Interface messages .....	24
5.3 Scenario's .....	24
5.3.1 Set filters into a mobile.....	24
5.4 Detailed description of the messages.....	25
5.4.1 L3DRV_DATA .....	25

5.4.2 L3DRV_SET_MOBILE_FILTER.....	25
5.4.3 L3DRV_OK .....	26
5.4.4 L3DRV_ERROR .....	26
<b>6. Interface LTS– L3Viewer.....</b>	<b>27</b>
6.1 Interface description.....	27
6.2 Interface messages .....	27
6.3 Detailed description of the messages.....	27
6.3.1 L3VWR_TO_FRONT.....	27
6.3.2 L3VWR_OK.....	27
<b>7. Interface L3Driver – TST Mobile.....</b>	<b>28</b>
7.1 Interface characteristics.....	28
7.2 Message structure.....	28
7.3 Interface messages .....	28
7.3.1 Setting of trace filters .....	29
7.3.2 Duplicating messages to LTS .....	29
7.3.3 L3 Primitives.....	30

Obsolete

# 1. Introduction

## 1.1 Purpose

This document describes the interfacing details for the Layer-3 work package. This Interface Design Document (IDD) is intended to extend the global design as described in the Architectural Design Document for Layer-3

## 1.2 Terms and abbreviations

Term	Description
Primitive	A structure sent from one entity to another within the GSM stack of the mobile
Trace	A text string sent by an entity to an outside test system
Air message	An encoded structure sent within a primitive

Abbreviation	Description
IDD	Interface Design Document

## 1.3 References

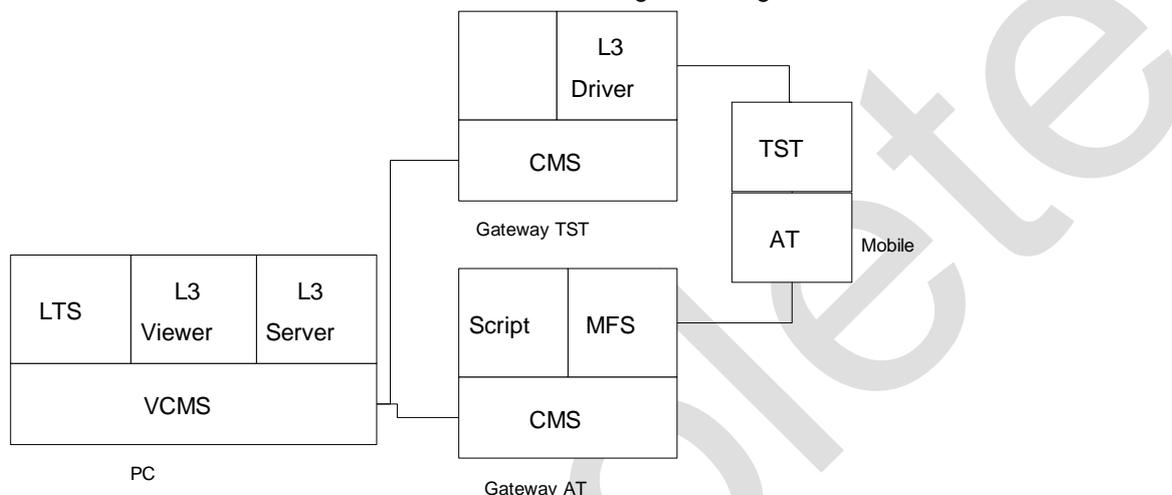
Reference	Document ID	Description
[C-GROBSP]	Condat 8201.400.98.1	Grobspezifikation: Condat LTS-G900 Load Test System for Network Manufacturers
[C-CMS]	Condat 8309.201.97.002	CMS 3 Handbuch, Condat GmbH, Frank Reglin & S.Kortmann
[C-TF]	Condat 8410.003.99.104	GSM Protocol Stack; TF – Test Facilities; Description
[C-PCO]	Condat 6519.022.01.001	PCO for LTS filtering; PowerPoint presentation; 16 Aug. 2001; Ronny Kiessling
[ICT-SRD2]	ICT 2001050304.2.3-1	Condat LTS-G, Part 2; Software Requirements Document; J.F. Meijer
[ICT-ADDL3]	ICT 2001050304.2.4-17	Condat LTS-G, Part 2; Layer 3 Architecture; G.J. Steinfeldler

## 2. System overview

The Condat LTS-G system is a testing system for GSM network infrastructure manufacturers and network providers, which tests the behaviour of infrastructure components under load conditions. "Load" is the traffic and signalling information caused by the subscriber mobile stations that must be processed by the system infrastructure entities.

### 2.1 Layer-3, Logging and presenting

One of the tasks within the LTS-System is logging and presenting layer-3 messages. The relevant processes of LTS involved in this tasks are shown in the block diagram of *Figure 1*.



*Figure 1: Processes of the Layer-3 applications*

These processes are:

- CMS and VCMS: the Condat Multitasking System, responsible, a.o., for the communications between the several processes.
- LTS: the main application, responsible a.o. for starting a viewer and setting filter parameters
- Scripts: the testscripts, running on the gateway, and communicating with the mobile through the MFS-AT interface. These scripts generate the network load.
- L3-Viewer: displays (filtered) air messages to the user
- L3-Server: stores primitives into a file and relays primitives to the L3Viewer
- L3-Driver: process on the TST gateway for the communication between the L3Server and the TST interface of the mobile

### 2.2 External interfaces of L3Server

The L3Server, to be developed by ICT, will have interfaces to the following processes:

- L3Server – L3Viewer
- L3Server – LTS
- L3Server – L3Driver

Besides these interfaces, the following interface are described in this document:

- LTS – L3Viewer
- L3Driver – TST-Mobile

## 3. Interface L3Server - L3Viewer

### 3.1 Interface description

The L3Server – L3Viewer interface handles the communication between the L3Server process and one or more L3Viewer processes. The interface uses the VCMS message methods as described in [C-CMS] for exchanging messages. See Appendix A for detailed message structure.

The responsibilities of an L3Viewer are:

- displaying L3 air messages, whether it is live data or stored in a log file (i.e. also replay functionality like play, pause),
- setting filters on the primitives in the server, so only selected air messages are displayed.
- searching through the received L3 air messages.

The L3Server shall serve the L3Viewer.

### 3.2 Interface messages

The following messages can be sent from a L3Viewer towards the L3Server:

- **L3SRV\_CONNECT**  
With this message the L3Viewer registers itself with the L3Server
- **L3SRV\_DISCONNECT**  
With this message the L3Viewer unregisters itself
- **L3SRV\_SUBSCRIBE**  
Causes the L3Server to forward live data from the mobile associated with the registered L3Viewer to that L3Viewer, until the L3Viewer unsubscribes itself.
- **L3SRV\_UNSUBSCRIBE**  
Cause the L3Server to stop forwarding live data, from the mobile associated with the L3Viewer, towards that L3Viewer.
- **L3SRV\_LOAD\_LOG\_FILE**  
Upon receipt of this message, the L3Server will open the logfile specified in the message. The data in this log file will be send to the requesting L3Viewer in reply messages (multiple reply messages are required, since one message can contain only a limited amount of data).
- **L3SRV\_SET\_SERVER\_FILTER**  
This message sets a filter on the primitives in the L3Server. Subsequent primitives will be filtered. Only primitives that passes the filter will be sent to the L3Viewer.
- **L3SRV\_COPY\_LOG\_FILE**  
Upon receipt of this request, the L3Server will load the logfile, filter the primitives using the last filter set, and write the filtered primitives to the file identified in the argument of the message.

The L3Server can send the following messages to the L3Viewer in reply on the requests above:

- **L3SRV\_DATA**  
This message contains primitives, either from the mobile directly or read from a log file. When the primitive is read from a log file, the time stamp of the L3SRV\_DATA message will be the time the primitive was received from the mobile.
- **L3SRV\_CONNECTED**  
This message is returned when the L3Server processed a connection request from an L3Viewer correctly.
- **L3SRV\_OK**  
This message is returned by the L3Server when a request (except the L3SRV\_CONNECT) from the viewer is processed correctly.
- **L3SRV\_ERROR**  
This message is returned by the L3Server when an error has occurred during the processing of a request from the viewer.

In addition to these reply messages, the L3Server can also send the following message towards the L3Viewer:

- L3SRV\_EXIT  
This message is send to the L3Viewer when the server exists, so the viewer can exit too.

### 3.3 Scenario's

The following scenario's will be described:

- Show (filtered) information of a running test case
- Show (filtered) information of a logged test case
- Save filtered information in a new file

#### 3.3.1 Show information of a running test case

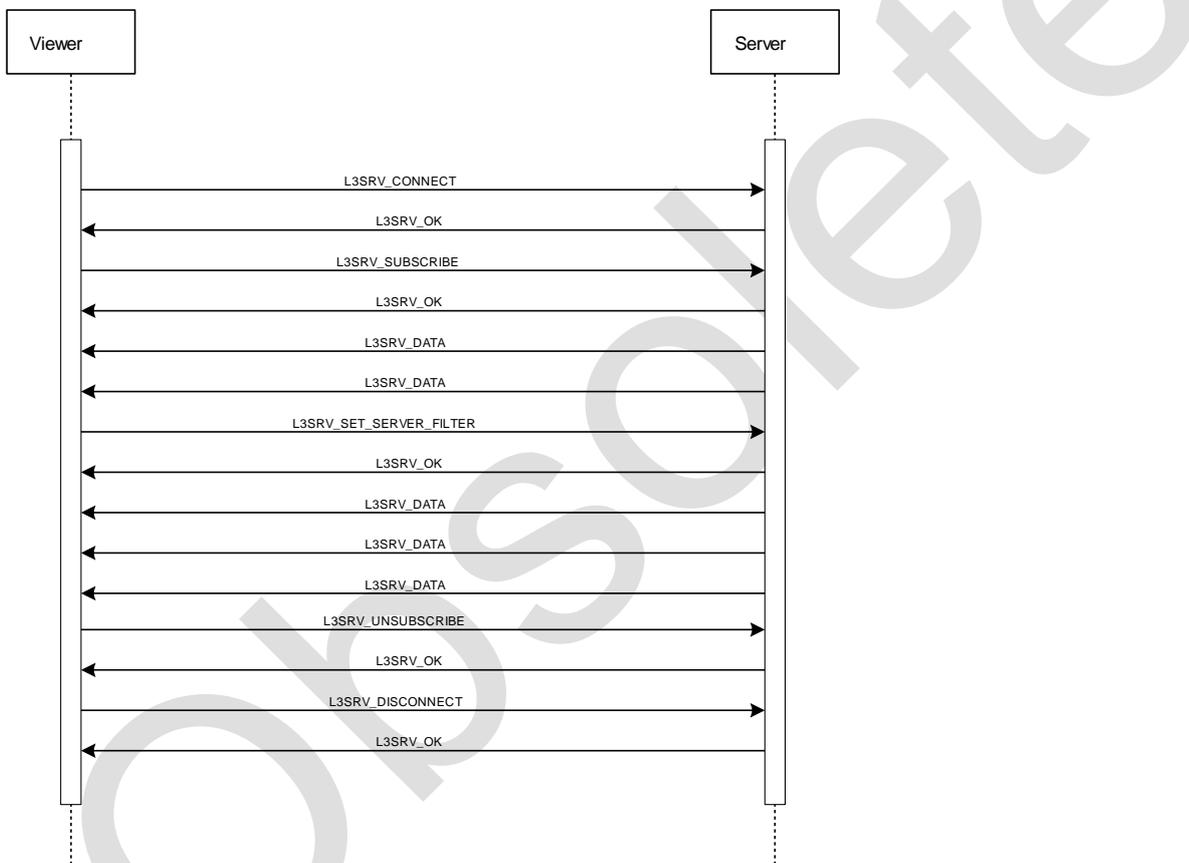


Figure 2 Sequence diagram for live data

Figure 2 shows a sequence diagram of a L3Viewer and the L3Server for viewing live data of a mobile. The viewer starts the communication by connecting and subscribing itself with the L3Server and informing the L3Server for which mobile it wants to receive primitives. The L3Server responds by numerous L3SRV\_DATA messages, sending one each time the mobile sends new information. During this process L3Viewer can request L3Server to set a filter. Subsequent primitives are then sent to the L3Viewer if, and only if, new received primitives pass this filter. The L3Server stops sending L3SRV\_DATA messages as soon as it receives an unsubscribe message from the associated L3Viewer.

#### 3.3.2 Show information of a logged test case

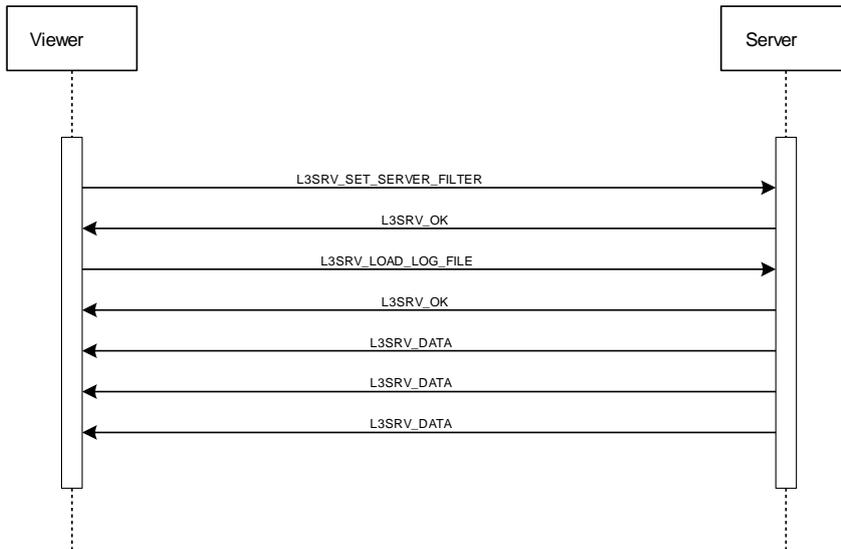


Figure 3 Sequence diagram for viewing stored data

Figure 3 shows a sequence diagram of a L3Viewer and the L3Server for viewing stored log data of a mobile. While the viewer is already connected to the server, it starts the communication by setting a filter in the L3Server and requesting data from the L3Server for a particular mobile. The L3Server responds by numerous L3SRV\_DATA messages, sending one for each message stored and that passed the filter. As the whole log file has been processed, the L3Server stops sending messages to the L3Viewer, and no further action is required.

### 3.3.3 Save filtered information in a new file

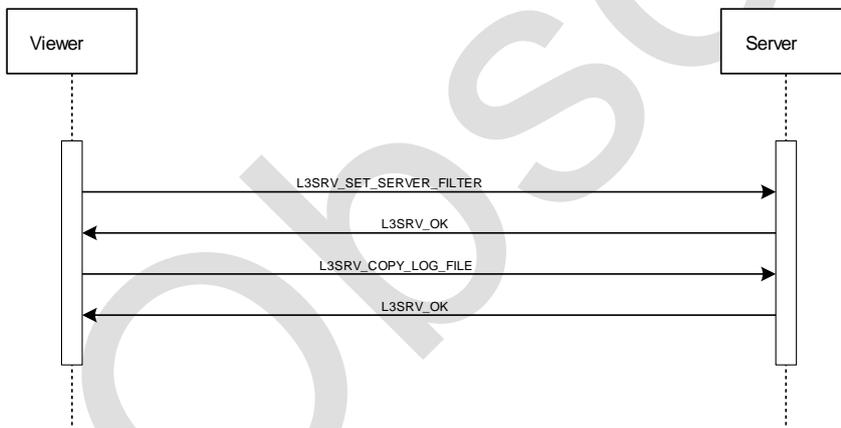


Figure 4 Sequence diagram for filter data and storing filtered data

Figure 4 shows a sequence diagram of a L3Viewer and the L3Server for filtering stored log data and writing the filtered primitives in a new log file. First of all the connected viewer has to set a filter in the L3Server. Then the L3Viewer request the L3Server to copy filtered data from a specific log file to a new log file.

## 3.4 Detailed description of the messages

### 3.4.1 L3SRV\_CONNECT

#### 3.4.1.1 Purpose

Establish a connection between an L3Viewer and the L3Server, and store communication information in the server.

#### 3.4.1.2 Pre-condition

Viewer is not connected

#### 3.4.1.3 Post-condition

Viewer is connected

#### 3.4.1.4 Responses

- L3SRV\_CONNECTED
- L3SRV\_ERROR

#### 3.4.1.5 Layout

<i>Message name</i>	L3SRV_CONNECT
<i>Sender</i>	L3Viewer
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	queueName	string	name of the queue used by the viewer to store the primitives (L3DRV_DATA) received from the server

### 3.4.2 L3SRV\_DISCONNECT

#### 3.4.2.1 Purpose

Release the connection between an L3Viewer and the L3Server.

#### 3.4.2.2 Pre-condition

Viewer must be connected

#### 3.4.2.3 Post-condition

Viewer is disconnected

#### 3.4.2.4 Responses

- L3SRV\_OK
- L3SRV\_ERROR

#### 3.4.2.5 Layout

<i>Message name</i>	L3SRV_DISCONNECT
<i>Sender</i>	L3Viewer
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>		
	<n.a.>	

### 3.4.3 L3SRV\_SUBSCRIBE

#### 3.4.3.1 Purpose

Subscribe viewer with the server to receive live data from the specified mobile.

#### 3.4.3.2 Pre-condition

Viewer must be connected and not subscribed

#### 3.4.3.3 Post-condition

After this request has been received, all data received by the L3Server for the specified mobile is relayed to the subscribed L3Viewer, until the L3Server receives an unsubscribe message for the same mobile and from the same L3Viewer. Filter settings at the L3Server are unchanged.

#### 3.4.3.4 Responses

- L3SRV\_OK followed by a number of L3SRV\_DATA messages, one for each primitive message that the L3Server receives from the mobile, and that passes the filter as set by the L3Viewer. If the mobile does not send any information to the L3Server, the L3Server will not send an L3SRV\_DATA message at all to the L3Viewer. The L3DRV\_DATA messages will be sent to the queue as defined in L3SRV\_CONNECT.
- L3SRV\_ERROR

#### 3.4.3.5 Layout

<i>Message name</i>	L3SRV_SUBSCRIBE
<i>Sender</i>	L3Viewer
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>		
1	mobileID	string e.g. "Mobile001"

### 3.4.4 L3SRV\_UNSUBSCRIBE

#### 3.4.4.1 Purpose

Unsubscribe viewer with the server, so it will not receive live data from the mobile anymore.

#### 3.4.4.2 Pre-condition

Viewer must be connected and subscribed.

#### 3.4.4.3 Post-condition

After this request has been received, no data received by the L3Server is relayed to the L3Viewer anymore. Filter settings at the L3Server are unchanged.

#### 3.4.4.4 Responses

- L3SRV\_OK
- L3SRV\_ERROR

#### 3.4.4.5 Layout

<i>Message name</i>	L3SRV_UNSUBSCRIBE
<i>Sender</i>	L3Viewer

Receiver	L3Server
----------	----------

Data Field (i.e. contents of pcBuffer as defined in Appendix A)		
	<n.a.>	

### 3.4.5 L3SRV\_LOAD\_LOG\_FILE

#### 3.4.5.1 Purpose

Loads a stored log file in the server and sends all information out of this log file towards the requesting L3Viewer. The forwarded data may or may not be filtered, depending on previous requests of the L3Viewer.

#### 3.4.5.2 Pre-condition

L3Server must be running, and the supplied file name must be of an existing and valid log file. A filter may be set.

#### 3.4.5.3 Post-condition

All data found in the supplied log file is filtered, and all data that passed the filter is forwarded to the requesting L3Viewer.

#### 3.4.5.4 Responses

- L3SRV\_OK followed by a number of L3SRV\_DATA messages, one for each primitive that the L3Server reads from the log file and that passes the filter. If the log file is empty or none of the primitives passes the filter, the L3Server will not send any L3SRV\_DATA message at all to the L3Viewer. The L3DRV\_DATA messages will be sent to the queue as defined in L3SRV\_CONNECT.
- L3SRV\_ERROR

#### 3.4.5.5 Layout

Message name	L3SRV_LOAD_LOG_FILE
Sender	L3Viewer
Receiver	L3Server

Data Field (i.e. contents of pcBuffer as defined in Appendix A)			
	name	type / length	defined values (description)
1	fileName	string	e.g. "c:\log\TestCase1\Mobile001\20010814101534.L3L"

### 3.4.6 L3SRV\_SET\_SERVER\_FILTER

#### 3.4.6.1 Purpose

Set a filter in the server, so that only a subset of primitives will be forwarded from the server to a viewer.

#### 3.4.6.2 Pre-condition

L3Viewer connected

#### 3.4.6.3 Post-condition

Only primitives sent by entities not listed in the filterset, will be forwarded to the viewer. Notice that this is different from L3SRV\_SET\_MOBILE\_FILTER (see section 4.4.1) and L3DRV\_SET\_MOBILE\_FILTER (see section 5.4.2).

#### 3.4.6.4 Responses

- L3SRV\_OK

- L3SRV\_ERROR

### 3.4.6.5 Layout

<i>Message name</i>	L3SRV_SET_SERVER_FILTER
<i>Sender</i>	L3Viewer
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	filterSet	char[]	zero separated entity names [MM,RR,SS,CC,SMS] ended with "\0\0" e.g.: "MM\0RR\0SS\0\0"

### 3.4.7 L3SRV\_COPY\_LOG\_FILE

#### 3.4.7.1 Purpose

Copies the filtered primitives to a new file.

#### 3.4.7.2 Pre-condition

Filter is already set.

#### 3.4.7.3 Post-condition

All primitives that passed the filter are copied to a new file.

#### 3.4.7.4 Responses

- L3SRV\_OK
- L3SRV\_ERROR

#### 3.4.7.5 Layout

<i>Message name</i>	L3SRV_COPY_LOG_FILE
<i>Sender</i>	L3Viewer
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	fileNameRead	string	
2	fileNameWrite	string	

### 3.4.8 L3SRV\_EXIT

#### 3.4.8.1 Purpose

Notifies the L3Viewer that the L3Server is about to exit.

#### 3.4.8.2 Pre-condition

L3Viewer is connected.

#### 3.4.8.3 Post-condition

N.a.

#### 3.4.8.4 Responses

None.

### 3.4.8.5 Layout

<i>Message name</i>	L3SRV_EXIT
<i>Sender</i>	L3Server
<i>Receiver</i>	L3Viewer

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
	n.a.		

### 3.4.9 L3SRV\_DATA

This message is the same as L3DRV\_DATA, see section 5.4.1. For clarity in the interface definition, the message has its own name.

If the data is loaded from a log file, the time stamp of the message will not be the actual time, but the time that the message was received from the mobile.

### 3.4.10 L3SRV\_CONNECTED

#### 3.4.10.1 Purpose

Notifies that the connect request was handled correctly.

#### 3.4.10.2 Pre-condition

L3Viewer connected successfully.

#### 3.4.10.3 Post-condition

N.a.

#### 3.4.10.4 Responses

None.

#### 3.4.10.5 Layout

<i>Message name</i>	L3SRV_CONNECTED
<i>Sender</i>	L3Server
<i>Receiver</i>	L3Viewer

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	serverID	byte	0x1E: L3SRV

### 3.4.11 L3SRV\_OK

#### 3.4.11.1 Purpose

Notifies that the viewer request (except for L3SRV\_CONNECT) was handled correctly.

#### 3.4.11.2 Pre-condition

L3Server handles a previous request from an L3Viewer correctly

#### 3.4.11.3 Post-condition

N.a.

#### 3.4.11.4 Responses

None.

### 3.4.11.5 Layout

<i>Message name</i>	L3SRV_OK
<i>Sender</i>	L3Server
<i>Receiver</i>	L3Viewer

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	commandRef	U16	message ID on which this OK is a response

### 3.4.12 L3SRV\_ERROR

#### 3.4.12.1 Purpose

Error message.

#### 3.4.12.2 Pre-condition

L3Server cannot handle the request from the viewer.

#### 3.4.12.3 Post-condition

N.a.

#### 3.4.12.4 Responses

None.

#### 3.4.12.5 Layout

<i>Message name</i>	L3SRV_ERROR
<i>Sender</i>	L3Server
<i>Receiver</i>	L3Viewer

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	commandRef	U16	message ID on which this error is a response
2	errorCode	byte	0x01: too many viewers 0x02: already connected 0x03: already subscribed 0x04: not connected 0x05: not subscribed 0x06: file not found 0x07: log file not for this viewer

## 4. Interface L3Server – LTS

### 4.1 Interface description

The L3Server – LTS interface handles the communication between the L3Server process and the GBFS and TEXE processes of LTS. The interface uses the VCMS message methods as described in [C-CMS] for exchanging messages.

The responsibilities of the LTS GBFS and TEXE processes are, amongst others:

- enabling/disabling primitives from the mobile,
- setting filters on primitives within the mobile,
- enabling logging by the L3 Server,
- disable logging by the L3 Server,
- informing the L3 Server when a test case starts (and supply the name of the test case), and
- informing the L3 Server when a test case has finished.

### 4.2 Interface messages

The following messages can be sent from LTS towards the L3Server:

- L3SRV\_SET\_MOBILE\_FILTER  
To select the entities for which primitives will not be duplicated
- L3SRV\_GET\_MOBILE\_FILTER  
To retrieve the entities for which primitives are not duplicated
- L3SRV\_ENABLE\_LOGGING  
To enable or disable the logging of primitives received from a mobile
- L3SRV\_IS\_LOGGING\_ENABLED  
To retrieve the current log setting for a specific mobile.
- L3SRV\_LOG\_FILE  
To inform LTS that a log file for L3 primitives has been opened or closed by the server.
- L3SRV\_LOGGING\_ENABLED  
To inform LTS on the current status of logging for a mobile.
- TEXE\_STARTED\_TC  
To inform the L3Server that the execution of a new case starts
- TEXE\_RESULT  
To inform the L3Server that a test case is finished or cancelled.

The L3Server can send the following message to LTS in reply on the requests above:

- L3SRV\_MOBILE\_FILTER  
Message contains actual filter at the server. If the mobile is running, this filter is also set at the mobile.

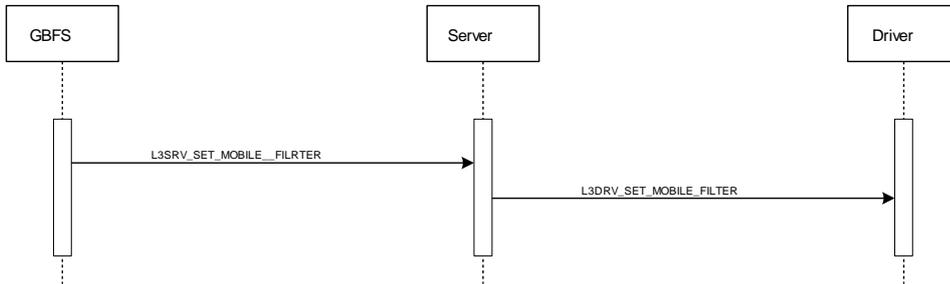
### 4.3 Scenario's

The following scenario will be described:

- Set filters into a mobile

Enable logging is analogue to setting filters.

#### 4.3.1 Set filters into a mobile



## 4.4 Detailed description of the messages

### 4.4.1 L3SRV\_SET\_MOBILE\_FILTER

#### 4.4.1.1 Purpose

Set a filter in the mobile, so that only the selected primitives will be forwarded from the mobile to the L3Driver.

#### 4.4.1.2 Pre-condition

L3Driver running, mobile OK

#### 4.4.1.3 Post-condition

Only primitives from entities listed in filter settings will be sent to the L3Driver. Notice that this is different from L3SRV\_SET\_SERVER\_FILTER (see section 3.4.6).

#### 4.4.1.4 Responses

None.

#### 4.4.1.5 Layout

<i>Message name</i>	L3SRV_SET_MOBILE_FILTER
<i>Sender</i>	GBFS
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	mobileID	string	
2	filterSettings	byte	logical or, filter set CC = 1 RR = 2 SS = 4 MM = 8 SMS = 16
3	opcodeFilter	dynamic array of 16-bytes entries	This field will not be used and no opcode mask will be sent to the mobile, until Condat defines the opcode mask. The mask can be hard-coded in LTS. For each bit set in the filtersetting, one set of 16 bytes should be placed in the opcodeFilterField, in the same order as the bits set in filterSettings, starting with the lowest bit.

## 4.4.2 L3SRV\_GET\_MOBILE\_FILTER

### 4.4.2.1 Purpose

Get a filter from the L3Server, as it is set in the mobile.

### 4.4.2.2 Pre-condition

None.

### 4.4.2.3 Post-condition

Filter setting as is currently set in mobile, or as it will be set in the mobile as soon as the mobile is switched on, will be returned.

### 4.4.2.4 Responses

- L3SRV\_MOBILE\_FILTER.

### 4.4.2.5 Layout

Message name	L3SRV_GET_MOBILE_FILTER
Sender	GBFS
Receiver	L3Server

Data Field (i.e. contents of pcBuffer as defined in Appendix A)			
	name	type / length	defined values (description)
1	mobileID	string	

## 4.4.3 L3SRV\_ENABLE\_LOGGING

### 4.4.3.1 Purpose

Forces L3Server to start/stop logging of the data it receives from L3Driver.

### 4.4.3.2 Pre-condition

n.a.

### 4.4.3.3 Post-condition

n.a.

### 4.4.3.4 Responses

None.

### 4.4.3.5 Layout

Message name	L3SRV_ENABLE_LOGGING
Sender	GBFS
Receiver	L3Server

Data Field (i.e. contents of pcBuffer as defined in Appendix A)			
	name	type / length	defined values (description)
1	enable	1 byte	0 = false (stop logging) 1 = true (start logging)
2	mobileID	string	

#### 4.4.4 L3SRV\_IS\_LOGGING\_ENABLED

##### 4.4.4.1 Purpose

Retrieves the current log setting of the L3Server for a specific mobile.

##### 4.4.4.2 Pre-condition

n.a.

##### 4.4.4.3 Post-condition

Log setting has not been changed on the L3Server.

##### 4.4.4.4 Responses

- L3SRV\_LOGGING\_ENABLED

##### 4.4.4.5 Layout

<i>Message name</i>	L3SRV_IS_LOGGING_ENABLED
<i>Sender</i>	GBFS
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	mobileID	string	

#### 4.4.5 L3SRV\_LOG\_FILE

##### 4.4.5.1 Purpose

Tells GBFS that a log file (for logging of live data, not reload) has been opened or closed.

##### 4.4.5.2 Pre-condition

None.

##### 4.4.5.3 Post-condition

Filtersetting is unchanged.

##### 4.4.5.4 Responses

None.

##### 4.4.5.5 Layout

<i>Message name</i>	L3SRV_LOGFILE
<i>Sender</i>	L3Server
<i>Receiver</i>	GBFS

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	open	1 byte	0 = close file 1 = opened file
2	mobileID	string	

#### 4.4.6 L3SRV\_MOBILE\_FILTER

##### 4.4.6.1 Purpose

Returns the filter set in the server as it is set in the mobile or will be set in the mobile when a test case starts.

##### 4.4.6.2 Pre-condition

None.

##### 4.4.6.3 Post-condition

Filter setting is not changed.

##### 4.4.6.4 Responses

None.

##### 4.4.6.5 Layout

Message name	L3SRV_MOBILE_FILTER
Sender	L3Server
Receiver	GBFS

Data Field (i.e. contents of pcBuffer as defined in Appendix A)			
	name	type / length	defined values (description)
1	mobileID	string	
2	filterSettings	byte	logical or, filter set CC = 1 RR = 2 SS = 4 MM = 8 SMS = 16
3	opcodeFilter	dynamic array of 16-bytes entries	This field will not be used and no opcode mask will be sent to the mobile, until Condat defines the opcode mask. The mask can be hard-coded in LTS. For each bit set in the filtersetting, one set of 16 bytes should be placed in the opcodeFilterField, in the same order as the bits set in filterSettings, starting with the lowest bit.

#### 4.4.7 L3SRV\_LOGGING\_ENABLED

##### 4.4.7.1 Purpose

Contains the current log setting of the L3Server for a specific mobile.

##### 4.4.7.2 Pre-condition

n.a.

##### 4.4.7.3 Post-condition

n.a.

##### 4.4.7.4 Responses

None

#### 4.4.7.5 Layout

<i>Message name</i>	L3SRV_LOGGING_ENABLED
<i>Sender</i>	L3Server
<i>Receiver</i>	GBFS

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	mobileID	string	
2	enabled	byte	0 = false (logging disabled) 1 = true (logging enabled)

#### 4.4.8 TEXE\_STARTED\_TC

##### 4.4.8.1 Purpose

Announces the start of a new run of a specific testcase.

##### 4.4.8.2 Pre-condition

No testcase running

##### 4.4.8.3 Post-condition

Timestamp stored, logfile opened.

##### 4.4.8.4 Responses

None.

##### 4.4.8.5 Layout

<i>Message name</i>	TEXE_STARTED_TC
<i>Sender</i>	TEXE
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	testcaseName	string	

#### 4.4.9 TEXE\_RESULT

##### 4.4.9.1 Purpose

Announces the termination of a running testcase

##### 4.4.9.2 Pre-condition

Testcase running

##### 4.4.9.3 Post-condition

Logfile closed

##### 4.4.9.4 Responses

None.

#### 4.4.9.5 Layout

<i>Message name</i>	TEXE_RESULT
<i>Sender</i>	TEXE
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	result	short	TEXE_TC_FINISHED = 1, TEXE_TC_TERMINATED=2

Obsolete

## 5. Interface L3Server – L3Driver

### 5.1 Interface description

The L3Server – L3Driver interface handles the communication between the L3Server process and the L3Driver process on the gateway. The interface uses the VCMS message methods as described in [C-CMS] for exchanging messages.

The responsibilities of the L3Driver are:

- forwarding commands from the L3 Server towards the mobile, and
- forwarding primitives from the mobile towards the L3 Server.

### 5.2 Interface messages

The following message can be sent from L3Server to L3Driver:

- L3DRV\_SET\_MOBILE\_FILTER  
This message is sent to the L3Driver in order to set a filter inside the mobile

L3Driver can send the following message to the L3Server:

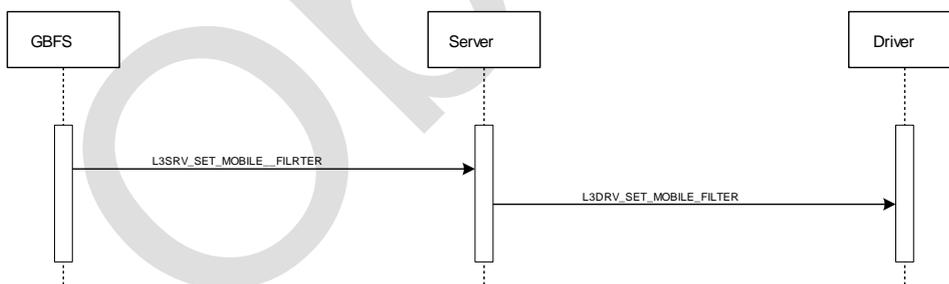
- L3DRV\_DATA  
This message contains primitives received from the mobile.
- L3DRV\_OK  
This message is returned upon a L3DRV\_SET\_MOBILE\_FILTER command that is confirmed by the mobile.
- L3DRV\_ERROR  
This message is returned upon a L3DRV\_SET\_MOBILE\_FILTER command that is not confirmed by the mobile.

### 5.3 Scenario's

The following scenario will be described:

- Set filters into a mobile

#### 5.3.1 Set filters into a mobile



## 5.4 Detailed description of the messages

### 5.4.1 L3DRV\_DATA

#### 5.4.1.1 Purpose

Contains the data (CC, RR, SS, MM, SMS primitives)

#### 5.4.1.2 Pre-condition

Duplicating of primitives within the mobile is enabled.

#### 5.4.1.3 Post-condition

N.a.

#### 5.4.1.4 Responses

None.

#### 5.4.1.5 Layout

<i>Message name</i>	L3DRV_DATA
<i>Sender</i>	L3Driver
<i>Receiver</i>	L3Server

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
	primitive	bytes	The complete message as defined in section 7.3.3, excluding the first and last byte (i.e. without STX and LF).

### 5.4.2 L3DRV\_SET\_MOBILE\_FILTER

#### 5.4.2.1 Purpose

Set a filter in the mobile, so that only a subset of primitives will be forwarded from the mobile to the L3Driver.

#### 5.4.2.2 Pre-condition

N.a.

#### 5.4.2.3 Post-condition

Only primitives specified in filter settings will be sent to the driver. Notice that this is different from L3SRV\_SET\_SERVER\_FILTER (see section 3.4.6).

#### 5.4.2.4 Responses

- L3DRV\_OK
- L3DRV\_ERROR

#### 5.4.2.5 Layout

<i>Message name</i>	L3DRV_SET_MOBILE_FILTER
<i>Sender</i>	L3Server
<i>Receiver</i>	L3Driver

<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>			
--	--	--	--

	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	filterSettings	byte	logical or, filter set CC = 1 RR = 2 SS = 4 MM = 8 SMS = 16
2	opcodeFilter	dynamic array of 16-bytes entries	This field will not be used and no opcode mask will be sent to the mobile, until Condat defines the opcode mask. The mask can be hard-coded in LTS. For each bit set in the filtersetting, one set of 16 bytes should be placed in the opcodeFilterField, in the same order as the bits set in filterSettings, starting with the lowest bit.

#### 5.4.3 L3DRV\_OK

This message is the same as L3SRV\_OK, see section 3.4.11. For clarity in the interface definition, the message has its own name.

#### 5.4.4 L3DRV\_ERROR

This message is the same as L3SRV\_ERROR, see section 3.4.12. For clarity in the interface definition, the message has its own name.

## 6. Interface LTS– L3Viewer

### 6.1 Interface description

LTS is responsible for starting L3Viewers. When a L3Viewer for a particular mobile is already running, LTS shall not start another one. Instead, LTS shall notify this L3Viewer, so it will bring itself to the front. LTS will detect whether or not a L3Viewer for a particular mobile is running by querying CMS for its processes. Therefore, the L3Viewer has to be registered under a fixed name. This name will be “L3Viewer#<mobileID>”. The mobileID will be passed to the L3Viewer as the command-line argument. The command line will be “L3Viewer <mobileID>.svc –mobname <mobileID>”

### 6.2 Interface messages

The following messages can be sent from LTS to L3Viewer:

- L3VWR\_TO\_FRONT  
Brings the window of the L3Viewer to the top, so it is visible.

The L3Viewer will respond with:

- L3VWR\_OK  
This confirms a successful handling of L3VWR\_TO\_FRONT

### 6.3 Detailed description of the messages

#### 6.3.1 L3VWR\_TO\_FRONT

##### 6.3.1.1 Purpose

Bring the L3Viewer to the front, so it is visible.

##### 6.3.1.2 Pre-condition

N.a.

##### 6.3.1.3 Post-condition

Viewer is visible and top window.

##### 6.3.1.4 Responses

L3SRV\_OK

##### 6.3.1.5 Layout

<i>Message name</i>	L3VWR_TO_FRONT
<i>Sender</i>	LTS-GBFS
<i>Receiver</i>	L3Viewer

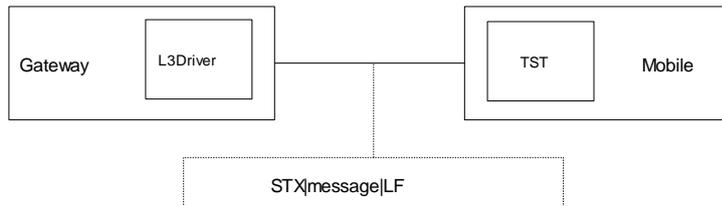
<i>Data Field (i.e. contents of pcBuffer as defined in Appendix A)</i>		
<n.a.>		

#### 6.3.2 L3VWR\_OK

This message is the same as L3SRV\_OK, see section 3.4.11. For clarity in the interface definition, the message has its own name.

## 7. Interface L3Driver – TST Mobile

The L3Driver – TST Mobile interface handles the communication between the L3Driver process in the gateway and the TST process in the mobile.



### 7.1 Interface characteristics

Serial port configuration, as defined in [C-PCO]:

- default baudrate: 38400
- default flow control: none
- default data bits: 8
- default stop bits: 1
- default parity: none

Data structure

- The message is preceded by an STX-byte (start of text, 0x02)
- The message is followed by an LF-byte (line feed, 0x0A)

### 7.2 Message structure

The message structure is described in the table below, as defined in [C-PCO]:

<i>Message structure</i>			
	<i>name</i>	<i>Length (bytes)</i>	<i>defined values (description)</i>
1	Id	1	"S", meaning config message sent to mobile "T", meaning trace information received from mobile "P", meaning primitive (possibly containing L3 air message info) received from mobile
2	Time	4	Time stamp in ms (binary code hexadecimal)
3	Size	4	Number of bytes following this field (binary code hexadecimal)
4	Sender	4	Entity name of sender
5	Receiver	4	Entity name of receiver

When Id = "S" or "T":

6	Data	Size-8	Data block, ASCII text
---	------	--------	------------------------

When Id = "P":

6	Opcode	4	Identifies the type of primitive
7	Data	Size-12	Data block, ASCII text

### 7.3 Interface messages

The following messages will be sent between L3Driver and TST, as defined in [C-TF]:

- TRACECLASS, disable trace information

- DUPLICATE, duplicating primitive messages to LTS
- L3 Primitives

### 7.3.1 Setting of trace filters

#### 7.3.1.1 Purpose

The TRACECLASS command is used to set the class of traces that are to be reported. Within LTS all traces will be disabled.

#### 7.3.1.2 Pre-condition

N.a.

#### 7.3.1.3 Post-condition

Trace filter set within the mobile. No trace information will be received from the mobile.

#### 7.3.1.4 Responses

TRACECLASS: OK (will be ignored)

#### 7.3.1.5 Layout

<i>name</i>	<i>defined values (description)</i>
Id	"S"
Sender	"LTS"
Receiver	one of [RR, SS, MM, CC, SMS]
Data	"TRACECLASS 00" (deactivate)

### 7.3.2 Duplicating messages to LTS

#### 7.3.2.1 Purpose

The DUPLICATE command is used to copy the messages of a specific SAP between two entities of the protocol stack to the LTS system.

#### 7.3.2.2 Pre-condition

n.a.

#### 7.3.2.3 Post-condition

The mobile sends (or does not send anymore) primitives to the LTS system.

#### 7.3.2.4 Responses

DUPLICATE ALL [CLEAR] LTS: OK (will be ignored)

#### 7.3.2.5 Layout

<i>name</i>	<i>defined values (description)</i>
Id	"S"
Sender	"LTS"
Receiver	one of [RR, SS, MM, CC, SMS]
Data	"DUPLICATE ALL [<opc mask>] LTS" (activate) "DUPLICATE ALL LTS CLEAR" (deactivate)

### 7.3.3 L3 Primitives

#### 7.3.3.1 Purpose

Contains the requested L3 primitives.

#### 7.3.3.2 Pre-condition

Duplicating is activated.

#### 7.3.3.3 Post-condition

N.a.

#### 7.3.3.4 Responses

None

#### 7.3.3.5 Layout

<i>name</i>	<i>defined values (description)</i>
Id	"P"
Receiver	"LTS"
Opcode	
Data	raw data

This total message will be encapsulated in the L3DRV\_DATA message as described in 5.4.1.

## Appendix A CMS Message Structure

Messages between processes are sent using the defined structure as listed in the table below. More information can be found in [C-GROBSP, section 4.3].

<i>Message structure</i>			
	<i>name</i>	<i>type / length</i>	<i>defined values (description)</i>
1	ubSize	U8	header size in bytes
2	ubReserved	U8	<reserved>
3	uwSize	U16	length of data fields in bytes
4	ulTime	U32	time stamp of messages, in sec. after 01-01-1970
5	uwTenthOfMS	U16	extension to time stamp, tenth of milliseconds
6	uwID	U16	message identification
7	Receiver	string	name of receiver (null terminating ASCII string)
8	Sender	string	name of sender (null terminating ASCII string)
9	DataField	byte / uwSize	data field (uwSize bytes)

## Appendix B Message numbers

<i>Message name</i>	<i>Message identification</i>
L3SRV_CONNECT	0x1E01
L3SRV_DISCONNECT	0x1E02
L3SRV_SUBSCRIBE	0x1E03
L3SRV_UNSUBSCRIBE	0x1E04
L3SRV_ENABLE_LOGGING	0x1E05
L3SRV_LOAD_LOG_FILE	0x1E06
L3SRV_COPY_LOG_FILE	0x1E07
L3SRV_SET_SERVER_FILTER	0x1E08
L3SRV_SET_MOBILE_FILTER	0x1E09
L3SRV_GET_MOBILE_FILTER	0x1E0A
L3SRV_EXIT	0x1E0B
L3SRV_LOG_FILE	0x1E0C
L3SRV_DATA	0x1E0D
L3SRV_FILTER	0x1E0E
L3SRV_CONNECTED	0x1E0F
L3SRV_OK	0x1E10
L3SRV_ERROR	0x1E11
L3SRV_IS_LOGGING_ENABLED	0x1E12
L3SRV_LOGGING_ENABLED	0x1E13

<i>Message name</i>	<i>Message identification</i>
TEXE_STARTED_TC	0x0307
TEXE_RESULT	0x0302

<i>Message name</i>	<i>Message identification</i>
L3DRV_SET_MOBILE_FILTER	0x2001
L3DRV_DATA	0x2002
L3DRV_OK	0x2003
L3DRV_ERROR	0x2004

<i>Message name</i>	<i>Message identification</i>
L3VWR_TO_FRONT	0x1F01