

# Link Layer Interface Specification



# Link Layer Interface Specification

---

Version 0.9.2 Edition 1

Updated 2008-10-31

Distributed with Package strx25-0.9.2.1

Copyright © 2008 OpenSS7 Corporation  
All Rights Reserved.

## Abstract

This document is a Specification containing technical details concerning the implementation of the Link Layer Interface for OpenSS7. It contains recommendations on software architecture as well as platform and system applicability of the Link Layer Interface. It provides abstraction of the LAPB Protocol (ISO/IEC 7776) and LLC2 Protocol (ISO/IEC 8802) service interface to these components as well as providing a basis for link layer control for other protocols.

**Brian Bidulock** <[bidulock@openss7.org](mailto:bidulock@openss7.org)> for  
**The OpenSS7 Project** <<http://www.openss7.org/>>

---

Copyright © 2001-2008 OpenSS7 Corporation

Copyright © 1997-2000 Brian F. G. Bidulock

All Rights Reserved.

## Published by:

OpenSS7 Corporation

1469 Jefferys Crescent

Edmonton, Alberta T6L 6T1

Canada

Unauthorized distribution or duplication is prohibited.

Permission to use, copy and distribute this documentation without modification, for any purpose and without fee or royalty is hereby granted, provided that both the above copyright notice and this permission notice appears in all copies and that the name of OpenSS7 Corporation not be used in advertising or publicity pertaining to distribution of this documentation or its contents without specific, written prior permission. OpenSS7 Corporation makes no representation about the suitability of this documentation for any purpose. It is provided “as is” without express or implied warranty.

## Notice:

**OPENSS7 CORPORATION DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS DOCUMENTATION INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE, OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ON ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. IN NO EVENT SHALL OPENSS7 CORPORATION BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH ANY USE OF THIS DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.**

OpenSS7 Corporation reserves the right to revise this software and documentation for any reason, including but not limited to, conformity with standards promulgated by various agencies, utilization of advances in the state of the technical arts, or the reflection of changes in the design of any techniques, or procedures embodied, described, or referred to herein. OpenSS7 Corporation is under no obligation to provide any feature listed herein.

## Short Contents

1	Introduction .....	3
2	Model of the Link Layer .....	5
3	LLI Services .....	7
4	LLI Message Primitives .....	9
5	LLI Input-Output Controls .....	11
6	LLI Header Files .....	27
7	LLI Drivers and Modules .....	29
8	LLI Utilities .....	31
9	LLI File Formats .....	33
A	LLI Compatibility and Porting .....	39
	Glossary of LLI Terms and Acronyms .....	41
	References .....	43
	Index .....	47



# Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>3</b>
<b>2</b>	<b>Model of the Link Layer</b> .....	<b>5</b>
<b>3</b>	<b>LLI Services</b> .....	<b>7</b>
3.1	LLI Commands .....	7
3.2	LLI Data Structures.....	7
<b>4</b>	<b>LLI Message Primitives</b> .....	<b>9</b>
<b>5</b>	<b>LLI Input-Output Controls</b> .....	<b>11</b>
5.1	Input-Output Control Data Structures .....	11
5.1.1	LLPLAIN - Plain Data Structures .....	11
5.1.1.1	ll_hdioc Structure .....	11
5.1.2	LLSNID - Subnetwork Identifier Data Structures .....	11
5.1.2.1	ll_snioc Structure.....	12
5.1.3	LLSTATUS - Status Data Structures.....	13
5.1.3.1	ll_stnioc Structure.....	13
5.1.4	LLSTATS - Statistics Data Structures.....	13
5.1.4.1	lapb_stioc Structure .....	13
5.1.4.2	lapbstats_t Structure .....	14
5.1.4.3	llc2_stioc Structure.....	15
5.1.4.4	llc2stats_t Structure .....	15
5.1.4.5	mlp_stioc Structure.....	17
5.1.4.6	mlpstats_t Structure.....	17
5.1.5	LLGSTATS - Global Statistics Data Structures.....	18
5.1.5.1	lapb_gstioc Structure .....	18
5.1.5.2	llc2_gstioc Structure.....	18
5.1.5.3	mlp_gstioc Structure .....	19
5.1.6	LLLAPBTUNE - LAPB Tuning Data Structures.....	19
5.1.6.1	lapb_tnioc Structure.....	19
5.1.6.2	lapbtune_t Structure .....	20
5.1.7	LLLLC2TUNE - LLC2 Tuning Data Structures .....	21
5.1.7.1	llc2_tnioc Structure.....	21
5.1.7.2	llc2tune_t Structure .....	21
5.1.8	LLMLPTUNE - MLP Tuning Data Structures.....	22
5.1.8.1	mlp_tnioc Structure .....	22
5.1.8.2	mlptune_t Structure.....	22
5.2	Input-Output Control Commands .....	23
5.2.1	SLP Input-Output Control Commands .....	23
5.2.1.1	L_SETSNID.....	23
5.2.1.2	L_GETSNID.....	23

5.2.1.3	L_SETTUNE	24
5.2.1.4	L_GETTUNE	24
5.2.1.5	L_GETSTATS	24
5.2.1.6	L_ZEROSTATS	24
5.2.1.7	L_TRACEON	24
5.2.1.8	L_TRACEOFF	24
5.2.1.9	L_GETGSTATS	24
5.2.1.10	L_ZEROGSTATS	24
5.2.1.11	L_LINKDISABLE	25
5.2.1.12	L_LINKENABLE	25
5.2.1.13	L_PUTX32MAP	25
5.2.1.14	L_GETX32MAP	25
5.2.2	MLP Input-Output Control Commands	25
5.2.2.1	M_SETSNID	25
5.2.2.2	M_GETSNID	25
5.2.2.3	M_SETTUNE	25
5.2.2.4	M_GETTUNE	25
5.2.2.5	M_GETSTATS	25
5.2.2.6	M_ZEROSTATS	25
5.2.2.7	M_TRACEON	26
5.2.2.8	M_TRACEOFF	26
5.2.2.9	M_GETGSTATS	26
5.2.2.10	M_ZEROGSTATS	26
<b>6</b>	<b>LLI Header Files</b>	<b>27</b>
6.1	ll_control.h Header File	27
6.2	ll_proto.h Header File	27
6.3	mlp_control.h Header File	27
6.4	mlp_proto.h Header File	27
<b>7</b>	<b>LLI Drivers and Modules</b>	<b>29</b>
7.1	hdlc Driver	29
7.2	lapb Driver	29
7.3	llc1 Driver	29
7.4	llc2 Driver	29
7.5	s_llic Module	30
7.6	s_lll Module	30
<b>8</b>	<b>LLI Utilities</b>	<b>31</b>
8.1	lltune Utility	31
8.2	mlptune Utility	31
8.3	linkadd Utility	31
8.4	linkdel Utility	31
8.5	linklist Utility	31
8.6	linkreset Utility	31
8.7	linkstart Utility	31
8.8	linkstate Utility	31
8.9	linkstop Utility	31



<b>9</b>	<b>LLI File Formats .....</b>	<b>33</b>
9.1	lapbtemplate File Format .....	33
9.2	llc2template File Format .....	36
9.3	mlptemplate File Format .....	38
<b>Appendix A</b>	<b>LLI Compatibility and Porting ..</b>	<b>39</b>
A.1	Compatibility with AIXlink/X.25 .....	39
A.2	Compatibility with HP X.25/9000 .....	39
A.3	Compatibility with IRIS SX.25 .....	39
A.4	Compatibility with PT X.25 .....	39
A.5	Compatibility with RadiSys WAN .....	39
A.6	Compatibility with SBE X.25 .....	39
A.7	Compatibility with Solstice X.25 .....	39
	<b>Glossary of LLI Terms and Acronyms .....</b>	<b>41</b>
	<b>References .....</b>	<b>43</b>
	<b>Index .....</b>	<b>47</b>



## List of Figures

## List of Tables

# 1 Introduction

*UNIX System V Release 3.2* included a Logical Link Interface (LLI) that was later standardized as the Data Link Provider Interface (DLPI) by the *Open Group*.<sup>1</sup>

The Link Layer Interface (LLI) was developed by Spider Systems, Ltd., (now a division of Emerson Power) and is widely available on many platforms. For example, *AIX AIXlink/X.25*, *HP-UX X.25/9000*, *IRIX IRIS SX.25*, *PT X.25*, *RadiSys WAN*, *SBE X.25*, *Solaris SunLink X.25* and *Solaris Solstice X.25*, implement the Link Layer Interface (LLI).<sup>2</sup>

The Link Layer Interface (LLI) as designed to be used directly with standard *STREAMS* system calls and does not require the use of a cooperating user space shared library. Applications programs directly use the `getmsg(2s)`, `getpmsg(2)`, `putmsg(2s)`, `putpmsg(2)` and `ioctl(2)` system calls.<sup>3</sup> Nevertheless, user shared object libraries can easily be constructed using this *STREAMS* service primitive interface.

The system header files that must be included when compiling user applications, or *STREAMS* drivers and modules that use the interface, are detailed in [Chapter 6 \[LLI Header Files\]](#), page 27.

A user library, 'libsx25', is provided, not for interfacing to the message primitive service interface, but for providing various helper functions when using the *STREAMS* service interface. This library is detailed in [\(undefined\) \[\(undefined\)\]](#), page (undefined).

---

<sup>1</sup> Evidence of this can still be found in the 'dlpi.h' header file distributed by *The Open Group* in the DLPI standard. Reference is made in the header file to backward compatibility to the LLI interface.

<sup>2</sup> But, for the most part, it is not called the *Link Layer Interface*, but is instead sunk inside the Network Layer Interface (NLI).

<sup>3</sup> See `getmsg(2s)`, `getpmsg(2)`, `putmsg(2s)`, `putpmsg(2)` and `ioctl(2)` manual pages.



## 2 Model of the Link Layer





## **3 LLI Services**

### **3.1 LLI Commands**

### **3.2 LLI Data Structures**



## 4 LLI Message Primitives

Although it perhaps did not at first, the LLI uses the service primitives and message format of the *Data Link Provider Interface*, [\[DLPI\]](#), page 43.



## 5 LLI Input-Output Controls

### 5.1 Input-Output Control Data Structures

The `‘/usr/include/strx25/sys/snet/ll_control.h’` header file (`‘<sys/snet/ll_control.h>’` with proper compile flags) defines a number of structures, pointers to which are used as arguments to input-output controls. These structures fall into four classes, identified by the value of the first byte of the structure, as follows:

LI_PLAIN	A <code>ll_hdioc</code> structure that identifies the subnetwork (link)
LI_SNID	A <code>ll_snioc</code> structure that identifies the subnetwork (link)
LI_STATUS	A <code>ll_stnioc</code> structure that identifies the subnetwork (link)
LI_STATS	A <code>ll_stioc</code> or <code>mlp_stioc</code> structure that identifies the subnetwork (link)
LI_GSTATS	A <code>ll_gstioc</code> or <code>mlp_gstioc</code> structure that identifies the subnetwork (link)
LI_LAPBTUNE	A <code>lapb_tnioc</code> structure that identifies the subnetwork (link)
LI_LLC2TUNE	A <code>llc2_tnioc</code> structure that identifies the subnetwork (link)
LI_MLPTUNE	A <code>mlp_tnioc</code> structure that identifies the subnetwork (link)

#### 5.1.1 LI\_PLAIN - Plain Data Structures

##### 5.1.1.1 ll\_hdioc Structure

The `ll_hdioc` structure is formatted as follows:

```
struct ll_hdioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
};
```

The `ll_hdioc` structure contains the following members:

`lli_type` Always LLI\_PLAIN.

`lli_spare` Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

#### 5.1.2 LI\_SNID - Subnetwork Identifier Data Structures

### 5.1.2.1 ll\_snioc Structure

The `ll_snioc` structure is used in the `L_SETSNID`, `L_GETSNID`, `M_SETSNID` and `M_GETSNID` input-output controls.

The `ll_snioc` structure is formatted as follows:

```

struct ll_snioc {
    uint8_t lli_type;
    uint8_t lli_class;
    uint8_t lli_spare[2];
    uint32_t lli_snid;
    uint32_t lli_index;
    uint32_t lli_slp_snid;
    uint16_t lli_slp_pri;
};

```

The `ll_snioc` structure contains the following members:

`lli_type` Always `LLI_SNID`.

`lli_class`

Specifies the class of the link being registered. This field can be one of the following values:

<code>LC_LLC1</code>	LLC Type 1 link.
<code>LC_LLC2</code>	LLC TYpe 2 link.
<code>LC_LAPBDTE</code>	LAPB DTE.
<code>LC_LAPBXDTE</code>	LAPB DTE with extended addressing.
<code>LC_LAPBDCE</code>	LAPB DCE.
<code>LC_LAPBXDCE</code>	LAPB DCE with extended addressing.
<code>LC_LAPDTE</code>	LAP DTE.
<code>LC_LAPDCE</code>	LAP DCE.
<code>LC_HDLC</code>	HDLC.
<code>LC_HDLCX</code>	HDLC with extended addressing.
<code>LC_MLAPBDTE</code>	LAPB DTE multilink.
<code>LC_MLAPBXDTE</code>	LAPB DTE multilink with extended addressing.
<code>LC_MLAPBDCE</code>	LAPB DCE multilink.
<code>LC_MLAPBXDCE</code>	LAPB DCE multilink with extended addressing.
<code>LC_MLP</code>	MLP link.

`lli_spare`

Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

`lli_index`

Specifies the lower multiplex identifier of the Stream linked under a multiplexing driver as returned from `I_LINK(7)`.<sup>1</sup>

<sup>1</sup> See the `I_LINK(7)` manual page.

`lli_slp_snid`  
Specifies the subnetwork (link) identifier for an SLP link within an MLP link.

`lli_slp_pri`  
Specifies the subnetwork (link) priority within an MLP link.

### 5.1.3 LLI\_STATUS - Status Data Structures

#### 5.1.3.1 ll\_stnioc Structure

The `ll_stnioc` structure is formatted as follows:

```
struct ll_stnioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
};
```

The `ll_stnioc` structure contains the following members:

`lli_type` Always `LLI_STATUS`.

`lli_spare`  
Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

### 5.1.4 LLI\_STATS - Statistics Data Structures

#### 5.1.4.1 lapb\_stioc Structure

The `lapb_stioc` structure is formatted as follows:

```
struct lapb_stioc {
    uint8_t lli_type;
    uint8_t state;
    uint16_t lli_spare;
    uint32_t lli_snid;
    lapbstats_t lli_stats;
};
```

The `lapb_stioc` structure contains the following members:

`lli_type` Always `LLI_STATS`.

`state`

`lli_spare`  
Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

`lli_stats`  
Contains the `lapbstats_t` structure described in [Section 5.1.4.2 \[lapbstats't Structure\], page 14](#).

### 5.1.4.2 lapbstats\_t Structure

The `lapbstats_t` structure is formatted as follows:

```
typedef struct lapb_stats {
    uint32_t lapbmonarray[49];
} lapbstats_t;
```

The `lapbstats_t` structure contains the following members:

#### lapbmonarray

Provides an array of 32-bit unsigned integers containing statistics. The indexes of the elements of the array are as follows:

0	tx_ign	-
1	rx_badlen	-
2	rx_unknown	-
3	t1_exp	-
4	t4_exp	-
5	t4_n2_exp	-
6	RR_rx_cmd	-
7	RR_rx_rsp	-
8	RR_tx_cmd	-
9	RR_tx_rsp	-
10	RR_tx_cmd_p	-
11	RNR_rx_cmd	-
12	RNR_rx_rsp	-
13	RNR_tx_cmd	-
14	RNR_tx_rsp	-
15	RNR_tx_cmd_p	-
16	REJ_rx_cmd	-
17	REJ_rx_rsp	-
18	REJ_tx_cmd	-
19	REJ_tx_rsp	-
20	REJ_tx_cmd_p	-
21	SABME_rx_cmd	-
22	SABME_tx_cmd	-
23	DISC_rx_cmd	-
24	DISC_tx_cmd	-
25	UA_rx_rsp	-
26	UA_tx_rsp	-
27	DM_rx_rsp	-
28	DM_tx_rsp	-
29	I_rx_cmd	-
30	I_tx_cmd	-
31	FRMR_rx_rsp	-
32	FRMR_tx_rsp	-
33	tx_rtr	-
34	rx_bad	-



35	rx_dud	-
36	rx_ign	-
37	XID_rx_cmd	-
38	XID_rx_rsp	-
39	XID_tx_cmd	-
40	XID_tx_rsp	-
41	TEST_rx_cmd	-
42	TEST_rx_rsp	-
43	TEST_tx_cmd	-
44	TEST_tx_rsp	-
45	SABM_rx_cmd	-
46	SABM_tx_cmd	-
47	SARM_rx_cmd	-
48	SARM_tx_cmd	-

### 5.1.4.3 llc2\_stioc Structure

The llc2\_stioc structure is formatted as follows:

```
struct llc2_stioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
    llc2stats_t lli_stats;
};
```

The llc2\_stioc structure contains the following members:

**lli\_type** Always LLI\_STATS.

**lli\_spare** Spare bytes for alignment: set to zero by issuer and ignored by responder.

**lli\_snid** Specifies the subnetwork identifier (link identifier) for the operation.

**lli\_stats** Contains the llc2stats\_t structure containing the statistics described in [Section 5.1.4.4 \[llc2stats\\_t Structure\]](#), page 15.

### 5.1.4.4 llc2stats\_t Structure

The llc2stats\_t structure is formatted as follows:

```
typedef struct llc2_stats {
    uint32_t llc2monarray[49];
} llc2stats_t;
```

The llc2stats\_t structure contains the following members:

**llc2monarray** Provides an array of 32-bit unsigned integers containing the statistics. The indices of the individual elements are as follows:

0	tx_ign	-
---	--------	---

## Chapter 5: LLI Input-Output Controls

1	rx_badlen	-
2	rx_unknown	-
3	t1_exp	-
4	t4_exp	-
5	t4_n2_exp	-
6	RR_rx_cmd	-
7	RR_rx_rsp	-
8	RR_tx_cmd	-
9	RR_tx_rsp	-
10	RR_tx_cmd_p	-
11	RNR_rx_cmd	-
12	RNR_rx_rsp	-
13	RNR_tx_cmd	-
14	RNR_tx_rsp	-
15	RNR_tx_cmd_p	-
16	REJ_rx_cmd	-
17	REJ_rx_rsp	-
18	REJ_tx_cmd	-
19	REJ_tx_rsp	-
20	REJ_tx_cmd_p	-
21	SABME_rx_cmd	-
22	SABME_tx_cmd	-
23	DISC_rx_cmd	-
24	DISC_tx_cmd	-
25	UA_rx_rsp	-
26	UA_tx_rsp	-
27	DM_rx_rsp	-
28	DM_tx_rsp	-
29	I_rx_cmd	-
30	I_tx_cmd	-
31	FRMR_rx_rsp	-
32	FRMR_tx_rsp	-
33	tx_rtr	-
34	rx_bad	-
35	rx_dud	-
36	rx_ign	-
37	XID_rx_cmd	-
38	XID_rx_rsp	-
39	XID_tx_cmd	-
40	XID_tx_rsp	-
41	TEST_rx_cmd	-
42	TEST_rx_rsp	-
43	TEST_tx_cmd	-
44	TEST_tx_rsp	-
45	I_rx_rsp	-

```

46     I_tx_rsp           -
47     UI_rx_cmd         -
48     UI_tx_cmd         -

```

#### 5.1.4.5 mlp\_stioc Structure

The `mlp_stioc` structure is formatted as follows:

```

struct mlp_stioc {
    uint8_t lli_type;
    uint8_t state;
    uint16_t lli_spare;
    uint32_t lli_snid;
    mlpstats_t lli_stats;
};

```

The `mlp_stioc` structure contains the following members:

`lli_type` Always `LLI_STATS`.

`lli_state`  
Provides the state of the link.

`lli_spare`  
Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

`lli_stats`  
Contains the `mlpstats_t` structure containing the statistics described in [Section 5.1.4.6 \[mlpstats't Structure\], page 17](#).

#### 5.1.4.6 mlpstats\_t Structure

The `mlpstats_t` structure is formatted as follows:

```

typedef struct mlp_stats {
    uint32_t mlpmonarray[12];
} mlpstats_t;

```

The `mlpstats_t` structure contains the following members:

`mlpmonarray`  
Contains an array of 32-bit unsigned integers containing the statistics. The indices of the individual elements of the array are as follows:

```

0     MLP_frames_tx     -
1     MLP_frames_rx     -
2     MLP_reset_tx      -
3     MLP_reset_rx      -
4     MLP_confs_tx      -
5     MLP_confs_rx      -
6     MLP_slps          -
7     MLP_num_slps      -

```

8	MLP_mt1_exp	-
9	MLP_mt2_exp	-
10	MLP_mt3_exp	-
11	MLP_mn1_exp	-

## 5.1.5 LLI\_GSTATS - Global Statistics Data Structures

### 5.1.5.1 lapb\_gstioc Structure

The `lapb_gstioc` structure is formatted as follows:

```
struct lapb_gstioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lapbgstats[6];
};
```

The `lapb_gstioc` structure contains the following members:

`lli_type` Always `LLI_GSTATS`.

`lli_spare`  
Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lapbgstats`  
Contains an array of 32-bit unsigned integers containing the statistics. The indices of the individual elements of the array are as follows:

0	frames_tx	-
1	frames_rx	-
2	sabm_tx	-
3	sabm_rx	-
4	bytes_tx	-
5	bytes_rx	-

### 5.1.5.2 llc2\_gstioc Structure

The `llc2_gstioc` structure is formatted as follows:

```
struct llc2_gstioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t llc2gstats[6];
};
```

The `llc2_gstioc` structure contains the following members:

`lli_type` Always `LLI_GSTATS`.

`lli_spare`  
Spare bytes for alignment: set to zero by issuer and ignored by responder.

`llc2gstats`  
Contains an array of 32-bit unsigned integers containing the statistics. The indices of the individual elements of the array are as follows:

0	frames_tx	-
1	frames_rx	-
2	sabme_tx	-
3	sabme_rx	-
4	bytes_tx	-
5	bytes_rx	-

### 5.1.5.3 mlp\_gstioc Structure

The `mlp_gstioc` structure is formatted as follows:

```
struct mlp_gstioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t mlpgstats[10];
};
```

The `mlp_gstioc` structure contains the following members:

`lli_type` Always `LLI_GSTATS`.

`lli_spare`

Spare bytes for alignment: set to zero by issuer and ignored by responder.

`mlpgstats`

Contains an array of 32-bit unsigned integers containing the statistics. The indices of the individual elements of the array are as follows:

0	MLP_frames_tx	-
1	MLP_frames_rx	-
2	MLP_reset_tx	-
3	MLP_reset_rx	-
4	MLP_confs_tx	-
5	MLP_confs_rx	-
6	MLP_slps	-
7	MLP_num_slps	-
8	MLP_bytes_tx	-
9	MLP_bytes_rx	-

## 5.1.6 LI\_LAPBTUNE - LAPB Tuning Data Structures

### 5.1.6.1 lapb\_tnioc Structure

The `lapb_tnioc` structure is formatted as follows:

```
struct lapb_tnioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
    lapbtune_t lapb_tune;
};
```

The `lapb_tnioc` structure contains the following members:

- `lli_type` Always LLI\_LAPBTUNE.
- `lli_spare` Spare bytes for alignment: set to zero by issuer and ignored by responder.
- `lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.
- `lapb_tune` Contains the `lapbtune_t` tuning structure containing tuning parameters described in [Section 5.1.6.2 \[lapbtune\\_t Structure\]](#), page 20.

### 5.1.6.2 `lapbtune_t` Structure

The `lapbtune_t` structure is formatted as follows:

```
typedef struct lapbtune {
    uint16_t N2;
    uint16_t T1;
    uint16_t Tpf;
    uint16_t Trej;
    uint16_t Tbusy;
    uint16_t Tidle;
    uint16_t ack_delay;
    uint16_t notack_max;
    uint16_t tx_window;
    uint16_t tx_probe;
    uint16_t max_I_len;
    uint16_t llconform;
    uint16_t sabm_in_x32;
} lapbtune_t;
```

The `lapbtune_t` structure contains the following members:

`N2`  
`T1`  
`Tpf`  
`Trej`  
`Tbusy`  
`Tidle`  
`ack_delay`  
`notack_max`  
`tx_window`  
`tx_probe`  
`max_I_len`  
`llconform`  
`sabm_in_x32`

## 5.1.7 LI\_LLC2TUNE - LLC2 Tuning Data Structures

### 5.1.7.1 llc2\_tnioc Structure

The llc2\_tnioc structure is formatted as follows:

```
struct llc2_tnioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
    llc2tune_t llc2_tune;
};
```

The llc2\_tnioc structure contains the following members:

lli\_type Always LLI\_LLC2TUNE.

lli\_spare Spare bytes for alignment: set to zero by issuer and ignored by responder.

lli\_snid Specifies the subnetwork identifier (link identifier) for the operation.

llc2\_tune Contains the llc2tune\_t tuning structure containing tuning parameters described in [Section 5.1.7.2 \[llc2tune\\_t Structure\]](#), page 21.

### 5.1.7.2 llc2tune\_t Structure

The llc2tune\_t structure is formatted as follows:

```
typedef struct llc2tune {
    uint16_t N2;
    uint16_t T1;
    uint16_t Tpf;
    uint16_t Trej;
    uint16_t Tbusy;
    uint16_t Tidle;
    uint16_t ack_delay;
    uint16_t notack_max;
    uint16_t tx_window;
    uint16_t tx_probe;
    uint16_t max_I_len;
    uint16_t xid_window;
    uint16_t xid_Ndup;
    uint16_t xid_Tdup;
} llc2tune_t;
```

The llc2tune\_t structure contains the following members:

N2

T1

Tpf

Trej  
Tbusy  
Tidle  
ack\_delay  
notack\_max  
tx\_window  
tx\_probe  
max\_I\_len  
xid\_window  
xid\_Ndup  
xid\_Tdup

## 5.1.8 LI\_MLPTUNE - MLP Tuning Data Structures

### 5.1.8.1 mlp\_tnioc Structure

The `mlp_tnioc` structure is formatted as follows:

```
struct mlp_tnioc {
    uint8_t lli_type;
    uint8_t lli_spare[3];
    uint32_t lli_snid;
    mlptune_t mlp_tune;
};
```

The `mlp_tnioc` structure contains the following members:

`lli_type` Always LLI\_MLPTUNE.

`lli_spare` Spare bytes for alignment: set to zero by issuer and ignored by responder.

`lli_snid` Specifies the subnetwork identifier (link identifier) for the operation.

`mlp_tune` Contains the `mlptune_t` tuning structure containing tuning parameters described in [Section 5.1.8.2 \[mlptune't Structure\]](#), page 22.

### 5.1.8.2 mlptune\_t Structure

The `mlptune_t` structure is formatted as follows:

```
typedef struct mlptune {
    uint16_t mw;
    uint16_t mx;
    uint16_t mt1;
    uint16_t mt2;
    uint16_t mt3;
    uint16_t mn1;
} mlptune_t;
```

The `mlptune_t` structure contains the following members:



<code>mw</code>	The size of the MLP window.
<code>mx</code>	The size of the MLP guard region.
<code>mt1</code>	The time interval to wait for $MN(S) == MV(R)$ in deciseconds.
<code>mt2</code>	The time interval to wait for unblock in deciseconds.
<code>mt3</code>	The time interval to wait for reset confirmation.
<code>mn1</code>	The number of SLP transmission retries.

## 5.2 Input-Output Control Commands

<code>L_SETSNID</code>	Set subnetwork identifier.
<code>L_GETSNID</code>	Get subnetwork identifier.
<code>L_SETTUNE</code>	Set common tuning parameters.
<code>L_GETTUNE</code>	Get common tuning parameters.
<code>L_GETSTATS</code>	Get subnetwork statistics.
<code>L_ZEROSTATS</code>	Zero subnetwork statistics.
<code>L_TRACEON</code>	Turn message tracing on.
<code>L_TRACEOFF</code>	Turn message tracing off.
<code>L_GETGSTATS</code>	Get global statistics.
<code>L_ZEROGSTATS</code>	Zero global statistics.
<code>L_LINKDISABLE</code>	Disable link (subnetwork identifier).
<code>L_LINKENABLE</code>	Enable link (subnetwork identifier).
<code>L_PUTX32MAP</code>	Put X.32 table mapping.
<code>L_GETX32MAP</code>	Get X.32 table mapping.
<code>M_SETSNID</code>	Set subnetwork identifier.
<code>M_GETSNID</code>	Get subnetwork identifier.
<code>M_SETTUNE</code>	Set common tuning parameters.
<code>M_GETTUNE</code>	Get common tuning parameters.
<code>M_GETSTATS</code>	Get subnetwork statistics.
<code>M_ZEROSTATS</code>	Zero subnetwork statistics.
<code>M_TRACEON</code>	Turn message tracing on.
<code>M_TRACEOFF</code>	Turn message tracing off.
<code>M_GETGSTATS</code>	Get global statistics.
<code>M_ZEROGSTATS</code>	Zero global statistics.

### 5.2.1 SLP Input-Output Control Commands

#### 5.2.1.1 L\_SETSNID

The argument of this input-output control is a pointer to the `ll_snioc` structure described in [Section 5.1.2.1 \[ll'snioc Structure\]](#), page 12.

#### 5.2.1.2 L\_GETSNID

The argument of this input-output control is a pointer to the `ll_snioc` structure described in [Section 5.1.2.1 \[ll'snioc Structure\]](#), page 12.

### 5.2.1.3 L\_SETTUNE

The argument of this input-output control is a pointer to the `lapb_tnioc`, `llc2_tnioc` or `mlp_tnioc` structure.<sup>2</sup>

### 5.2.1.4 L\_GETTUNE

The argument of this input-output control is a pointer to the `lapb_tnioc`, `llc2_tnioc` or `mlp_tnioc` structure.<sup>3</sup>

### 5.2.1.5 L\_GETSTATS

The argument of this input-output control is a pointer to the `lapb_stioc`, `llc2_stioc` or `mlp_stioc` structure.<sup>4</sup>

### 5.2.1.6 L\_ZEROSTATS

The argument of this input-output control is a pointer to the `lapb_stioc`, `llc2_stioc` or `mlp_stioc` structure.<sup>5</sup>

### 5.2.1.7 L\_TRACEON

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.1.8 L\_TRACEOFF

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.1.9 L\_GETGSTATS

The argument of this input-output control is a pointer to the `lapb_gstioc`, `llc2_gstioc` or `mlp_gstioc` structure.<sup>6</sup>

### 5.2.1.10 L\_ZEROGSTATS

The argument of this input-output control is a pointer to the `lapb_gstioc`, `llc2_gstioc` or `mlp_gstioc` structure.<sup>7</sup>

---

<sup>2</sup> These structures are described in [Section 5.1.6.1 \[lapb'tnioc Structure\]](#), page 19, [Section 5.1.7.1 \[llc2'tnioc Structure\]](#), page 21, and [Section 5.1.8.1 \[mlp'tnioc Structure\]](#), page 22.

<sup>3</sup> These structures are described in [Section 5.1.6.1 \[lapb'tnioc Structure\]](#), page 19, [Section 5.1.7.1 \[llc2'tnioc Structure\]](#), page 21, and [Section 5.1.8.1 \[mlp'tnioc Structure\]](#), page 22.

<sup>4</sup> These structures are described in [Section 5.1.4.1 \[lapb'stioc Structure\]](#), page 13, [Section 5.1.4.3 \[llc2'stioc Structure\]](#), page 15, and [Section 5.1.4.5 \[mlp'stioc Structure\]](#), page 17.

<sup>5</sup> These structures are described in [Section 5.1.4.1 \[lapb'stioc Structure\]](#), page 13, [Section 5.1.4.3 \[llc2'stioc Structure\]](#), page 15, and [Section 5.1.4.5 \[mlp'stioc Structure\]](#), page 17.

<sup>6</sup> These structures are described in [Section 5.1.5.1 \[lapb'gstioc Structure\]](#), page 18, [Section 5.1.5.2 \[llc2'gstioc Structure\]](#), page 18, and [Section 5.1.5.3 \[mlp'gstioc Structure\]](#), page 19.

<sup>7</sup> These structures are described in [Section 5.1.5.1 \[lapb'gstioc Structure\]](#), page 18, [Section 5.1.5.2 \[llc2'gstioc Structure\]](#), page 18, and [Section 5.1.5.3 \[mlp'gstioc Structure\]](#), page 19.

### 5.2.1.11 L\_LINKDISABLE

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.1.12 L\_LINKENABLE

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.1.13 L\_PUTX32MAP

### 5.2.1.14 L\_GETX32MAP

## 5.2.2 MLP Input-Output Control Commands

### 5.2.2.1 M\_SETSNID

The argument of this input-output control is a pointer to the `ll_snioc` structure described in [Section 5.1.2.1 \[ll'snioc Structure\]](#), page 12.

### 5.2.2.2 M\_GETSNID

The argument of this input-output control is a pointer to the `ll_snioc` structure described in [Section 5.1.2.1 \[ll'snioc Structure\]](#), page 12.

### 5.2.2.3 M\_SETTUNE

The argument of this input-output control is a pointer to the `lapb_tnioc`, `llc2_tnioc` or `mlp_tnioc` structure.<sup>8</sup>

### 5.2.2.4 M\_GETTUNE

The argument of this input-output control is a pointer to the `lapb_tnioc`, `llc2_tnioc` or `mlp_tnioc` structure.<sup>9</sup>

### 5.2.2.5 M\_GETSTATS

The argument of this input-output control is a pointer to the `lapb_stioc`, `llc2_stioc` or `mlp_stioc` structure.<sup>10</sup>

### 5.2.2.6 M\_ZEROSTATS

The argument of this input-output control is a pointer to the `lapb_stioc`, `llc2_stioc` or `mlp_stioc` structure.<sup>11</sup>

---

<sup>8</sup> These structures are described in [Section 5.1.6.1 \[lapb'tnioc Structure\]](#), page 19, [Section 5.1.7.1 \[llc2'tnioc Structure\]](#), page 21, and [Section 5.1.8.1 \[mlp'tnioc Structure\]](#), page 22.

<sup>9</sup> These structures are described in [Section 5.1.6.1 \[lapb'tnioc Structure\]](#), page 19, [Section 5.1.7.1 \[llc2'tnioc Structure\]](#), page 21, and [Section 5.1.8.1 \[mlp'tnioc Structure\]](#), page 22.

<sup>10</sup> These structures are described in [Section 5.1.4.1 \[lapb'stioc Structure\]](#), page 13, [Section 5.1.4.3 \[llc2'stioc Structure\]](#), page 15, and [Section 5.1.4.5 \[mlp'stioc Structure\]](#), page 17.

<sup>11</sup> These structures are described in [Section 5.1.4.1 \[lapb'stioc Structure\]](#), page 13, [Section 5.1.4.3 \[llc2'stioc Structure\]](#), page 15, and [Section 5.1.4.5 \[mlp'stioc Structure\]](#), page 17.

### 5.2.2.7 M\_TRACEON

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.2.8 M\_TRACEOFF

The argument of this input-output control is a pointer to the `ll_hdioc` structure described in [Section 5.1.1.1 \[ll'hdioc Structure\]](#), page 11.

### 5.2.2.9 M\_GETGSTATS

The argument of this input-output control is a pointer to the `lapb_gstioc`, `llc2_gstioc` or `mlp_gstioc` structure.<sup>12</sup>

### 5.2.2.10 M\_ZEROGSTATS

The argument of this input-output control is a pointer to the `lapb_gstioc`, `llc2_gstioc` or `mlp_gstioc` structure.<sup>13</sup>

---

<sup>12</sup> These structures are described in [Section 5.1.5.1 \[lapb'gstioc Structure\]](#), page 18, [Section 5.1.5.2 \[llc2'gstioc Structure\]](#), page 18, and [Section 5.1.5.3 \[mlp'gstioc Structure\]](#), page 19.

<sup>13</sup> These structures are described in [Section 5.1.5.1 \[lapb'gstioc Structure\]](#), page 18, [Section 5.1.5.2 \[llc2'gstioc Structure\]](#), page 18, and [Section 5.1.5.3 \[mlp'gstioc Structure\]](#), page 19.

## **6 LLI Header Files**

### **6.1 ll\_control.h Header File**

### **6.2 ll\_proto.h Header File**

### **6.3 mlp\_control.h Header File**

### **6.4 mlp\_proto.h Header File**



## 7 LLI Drivers and Modules

### 7.1 hdlc Driver

The ‘hdlc’ driver is a pseudo-device *STREAMS* driver that provides raw HDLC framing as specified in *ISO/IEC 3309* described in reference [ISO3309], page 43.

The ‘hdlc’ driver provides a *Data Link Provider Interface* as its upper multiplex service interface, as described in [DLPI], page 43.

The ‘hdlc’ driver is documented in the `hdlc(4)` manual page.

### 7.2 lapb Driver

The ‘lapb’ driver is a pseudo-device *STREAMS* driver that provides the X.25 compatible LAPB procedures as specified in *ISO/IEC 7776* described in reference [ISO7776], page 43. This driver, in combination with the ‘x25’ multiplexing driver, provides X.25 over LAPB as described in reference [ISO8208], page 43.

The ‘lapb’ driver provides a *Data Link Provider Interface* as its upper multiplex service interface, as described in [DLPI], page 43.

The ‘lapb’ driver is documented in the `lapb(4)` manual page.

### 7.3 llc1 Driver

The ‘llc1’ driver is a pseudo-device *STREAMS* driver that provides the IEEE 802.2 LLC Type 1 (LLC1) procedures as specified in *ISO/IEC 8802-2* described in reference [ISO8802-2], page 44. This driver, in combination with the ‘x25’ multiplexing driver, provides X.25 over LLC1 as described in reference [ISO8881], page 44.

The ‘llc1’ driver provides a *Data Link Provider Interface* as its upper multiplex service interface, as described in [DLPI], page 43.

The ‘llc1’ driver is documented in the `llc1(4)` manual page.

### 7.4 llc2 Driver

The ‘llc2’ driver is a pseudo-device *STREAMS* driver that provides the IEEE 802.2 LLC Type 2 (LLC2) procedures as specified in *ISO/IEC 8802-2* described in reference [ISO8802-2], page 44. This driver, in combination with the ‘x25’ multiplexing driver, provides X.25 over LLC2 as described in reference [ISO8881], page 44.

The ‘llc2’ driver provides a *Data Link Provider Interface* as its upper multiplex service interface, as described in [DLPI], page 43.

The ‘llc2’ driver is documented in the `llc2(4)` manual page.

## 7.5 s\_llic Module

The ‘s\_llic’ module is a pushable *STREAMS* module that converts between the LLI input-output controls described in this manual and the DLPI input-output controls used by the [OpenSS7 Project](#). This particular module does not convert primitives, as both the LLI and the DLPI use the primitives of the *Data Link Provider Interface, Revision 2.0.0* described in reference [\[DLPI\]](#), page 43.

The ‘s\_llic’ module provides a *Data Link Provider Interface* as its upper and lower service interface, as described in [\[DLPI\]](#), page 43.

The ‘s\_llic’ module is documented in the [s\\_llic\(4\)](#) manual page.

## 7.6 s\_llli Module

The ‘s\_llli’ module is a pushable *STREAMS* module that converts between the LLI input-output controls described in this manual and the DLPI input-output controls used by the [OpenSS7 Project](#). This particular module converts between the LLI primitives described in this manual and the primitives of the *Data Link Provider Interface*, described in reference [\[DLPI\]](#), page 43.

The ‘s\_llli’ module provides a *Data Link Provider Interface* as its upper and lower service interface, as described in [\[DLPI\]](#), page 43.

The ‘s\_llli’ module is documented in the [s\\_llli\(4\)](#) manual page.



## 8 LLI Utilities

### 8.1 lltune Utility

The `lltune` utility is documented in the `lltune(8)` manual page.

### 8.2 mlptune Utility

The `mlptune` utility is documented in the `mlptune(8)` manual page.

### 8.3 linkadd Utility

The `linkadd` utility is documented in the `linkadd(8)` manual page.

### 8.4 linkdel Utility

The `linkdel` utility is documented in the `linkdel(8)` manual page.

### 8.5 linklist Utility

The `linklist` utility is documented in the `linklist(8)` manual page.

### 8.6 linkreset Utility

The `linkreset` utility is documented in the `linkreset(8)` manual page.

### 8.7 linkstart Utility

The `linkstart` utility is documented in the `linkstart(8)` manual page.

### 8.8 linkstate Utility

The `linkstate` utility is documented in the `linkstate(8)` manual page.

### 8.9 linkstop Utility

The `linkstop` utility is documented in the `linkstop(8)` manual page.



## 9 LLI File Formats

### 9.1 lapbtemplate File Format

The ‘lapbtemplate’ file format is documented in the `lapbtemplate(5)` manual page.

#### Name

‘lapbtemplate’ — Link Access Protocol (Balanced) File Format

#### Description

The ‘lapbtemplate’ describes the file format for input to the `lltune(8)` command for LAPB class subnetworks. The file format consists of a number of parameter values, one per line, formatted as described below. Each parameter value is described using its line number in the file, a parameter name, and a description of the format of the value. Only the value appears in the file, each value on a line by itself, one value per line.

Each of the LAPB configuration parameters corresponds to the member and values of the `lapb_tune` structure, that is carried in a `lapb_tnioc` structure by the `L_LAPBTUNE` input-output control.

These protocol parameters, and the default values that exist when tuning has not been applied to a newly created LAPB subnetwork, correspond directly to the protocol parameters and defaults in *ISO/IEC 7776*, *ITU-T Rec. X.25* and *X.75*.

#### Format

The LAPB template consists of 16 to 18 lines containing the following configuration information:

1. `N2_VAL` is the maximum number of times that a protocol data unit (PDU) is set following the expiry of the acknowledgement timer, the P-bit timer, or the reject timer. It also limits the number of times an RR with the P-bit set is sent when remote busy is true and the busy timer expires.
2. `T1_VAL` is the time during which the LAPB expects to receive an acknowledgement to an outstanding I-PDU or an expected response to a sent UI-PDU. The value is in units of 0.1 seconds (deciseconds).
3. `TPF_VAL` is the time during which the LAPB expects to receive a PDU with the F-bit set to 1 in response to a command with the P-bit set to 1. The value should be less than the acknowledgement timer. The value is in units of 0.1 seconds (deciseconds).
4. `TREJ_VAL` is the time interval during which the LAPB expects to receive a reply to a sent REJ DPU. The value is in units of 0.1 seconds (deciseconds).
5. `TBUSY_VAL` is the time interval during which the LAPB waits for an indication of the clearance of a busy condition at the other LAPB. The value is in units of 0.1 seconds (deciseconds).
6. `IDLE_VAL` is the time interval during which the LAPB expects to receive a PDU from the other LAPB. If it expires then the P/F cycle is initiated which may result in link disconnection. The value is in units of 0.1 seconds (deciseconds).

7. `ACK_DELAY` is the maximum delay in 0.1 second units before transmitting a delayed RR. This must be considerably less than the acknowledgement timer value, `T1_VAL`.
8. `NOTACK_MAX` is the maximum number of unacknowledged receive I PDUs before the RR acknowledging them all must be sent.
9. `LOC_WIND` is the number of unacknowledged I PDUs that may be sent.
10. `LOC_PROBE` is the position before the window is closed at which an I PDU is sent with the P-bit set to solicit an acknowledgement from the receiver.
11. `MAX_I_LEN` is the maximum size of a LAPB I-frame. LAPB requires all incoming I-frames above a certain size to be rejected by a FRMR. This parameter specifies the maximum size. It is constructed as the sum of the maximum X.25 data size, the X.25 protocol length and the LAPB protocol length.
12. `IGN_UA_ERROR` defines whether or not to ignore any UA frames received, when the connection is in `ERROR` state. The value is '1' for *true* and '0' for *false*. The default value is *false*.
13. `FRMR_FRMR_ERROR` defines whether or not to re-transmit a frame reject if a frame reject is received, when the connection is in `ERROR` state. The value is '1' for *true* and '0' for *false*. The default value is *false*.
14. `FRMR_INVRSP_ERROR` defines whether or not to transmit a frame reject if an invalid frame response is received, when the connection is in `ERROR` state. The value is '1' for *true* and '0' for *false*. The default value is *false*.
15. `SFRAME_PBIT` defines whether or not to send a frame reject if an S-frame is received without the P-bit set. The value is '1' for *true* and '0' for *false*. The default value is *false*.
16. `NO_DM_ADM` defines whether or not to send a DM on entry to `ADM` state after an N2 count expiry. The value is '1' for *true* and '0' for *false*. The default value is *false*.

The following two fields are optional extensions:

17. `IGN_DM_ERROR` defines whether or not to ignore DM frames received, when the connection is in `ERROR` state. The value is '1' for *true* and '0' for *false*. The default value is *false*.
18. `SABM_IN_X32` defines the action to take when a SABM is received in X.32 setup. The value is '1' for *true* and '0' for *false*. The default value is *false*.

The last two fields ('17' and '18') are enhancements.

## Files

Files following this format are normally kept in the `/etc/sysconfig/strx25/template/` directory.<sup>1</sup>

---

<sup>1</sup> Note that this directory varies depending on whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.

## See Also

- [lltune\(8\)](#)
- [lapb\(4\)](#)
- [x25netd\(8\)](#)

## Compatibility

The ‘lapbtemplate’ file format is compatible with *Spider X.25*, and implementations based on *Spider X.25*, such as *AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *Solstice X.25*, *PT X.25*, *SBE X.25*, with the following compatibility considerations:

- Most implementations only define the first 16 lines. This implementation defines 18 lines, where the first 16 lines are compatible with other implementations and the last additional two lines are optional.
- *PT X.25* documents the `SABM_IN_X32` LAPB template field but not the `IGN_DM_ERROR` LAPB template field. *Solstice X.25* and *IRIS SX.25* do not document either the `IGN_DM_ERROR` nor `SABM_IN_X32` LAPB template fields.

For additional compatibility information see, [lapb\(4\)](#), and [STREAMS\(9\)](#).

## Conformance

*AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *Solstice X.25*, *PT X.25*, *SBE X.25*, documentation.

## History

The ‘lapbtemplate’ file format first appeared in *Spider X.25*.

## 9.2 llc2template File Format

The ‘llc2template’ file format is documented in the `llc2template(5)` manual page.

### Name

‘llc2template’ — Logical Link Control Type 2 File Format

### Description

The ‘llc2template’ describes the file format for input to the `lltune(8)` command for LLC2 class subnetworks. The file format consists of a number of parameter values, one per line, formatted as described below. Each parameter value is described using its line number in the file, a parameter name, and a description of the format of the value. Only the value appears in the file, each value on a line by itself, one value per line.

Each of the LLC2 configuration parameters corresponds to the member and values of the `llc2_tune` structure, that is carried in a `llc2_tnioc` structure by the `L_LLC2TUNE` input-output control.

These protocol parameters, and the default values that exist when tuning has not been applied to a newly created LLC2 subnetwork, correspond directly to the protocol parameters and defaults in *ISO/IEC 8802-2:1998*.

### Format

The LLC2 template consists of 14 lines containing the following configuration information.

1. `N2_VAL` is the maximum number of times that a Protocol Data Unit (PDU) is sent following the expiry of the acknowledgement timer, the P-bit timer, or the reject timer. This parameter also limits the number of times an RR is sent with the P-bit set when remote busy is true and the busy timer expires.
2. `T1_VAL` is the time interval during which the LLC2 expects to receive an acknowledgement to an outstanding I-PDU or an expected response to a sent UI-PDU. The value is in units of 0.1 seconds.
3. `TPF_VAL` is the time during which the LLC2 expects to receive a PDU with the F-bit set to 1 in response to a command with the P-bit set to 1. The value should be less than that specified for the acknowledgement timer. The value is in units of 0.1 seconds.
4. `TREJ_VAL` is the time interval during which the LLC2 expects to receive a reply to a sent REJ PDU. The value is in units of 0.1 seconds.
5. `TBUSY_VAL` is the timer interval during which the LLC2 waits for an indication of the clearance of busy condition at the other LLC2. The value is in units of 0.1 seconds.
6. `TIDLE_VAL` is the time interval during which the LLC2 expects to receive a PDU from the other LLC2. The value is in units of 0.1 seconds.
7. `ACK_DELAY` is the RR delay time. This is the time interval for which the LLC2 will withhold acknowledgements of unacknowledged received I-PDUs. The value is in units of 0.1 seconds.
8. `NOTACK_MAX` is the maximum number of unacknowledged received I-frames.
9. `TX_WINDOW` is the transmit window (if no XID received).

10. `TX_PROBE` is the position before the window is closed at which an I-PDU is sent with the P-bit set to solicit an acknowledgement from the receiver.
11. `MAX_I_LEN` is the maximum size of an LLC2 I-frame. LLC2 requires all incoming I-frames above a certain size to be rejected by a FRMR. This parameter specifies the maximum size of data that may be received starting from the LLC2 protocol header.  
In an X.25 network, it is constructed as (maximum X.25 data length + X.25 protocol header length + LLC2 protocol header length). In an SNA network, it is constructed as (maximum SNA data length + SNA request header length + SNA transmission header length + LLC2 protocol header length).
12. `XID_WINDOW` is the XID window size (receive window), when the remote window size is unknown or zero.
13. `XID_NDUP` is the duplicate MAC XID count (0 means no test).
14. `XID_TDUP` is the duplicate MAC XID time. The value is in units of 0.1 seconds.

## Files

Files following this format are normally kept in the `‘/etc/sysconfig/strx25/template/’` directory.<sup>1</sup>

## See Also

- [lltune\(8\)](#)
- [llc2\(4\)](#)
- [x25netd\(8\)](#)

## Compatibility

The `‘llc2template’` file format is compatible with *Spider X.25*, and implementations based on *Spider X.25*, such as *AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *Solstice X.25*, *PT X.25*, *SBE X.25*, with the following compatibility considerations:

- *PT X.25* does not support LLC2. *OpenSS7 X.25 Networking* supports LLC2 in support of XOL and porting applications from *AIXlink/X.25*, *Solstice X.25*, *HP-UX*, *IRIS SX.25*, *VxWorks*, *pSOS*, *SpiderX*, and many other implementations based on *SpiderX.25* support LLC2. Portable X.25 and XOL applications will use *OpenSS7 X.25 Networking* instead of *PT X.25*.

For additional compatibility information see, [llc2\(4\)](#), and [STREAMS\(9\)](#).

## Conformance

*AIXlink/X.25*, *HP-UX*, *IRIS SX.25*, *Solstice X.25*, *PT X.25*, *SBE X.25*, documentation.

## History

The `‘llc2template’` file format first appeared in *Spider X.25*.

<sup>1</sup> Note that this directory varies depending on whether the build was on a `dpkg(1)`-based or `rpm(1)`-based system.

### 9.3 mlptemplate File Format

The 'mlptemplate' file format is documented in the `mlptemplate(5)` manual page.



## **Appendix A LLI Compatibility and Porting**

**A.1 Compatibility with AIXlink/X.25**

**A.2 Compatibility with HP X.25/9000**

**A.3 Compatibility with IRIS SX.25**

**A.4 Compatibility with PT X.25**

**A.5 Compatibility with RadiSys WAN**

**A.6 Compatibility with SBE X.25**

**A.7 Compatibility with Solstice X.25**



## Glossary of LLI Terms and Acronyms

<i>ANSI</i>	American National Standards Institute
<i>CCITT</i>	Old name for ITU-T
<i>CONS</i>	Connection-Oriented Network Service
<i>ENSDU</i>	Expedited Network Service Data Unit
<i>ETSI</i>	European Telecommunications Standards Institute
<i>IEEE</i>	Institute of Electrical and Electronics Engineers
<i>ITU</i>	International Telecommunications Union
<i>ITU-T</i>	ITU Telecom Sector
<i>LCI</i>	Logical Channel Identifier
<i>LLC1</i>	Logical Link Control Type 1
<i>LLC2</i>	Logical Link Control Type 2
<i>LLC</i>	Logical Link Control
<i>MAC</i>	Media Access Control
<i>NLI</i>	Network Layer Interface
<i>NPDU</i>	Network Protocol Data Unit
<i>NSAP</i>	Network Service Access Point
<i>NSDU</i>	Network Service Data Unit
<i>NSP</i>	Network Service Provider
<i>NS</i>	Network Service
<i>NSU</i>	Network Service User
<i>PDU</i>	Protocol Data Unit
<i>PVC</i>	Permanent Virtual Circuit
<i>SAP</i>	Service Access Point
<i>SDU</i>	Service Data Unit
<i>VC</i>	Virtual Circuit
<i>X.121</i>	ITU-T Recommendation X.121
<i>X.25</i>	ITU-T Recommendation X.25
<i>X.29</i>	ITU-T Recommendation X.29



## References

- [AIXlink/X.25] *AIXlink/X.25 Version 2.1 for AIX: Guide and Reference*, No: SC23-2520-07, Eighth Edition, September 2006, (Bolder, CO), International Business Machine Corp., IBM. [IBM Documentation Library](#).
- [ARTIC WAN] *ARTIC STREAMS Support WAN Driver Interface Reference*, Release 1.7, June 2004, (Hillsboro, OR), RadiSys Corporation, RadiSys. [Doc No: 007-01232-0003], [RadiSys Support Documentation](#).
- [CDI] *OpenSS7 CAE Specification: Communications Device Interface (CDI) Specification*, Revision 0.9.2, Draft 2, July 15, 2007, (Edmonton, Canada), B. Bidulock, OpenSS7 Corporation. Distributed with package ‘strxns-0.9.2’ and ‘openss7-0.9.2’. [OpenSS7 Documents](#).
- [DLPI] *Open Group CAE Specification: Data Link Provider Interface (DLPI) Specification*, Revision 2.0.0, Draft 2, August 20, 1992, (Parsippany, New Jersey), UNIX International, Inc., UNIX International Press. [The Open Group, The OpenSS7 Project](#).
- [IRIS SX.25] *IRIS SX.25 NLI Programmer’s Guide*, 1995, (Mountainview, CA), Silicon Graphics, Inc., SGI Technical Publications. [No: 007-2268-002]. [SGI Technical Publications](#).
- [ISO3309] **ISO/IEC 3309:1988**, *Information Technology – Data Communications – High-Level Data Link Control procedures – Frame Structure*, 1988, ISO/IEC, International Organization for Standardization. [International Organization for Standardization](#).
- [ISO8208] **ISO/IEC 8208:2000**, *Information Technology – Data Communications – X.25 Packet Layer Protocol for Data Terminal Equipment*, Fourth Edition, November 1, 2000, ISO/IEC, International Organization for Standardization. [International Organization for Standardization](#).
- [ISO7776] **ISO/IEC 7776:1995**, *Information technology – Telecommunications information exchange between systems – High-level data link control procedures – Description of the X.25 LAPB-compatible DTE data link procedures*, Second Edition, July 1, 1995, International Organization for Standardization. [International Organization for Standardization](#).

## References

- [ISO8802-2] **ANSI/IEEE Standard 802.2-1998 [ISO/IEC 8802-2:1998]**, *IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 2: Logical Link Control*, May 7, 1998, (New York), ANSI/IEEE, IEEE Computer Society. [ISBN 1-55937-959-6]. **Institute of Electrical and Electronics Engineers.**
- [ISO8881] **ISO/IEC 8881:1989**, *Information Processing Systems – Data Communications – User of the X.25 Packet Level Protocol in Local Area Networks*, 1989, ISO/IEC, International Organization for Standardization. **International Organization for Standardization.**
- [NLI] *OpenSS7 CAE Specification: Network Layer Interface (NLI) Specification*, Revision 0.9.2, Draft 1, June 2008, (Edmonton, AB), B. Bidulock, OpenSS7 Corporation. Distributed with package ‘strx25-0.9.2’ and ‘openss7-0.9.2’. **The OpenSS7 Project.**
- [NPI] *Open Group CAE Specification: Network Provider Interface (NPI) Specification*, Revision 2.0.0, Draft 2, August 17, 1992, (Parisppany, New Jersey), UNIX International, Inc., UNIX International Press. **The OpenSS7 Project.**
- [Solstice X.25] *Solstice X.25 9.2 Administration Guide*, October 1999, (Palo Alto, CA), Sun Microsystems, Inc., Sun. [Part No: 806-1234-10], **Solaris Documentation.**
- [TPI] *Open Group CAE Specification: Transport Provider Interface (TPI) Specification*, Revision 2.0.0, Draft 2, 1999, (Berkshire, UK), Open Group, Open Group Publication. **The Open Group, The OpenSS7 Project.**
- [V.25 bis] **ITU-T Recommendation V.25 bis (10/96)**, *Synchronous and asynchronous automatic dialing procedures on switched networks*, October 1996, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU, (Previously “CCITT Recommendation”), <http://www.itu.int/rec/T-REC-V.25bis/en/T-REC-V.25bis>.
- [X.21] **ITU-T Recommendation X.21 (09/92)**, *Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for synchronous operation on Public Data Networks*, September 1992, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU. (Previously “CCITT Recommendation”), **T-REC-X.21.**

- [X.21 bis] **ITU-T Recommendation X.21 bis (03/88)**, *Use on Public Data Networks of Data Terminal Equipment (DTE) which is designed for interfacing to synchronous V-series modems*, March 1988, (Geneva), ITU, ITU-T Telecommunication Standardization Sector of ITU. (Previously “CCITT Recommendation”), **T-REC-X.21bis**.
- [X.25] **ITU-T Recommendation X.25**. **T-REC-X.25**.
- [X.29] **ITU-T Recommendation X.29**. **T-REC-X.29**.
- [XX25] *X/Open CAE Specification: X.25 Programming Interface using XTI (XX25)*, No. c411, November 1995, (Berkshire, UK), X/Open, Open Group Publication. [ISBN: 1-85912-136-5]. **The Open Group**.





# Index

## A

ack\_delay ..... 20, 22  
 ACK\_DELAY ..... 34, 36  
 ADM ..... 34

## B

bytes\_rx ..... 18, 19  
 bytes\_tx ..... 18, 19

## D

DISC\_rx\_cmd ..... 14, 16  
 DISC\_tx\_cmd ..... 14, 16  
 DM ..... 34  
 DM\_rx\_rsp ..... 14, 16  
 DM\_tx\_rsp ..... 14, 16  
 dpkg(1) ..... 34, 37

## E

ERROR ..... 34

## F

frames\_rx ..... 18, 19  
 frames\_tx ..... 18, 19  
 FRMR\_FRMR\_ERROR ..... 34  
 FRMR\_INVRSP\_ERROR ..... 34  
 FRMR\_rx\_rsp ..... 14, 16  
 FRMR\_tx\_rsp ..... 14, 16

## G

getmsg(2s) ..... 3  
 getpmsg(2) ..... 3

## H

hdlc(4) ..... 29

## I

I\_LINK(7) ..... 12  
 I\_rx\_cmd ..... 14, 16  
 I\_rx\_rsp ..... 16  
 I\_tx\_cmd ..... 14, 16  
 I\_tx\_rsp ..... 17  
 IDLE\_VAL ..... 33  
 IGN\_DM\_ERROR ..... 34, 35  
 IGN\_UA\_ERROR ..... 34

ioctl(2) ..... 3

## L

L\_GETGSTATS ..... 23  
 L\_GETSNID ..... 12, 23  
 L\_GETSTATS ..... 23  
 L\_GETTUNE ..... 23  
 L\_GETX32MAP ..... 23  
 L\_LAPBTUNE ..... 33  
 L\_LINKDISABLE ..... 23  
 L\_LINKENABLE ..... 23  
 L\_LLC2TUNE ..... 36  
 L\_PUTX32MAP ..... 23  
 L\_SETSNID ..... 12, 23  
 L\_SETTUNE ..... 23  
 L\_TRACEOFF ..... 23  
 L\_TRACEON ..... 23  
 L\_ZEROGSTATS ..... 23  
 L\_ZEROSTATS ..... 23  
 labtune\_t ..... 20  
 lapb(4) ..... 29, 35  
 lapb\_gstioc ..... 18, 24, 26  
 lapb\_stioc ..... 13, 24, 25  
 lapb\_tnioc ..... 11, 19, 24, 25, 33  
 lapb\_tune ..... 20, 33  
 lapbgstats ..... 18  
 lapbmonarray ..... 14  
 lapbstats\_t ..... 13, 14  
 lapbtemplate(5) ..... 33  
 lapbtune\_t ..... 20  
 LC\_HDLC ..... 12  
 LC\_HDLCX ..... 12  
 LC\_LAPBDCE ..... 12  
 LC\_LAPBDTE ..... 12  
 LC\_LAPBXDCE ..... 12  
 LC\_LAPBXDTE ..... 12  
 LC\_LAPDCE ..... 12  
 LC\_LAPDTE ..... 12  
 LC\_LLC1 ..... 12  
 LC\_LLC2 ..... 12  
 LC\_MLAPBDCE ..... 12  
 LC\_MLAPBDTE ..... 12  
 LC\_MLAPBXDCE ..... 12  
 LC\_MLAPBXDTE ..... 12  
 LC\_MLP ..... 12  
 LI\_GSTATS ..... 11  
 LI\_LAPBTUNE ..... 11  
 LI\_LLC2TUNE ..... 11  
 LI\_MLPTUNE ..... 11  
 LI\_PLAIN ..... 11  
 LI\_SNID ..... 11

## Index

LI\_STATS ..... 11  
LI\_STATUS ..... 11  
linkadd(8) ..... 31  
linkdel(8) ..... 31  
linklist(8) ..... 31  
linkreset(8) ..... 31  
linkstart(8) ..... 31  
linkstate(8) ..... 31  
linkstop(8) ..... 31  
ll\_gstioc ..... 11  
ll\_hdioic ..... 11, 24, 25, 26  
ll\_snioc ..... 11, 12, 23, 25  
ll\_stioc ..... 11  
ll\_stnioc ..... 11, 13  
llc1(4) ..... 29  
llc2(4) ..... 29, 37  
llc2\_gstioc ..... 18, 24, 26  
llc2\_stioc ..... 15, 24, 25  
llc2\_tnioc ..... 11, 21, 24, 25, 36  
llc2\_tune ..... 21, 36  
llc2gstats ..... 18  
llc2monarray ..... 15  
llc2stats\_t ..... 15  
llc2template(5) ..... 36  
llc2tune\_t ..... 21  
llconform ..... 20  
lli\_class ..... 12  
LLI\_GSTATS ..... 18, 19  
lli\_index ..... 12  
LLI\_LAPBTUNE ..... 20  
LLI\_LLC2TUNE ..... 21  
LLI\_MLPTUNE ..... 22  
LLI\_PLAIN ..... 11  
lli\_slp\_pri ..... 13  
lli\_slp\_snid ..... 13  
lli\_snid ..... 11, 12, 13, 15, 17, 20, 21, 22  
LLI\_SNID ..... 12  
lli\_spare ..... 11, 12, 13, 15, 17, 18, 19, 20, 21, 22  
lli\_state ..... 17  
lli\_stats ..... 13, 15, 17  
LLI\_STATS ..... 13, 15, 17  
LLI\_STATUS ..... 13  
lli\_type ..... 11, 12, 13, 15, 17, 18, 19, 20, 21, 22  
lltune(8) ..... 31, 33, 35, 36, 37  
LOC\_PROBE ..... 34  
LOC\_WIND ..... 34

## M

M\_GETGSTATS ..... 23  
M\_GETSNID ..... 12, 23  
M\_GETSTATS ..... 23  
M\_GETTUNE ..... 23  
M\_SETSNID ..... 12, 23  
M\_SETTUNE ..... 23  
M\_TRACEOFF ..... 23

M\_TRACEON ..... 23  
M\_ZEROGSTATS ..... 23  
M\_ZEROSTATS ..... 23  
max\_I\_len ..... 20, 22  
MAX\_I\_LEN ..... 34, 37  
MLP\_bytes\_rx ..... 19  
MLP\_bytes\_tx ..... 19  
MLP\_confs\_rx ..... 17, 19  
MLP\_confs\_tx ..... 17, 19  
MLP\_frames\_rx ..... 17, 19  
MLP\_frames\_tx ..... 17, 19  
mlp\_gstioc ..... 11, 19, 24, 26  
MLP\_mn1\_exp ..... 18  
MLP\_mt1\_exp ..... 18  
MLP\_mt2\_exp ..... 18  
MLP\_mt3\_exp ..... 18  
MLP\_num\_slps ..... 17, 19  
MLP\_reset\_rx ..... 17, 19  
MLP\_reset\_tx ..... 17, 19  
MLP\_slps ..... 17, 19  
mlp\_stioc ..... 11, 17, 24, 25  
mlp\_tnioc ..... 11, 22, 24, 25  
mlp\_tune ..... 22  
mlpgstats ..... 19  
mlpmonarray ..... 17  
mlpstats\_t ..... 17  
mlptemplate(5) ..... 38  
mlptune(8) ..... 31  
mlptune\_t ..... 22  
mn1 ..... 23  
mt1 ..... 23  
mt2 ..... 23  
mt3 ..... 23  
mw ..... 23  
mx ..... 23

## N

N2 ..... 20, 21  
N2\_VAL ..... 33, 36  
NO\_DM\_ADM ..... 34  
notack\_max ..... 20, 22  
NOTACK\_MAX ..... 34, 36

## P

putmsg(2s) ..... 3  
putpmsg(2) ..... 3

## R

REJ\_rx\_cmd ..... 14, 16  
REJ\_rx\_rsp ..... 14, 16  
REJ\_tx\_cmd ..... 14, 16  
REJ\_tx\_cmd\_p ..... 14, 16  
REJ\_tx\_rsp ..... 14, 16

RNR\_rx\_cmd..... 14, 16  
 RNR\_rx\_rsp..... 14, 16  
 RNR\_tx\_cmd..... 14, 16  
 RNR\_tx\_cmd\_p..... 14, 16  
 RNR\_tx\_rsp..... 14, 16  
 rpm(1)..... 34, 37  
 RR\_rx\_cmd..... 14, 16  
 RR\_rx\_rsp..... 14, 16  
 RR\_tx\_cmd..... 14, 16  
 RR\_tx\_cmd\_p..... 14, 16  
 RR\_tx\_rsp..... 14, 16  
 rx\_bad..... 14, 16  
 rx\_badlen..... 14, 16  
 rx\_dud..... 15, 16  
 rx\_ign..... 15, 16  
 rx\_unknown..... 14, 16

## S

s\_lll(4)..... 30  
 s\_lllc(4)..... 30  
 sabm\_in\_x32..... 20  
 SABM\_IN\_X32..... 34, 35  
 sabm\_rx..... 18  
 SABM\_rx\_cmd..... 15  
 sabm\_tx..... 18  
 SABM\_tx\_cmd..... 15  
 sabme\_rx..... 19  
 SABME\_rx\_cmd..... 14, 16  
 sabme\_tx..... 19  
 SABME\_tx\_cmd..... 14, 16  
 SARM\_rx\_cmd..... 15  
 SARM\_tx\_cmd..... 15  
 SFRAME\_PBIT..... 34  
 state..... 13  
 STREAMS(9)..... 35, 37

## T

T1..... 20, 21  
 t1\_exp..... 14, 16

T1\_VAL..... 33, 34, 36  
 t4\_exp..... 14, 16  
 t4\_n2\_exp..... 14, 16  
 Tbusy..... 20, 22  
 TBUSY\_VAL..... 33, 36  
 TEST\_rx\_cmd..... 15, 16  
 TEST\_rx\_rsp..... 15, 16  
 TEST\_tx\_cmd..... 15, 16  
 TEST\_tx\_rsp..... 15, 16  
 Tidle..... 20, 22  
 TIDLE\_VAL..... 36  
 Tpf..... 20, 21  
 TPF\_VAL..... 33, 36  
 Trej..... 20, 22  
 TREJ\_VAL..... 33, 36  
 tx\_ign..... 14, 15  
 tx\_probe..... 20, 22  
 TX\_PROBE..... 37  
 tx\_rtr..... 14, 16  
 tx\_window..... 20, 22  
 TX\_WINDOW..... 36

## U

UA\_rx\_rsp..... 14, 16  
 UA\_tx\_rsp..... 14, 16  
 UI\_rx\_cmd..... 17  
 UI\_tx\_cmd..... 17

## X

x25netd(8)..... 35, 37  
 xid\_Ndup..... 22  
 XID\_NDUP..... 37  
 XID\_rx\_cmd..... 15, 16  
 XID\_rx\_rsp..... 15, 16  
 xid\_Tdup..... 22  
 XID\_TDUP..... 37  
 XID\_tx\_cmd..... 15, 16  
 XID\_tx\_rsp..... 15, 16  
 xid\_window..... 22  
 XID\_WINDOW..... 37

