COMMON APPLICATION LAYERS

V1.4

Version	Modifications	Date
1.0	Initial version	22/09/2017
1.1	Separation of common applicative layer elements into a specific document	07/06/2019
1.2	Revision of the L6App table for backward compatibility	01/02/2020
1.2.1	Editorial corrections (references and table numbers)	21/11/2020
1.3	Minor improvements for passive roaming support : - Modification of the semantics of COMMAND_WRITEKEY for simultaneous change of L6NetwId and KMAC - Addition of an optional L6PrimaryNetwId to INSPONG mlessages	08/07/2021
1.4	Editorial review, internal document reference updated in § 4.4.1.3, 4.4.1.4 and 4.4.1.5	11/10/2023

Summary

This document outlines the Detailed Functional and Technical Specification of the Wize LAN protocol common application layers (mandatory for any Wize device). The LAN network designates the medium range radio network between the devices and the gateways.

Table of contents

Summary *1*
Table of contents *3*
Tables and figures *4*
Contributors *5*
1 Introduction 6
2 Reference documents 7
2.1 Applicable standards 7
2.2 Main specifications 7
3 Application Layers index 8
4 Detailed specification of the common application layers 9
4.1 Mode for representation of dates and times 9
4.2 Device configuration model 9
4.3 APP-INSTALL application layer (mandatory) 11
4.3.1 INSTPING message 11
4.3.2 INSTPONG message 12
4.4 APP-ADMIN application layer (mandatory) 12
4.4.1 COMMAND messages 12
4.4.2 RESPONSE messages 17
4.5 APP-DOWNLOAD application layer (optional) 22
5 Appendix A: DATA LAN parameter dictionary 23
5.1 LAN parameters 23
6 Informative Appendix B: L6App table as available to publishing date of this document 26

Tables and figures

Figure 1 : Configuration model 10

Table 1 : Encoding time stamps 9
Table 2 : INSTPING application message format 11
Table 3 : INSTPONG application message format 12
Table 4 : COMMAND_READPARAMETERS application message format 13
Table 5 : COMMAND_WRITEPARAMETERS application message format 14
Table 6 : COMMAND_WRITEKEY application message format 15
Table 7 : COMMAND_ANNDOWNLOAD application message format 17
Table 8 : COMMAND_EXECINSTPING application message format 17
Table 9 : RESPONSE_CMDERROR application message format 18
Table 10: Response format in the case of a supported command 18
Table 11: Response format in the case of an error 18
Table 12 : RESPONSE_READPARAMETERS application message format 18
Table 13: Encoding the error causes of the READPARAMETERS command 19
Table 14:RESPONSE_WRITEPARAMETERS application message format 19
Table 15 : Encoding the error causes of the WRITEPARAMETERS command 19
Table 16 : RESPONSE_WRITEKEY application message format 20
Table 17 : Encoding the error causes of the COMMAND_WRITEKEY command 20
Table 18 : RESPONSE_ANNDOWNLOAD application message format 20
Table 19 : Encoding the error causes of the ANNDOWNLOAD command 21
Table 20 : RESPONSE_EXECINSTPING application message format 22
Table 21 : Encoding the error causes of the WRITEPARAMETERS command 22
Table 22 : LAN parameters 26
Table 23:L6App table 26

Contributors

See document [R1]

1 Introduction

This document **specifies the common Application layers** of the Wize LAN protocol : APP-INSTALL, APP-ADMIN, APP-DOWNLOAD.

.

2 Reference documents

2.1 Applicable standards

See document [R1].

2.2 Main specifications

Reference	Document	Version
R[1]	Wize Alliance : WIZE – 01 LAN protocol Specification	V1.3
R[2]	Wize Alliance : WIZE – 02 Regional parameters	V1.1

3 Application Layers index

In addition to physical, data link and presentation layers, the Wize Protocol specifies several application layers to satisfy the requirements of all targeted end application, while using the same infrastructure and transport protocols. These application layers are part of the Wize Specification but specified in separate documents published under the control of the Wize Alliance Technical Committee.

The Wize application layers are split into two categories :

- **Common application layers** (mandatory for any Wize device).
- **Specific application layers** (optional, specific to each target end application. A given Wize device can support one or several specific application layers, or even no specific application layer (test devices for example).

The selection of the application layer for a given message is done through the L6App field, transmitted as part of the L6 presentation layer header

The L6App table lists all the L6App IDs corresponding to all application layers allocated by the Wize Alliance. The updated version shall be found on the Wize Alliance Website, an example can be found in Informative Appendix B of this document.

NOTE: Members are free to create a new Application layer tailored to their specific needs. If one member wishes to share a new Application layer with the Wize community please get in touch with the Wize Alliance in order to submit it, have it provided with a specific L6App ID and added to the L6App management table.

In this case, please send an email to contact@wize-alliance.com, with the subject “New L6App ID request”.

4 Detailed specification of the common application layers

Link Layer format is defined in Wize – 01 LAN Protocol document, see §2.1 Format of LLC- Exchange frames

4.1 Mode for representation of dates and times

Date and time representation formats can be freely defined by each application layer. However, to guarantee homogeneity and flexibility, the following format is recommended to transfer the time indications (dates and times) on the LAN interface in the application level frames:

- For an absolute time indication: in the form of an EPOCH encoded over 32 bits and corresponding to the number of seconds since the 1st of January 2013 at 00:00 (easily convertible into EPOCH Unix by adding a constant to the Head-End system).
- For a relative time indication during the day: in the form of the number of seconds since midnight on the current day (0 to 86399), divided by two (0 to 43200) and encoded over 2 bytes, thus with a resolution of 2 seconds.
- For L6TStamp : the 2 least significant bytes of the absolute EPOCH with one second resolution.

Numerical value V to be represented	Byte 1	Byte 2	Byte 3	Byte 4
Absolute, 1/1/2013 00:00:00 to 7/2/2149 06:28:00	MSB EPOCH	MID1 EPOCH	MID2 EPOCH	LSB EPOCH
Relative in a day, from 00:00 to 23:59:59	MSB NSEC/2	LSB NSEC/2		
L6TStamp	MID2 EPOCH	LSB EPOCH		

Table 1 : Encoding time stamps

If necessary, differences in time between two dates are represented by the number of seconds between the two times.

4.2 Device configuration model

For a simple and consistent administration and monitoring model, the devices are managed via data table readings and writings. This table contains two types of data:

- Status variables and parameters relating to actual LAN protocol management and common application layers, specified in this document (see 5.1), numbered from $00 to $5F
- Status variables and parameters relating to specific application layers, numbered from $60 to $FF

Note: the same reading/writing mechanism for a generic parameter table is used to access the device via the local interface.

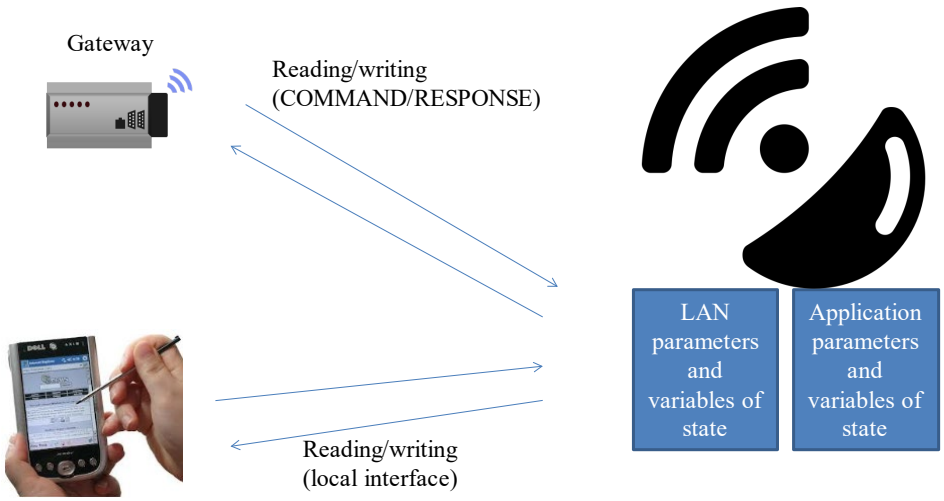


Figure 1 : Configuration model

The devices allow reading and/or modification of the values of each of the LAN and application parameters via COMMAND/RESPONSE messages of the APP-ADMIN application layer (see 4.4), even for variables and parameters defined by other specific application layers if they follows the recommended Wize model.

An error code must be returned by the device in the event of illegal operations :

- Attempt to read or write a non-existing parameter
- Attempt to read a parameter only accessible in writing or to write a parameter only accessible in reading
- Attempt to write a parameter with a value outside the acceptable ranges
- Other error cause specified by the manufacturer

4.3 APP-INSTALL application layer (mandatory)

4.3.1 INSTPING message

Only one message can be sent by a device using the APP-INSTALL application layer, named INSTPING. The format of the application Layer payload of this INSTPING message, sent by a device to determine the gateways and LAN modems nearby, is as follows:

Byte	Description	Comments
1	L7DownChannel	Frequency channel to be used by the gateway for the response (same coding as RF_UPLINK_CHANNEL parameter, see Regional Parameters document)
2	L7DownMod	Modulation to be used by the gateway for the response (same coding as RF_UPLINK_MOD parameter, see Regional Parameters document)
3	L7PingRxDelay	Value of the current PING_RX_DELAY parameter of the device, see APPENDIX A.
4	L7PingRxLength	Value of the current PING_RX_LENGTH parameter of the device, see APPENDIX A.

Table 2 : INSTPING application message format

INSTPING and INSTPONG messages are never encrypted (L6Ctrl.L6KeySel=0) and only contain an authentication footprint calculated using the Kmac key, see LAN Protocol document, section 6.1.

Note 1: in terms of security, INSTPING/INSTPONG exchanges will thus be on the same level as Kmac authentication.

Note 2: the L7DownChannel and L7DownMod parameters are positioned by the device at the values defined by the RF_DOWNLINK_CHANNEL and RF_DOWNLINK_MOD parameters, respectively. These parameters allow a gateway to respond to an INSTPING message even if there is no connection with the Head-End system.

On reception of an INSTPING message, the gateway respond by an INSTPONG message for each LAN modem that has received the message.

In particular, the INSTPING/INSTPONG mechanism do support passive roaming. More precisely, if several Wize operators provide service in a given area then all of them respond to the INSTPING message sent by a device, as long as the L6NetwId field of the INSTPING message is one of their supported virtual networks. In particular, if the L6NetwId field of the INSTPING message is the L6NetwId of the OneWize easy commissioning virtual network then all networks supporting OneWize service must answer with an INSTPONG.

4.3.2 INSTPONG message

The format of the application layer payload of the INSTPONG message, returned by each gateway for each LAN modem that has received an INSTPING message, is as follows:

When L6NetwId of the INSPING message is NOT the L6NetwId of OneWize easy commissioning service :

Byte	Description	Comments
1	L7ConcentId	Gateway numerical identification: same as the ID
2		
3		Used in WAN messages. MSBs first
4		
5		
6		
7	L7ModemId	Numerical identification of the LAN modem that has received the message
8	L7RSSI	Reception RSSI by the LAN modem of the INSTPING message transmitted by the device. Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)

When L6NetwIf of the INSTPING message is the L6NetwId of OneWize easy commissioning service :

Byte	Description	Comments
1	L7ConcentId	Gateway numerical identification: same as the ID
2		
3		Used in WAN messages. MSBs first
4		
5		
6		
7	L7ModemId	Numerical identification of the LAN modem that has received the message
8	L7RSSI	Reception RSSI by the LAN modem of the INSTPING message transmitted by the device. Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)
9	L6PrimaryNetwId	Primary L6NetwId of the receiving network/gateway

Table 3 : INSTPONG application message format

Note: as a reminder, INSTPING and INSTPONG messages are never encrypted and only contain an authentication footprint calculated using the Kmac key

4.4 APP-ADMIN application layer (mandatory)

4.4.1 COMMAND messages

4.4.1.1 COMMAND_READPARAMETERS

The format of the application part of the COMMAND_READPARAMETERS message allowing the Head-End system to read one or more parameters in a device, is as follows:

Byte	Description	Comments
1	L7CommandId	Fixed value $10 ID COMMAND_READPARAMETERS
2	ParamId1	ID of the first parameter to read
3	ParamId2	ID of the second parameter to read
4	…	Etc.

Table 4 : COMMAND_READPARAMETERS application message format

The parameter IDs are defined in appendix A.

On reception of a COMMAND_READPARAMETERS message, the device must respond by a RESPONSE_READPARAMETERS message (see 4.4.2.3).

4.4.1.2 COMMAND _WRITEPARAMETERS

The format of the application part of the COMMAND_WRITEPARAMETERS message allowing one or more parameters to be written in a device, is as follows:

Byte	Description	Comments
1	L7CommandId	Fixed value $20 ID COMMAND_WRITEPARAMETERS
2	ParamId1	ID of the first parameter to write
3	ParamValue1	Value of the first parameter (number of bytes T1)
…	…	…
3+T1	ParamId2	ID of the second parameter to write
4+T1	ParamValue2	Value of the second parameter (number of bytes T2)
…	…	Etc.

Table 5 : COMMAND_WRITEPARAMETERS application message format

The parameter IDs are defined in appendix A.

Each device can support the reception of application level frames of a given maximum length, according to its internal design, which in turn depends on the maximum size of the application messages to be received. This length is defined by the MANUFACTURER and is accessible by reading the parameter L7RECEIVE_LENGTH_MAX.

Should the gateway send the device a message of a size greater than the maximum size supported in reception by the device, the latter must ignore the command. A minimum value of parameter L7RECEIVE_LENGTH_MAX is defined in 5.1.

The Head-End system can thus either use this minimum value, by definition supported by all the devices, or optimise its requests to the capacities of a given device by reading the value of this parameter.

On reception of a COMMAND_WRITEPARAMETERS message, the device must first respond by a RESPONSE_WRITEPARAMETERS message (see 4.4.2.4), before taking into account the new parameters immediately after sending the response. The response must thus be sent using the old parameters (in particular in case of change of radio channel or similar).

4.4.1.3 COMMAND_WRITEKEY

The format of the application part of the COMMAND_WRITEKEY message allowing modification of the key of a device, is as follows:

Byte	Description	Comments
1	L7CommandId	Fixed value $28 ID COMMAND_WRITEKEY
2	L7KeyId	ID of the key to be modified. Values authorised = 01,corresponding to the Kmob key, 02 for the Kmac Key The other values are reserved for future upgrades of the protocol.
3 to 34	L7KeyVal	New value of the key (256 bits, MSB first)
35	L7KIndex	Index of new key For Kmob key, this field contains L7KmobIndex in case of Kmob, For Kmac key, this field : - Must be set to 0 if the device must keep its L6NETWID

		unchanged - Must be set to the new L6NETWID value of the device if the device must change simultaneously its L6NETWID. Value from 0 to 255

Table 6 : COMMAND_WRITEKEY application message format

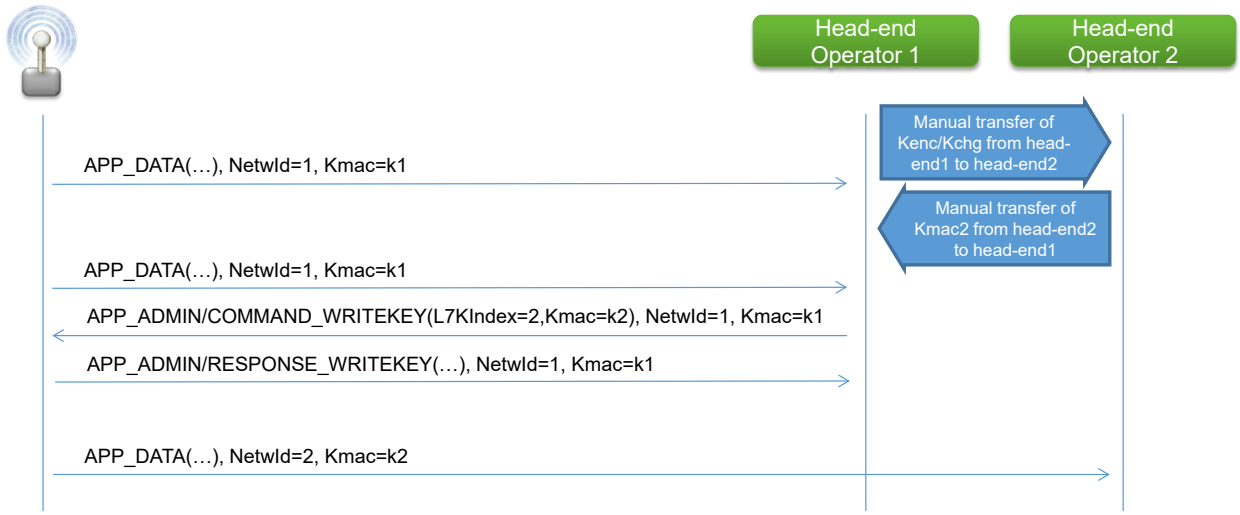
This message is always sent using the Kchg encryption and authentication key, where the L6Ctrl.L6KeySel key equals 15. The device must refuse the command and return an error code if this is not the case.

Note: the L7KeyVal field is defined over 128 bits in the current version of the LAN protocol specification. If a longer key needs to be supported, this will occur through a change in LAN specification, identified by the L6Ctrl.L6Vers protocol version number.

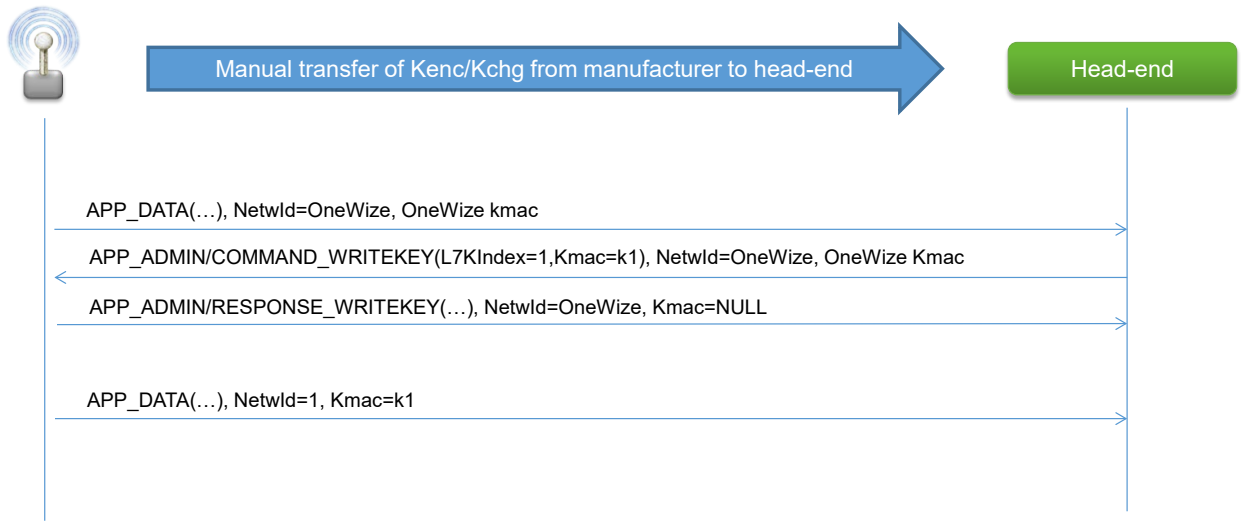
On reception of a COMMAND_WRITEKEY message, the device must respond by a RESPONSE_WRITEKEY message (see 4.4.2.5RESPONSE_WRITEKEY), using the Kchg encryption and authentication key.

Nota : The COMMAND_WRITEKEY message is then used to move a device from a virtual network to another virtual network. This can be used either to effectively change from one Wize operator to another one, or to reconfigure the device from OneWize virtual network to an actual Wize network identifier :

Example of moving from network 1 to network 2 :



Example of moving from OneWize to actual network identifier :



4.4.1.4 COMMAND_ANNDOWNLOAD

The format of the application part of the COMMAND_ANNDOWNLOAD message allowing notification of a software download sequence, is as follows:

Byte	Description	Comments
1	L7CommandId	Fixed value $30 ID COMMAND_ANNDOWNLOAD
2 3 4	L7DwnldId	Identification number of the software download sequence (the same ID is then used in the L2Dwnld field on effective download of the software, see 4.5 APP-DOWNLOAD application layer (optional)) MSBs first
5 … 20	L7Klog	Encryption key used for software download 128 bits, MSBs first
21 22	L7SwVersionIni	Initial software version of the device MSBs first
23 24	L7SwVersionTarget	Final software version of the device MSBs first
25 26	L7MField	Indication of the device MANUFACTURER as per EN13757-4 (LSBs first)
27 28	L7DcHwId	Identification of the hardware version of the device MSB first
29 30	L7BlocksCount	Number of software download blocks
31	L7ChannelId	Identification of the software download channel (same coding as RF_UPLINK_CHANNEL parameter, see Regional Parameters document)
32	L7ModulationId	Identification of the modulation used for software download (same coding as RF_UPLINK_MOD parameter, see Regional Parameters document)
33 34 35	L7DaysProg	First broadcasting date, encoded over 4 bytes as EPOCH. The device must check : Current_epoch-(10x24x3600)<= L7DaysProg and L7DaysProg <= Current_epoch+(60x24x3600). If not error is returned

36		
37	L7DaysRepeat	Number of broadcasting days (From 1 to 15).
38	L7DeltaSec	Time between the transmission of the two data blocks from beginning to beginning in tele distribution, in seconds According to the value of parameter L7ModulationId : • The parameter of L7DeltaSec can take on a value of 10 to 255 to 2400 bauds, and of 5 to 255 to 4800 bauds • The recommended value of L7DeltaSec is of 10 (10 seconds) to 2400 bauds, and of 5 (5 seconds) to 4800 bauds
39 40 41 42	HashSW	Integrity check of the complete software. This field is defined by the MANUFACTURER, transmitted in the software download notification by the Head-End system, and checked by the software developed by the MANUFACTURER once all the blocks for a software download have been received

Table 7 : COMMAND_ANNDOWNLOAD application message format

This message is always transmitted using the Kch encryption and authentication key, where the L6Ctrl.L6KeySel field equals 15. The device must refuse the command and return an error code if this is not the case.

Note: the L7Klog field is defined over 128 bits in the current version of the LAN protocol specification. If a longer key needs to be supported, this will be achieved by a change in LAN specification, identified by the L6Ctrl.L6Vers protocol version number

On reception of a COMMAND_ANNDOWNLOAD message, the device must respond by a RESPONSE_ANNDOWNLOAD message (see 4.4.2.6), using the Kchg encryption and authentication key.

4.4.1.5 COMMAND_EXECINSTPING

The format of the application part of the COMMAND_EXECINSTPING message requesting carrying out of an INSTPING connectivity verification sequence by the device, is as follows:

Byte	Description	Comments
1	L7CommandId	Fixed value $40 ID COMMAND_EXECINSTPING

Table 8 : COMMAND_EXECINSTPING application message format

On reception of a COMMAND_EXECINSTPING message, the device must carry out an INSTPING sequence (see 4.3.1 INSTPING message) and then respond by a RESPONSE_EXECINSTPING message.

4.4.2 RESPONSE messages

4.4.2.1 RESPONSE_CMDERROR

Should the device receive an unsupported command, the format of the application part of the returned message is as follows:

Byte	Field name	Description
1	L7ResponseId	Value of the L7CommandId field of the unsupported command
2	L7ErrorCode	Fixed value 255 (0xFF) identifying the “unsupported command” error type

Table 9 : RESPONSE_CMDERROR application message format

4.4.2.2 Response in the case of a supported command

In the case of a supported command being received by the device either a response message or an error message is transmitted back to the gateway.

The general format of a response message is as follows:

Byte	Field name	Description
1	L7ResponseId	Value of the L7CommandId field of the command
2	L7ErrorCode	00 if correct execution, see details in following sections
3 .. n	Fields of the response message	The response message data, see following sections

Table 10: Response format in the case of a supported command

The general format of an error message is as follows:

Byte	Field name	Description
1	L7ResponseId	Value of the L7CommandId field of the command
2	L7ErrorCode	Error code, see details in the following sections
3	L7ErrorParam	See details in the following sections

Table 11: Response format in the case of an error

4.4.2.3 RESPONSE_READPARAMETERS

The format of the application part of the RESPONSE_READPARAMETERS message returned in response to a COMMAND_READPARAMETERS message, is as follows:

Byte	Description	Comments
1	L7ResponseId	Fixed value $10 ID RESPONSE_READPARAMETERS
2	L7ErrorCode	00 if correct execution, see below
3	L7SwVersion	Current software version of the device
4		(MSBs first)
5	L7Rssi	Reception RSSI by the device of the corresponding COMMAND message Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)
6	ParamId1	ID of the first parameter read
7	ParamValue1	Value of the first parameter (number of bytes T1, as per 5.1)
…	…	…
7+T1	ParamId2	ID of the second parameter read
8+T1	ParamValue2	Value of the second parameter (number of bytes T2, as per 5.1)
…	…	Etc.

Table 12 : RESPONSE_READPARAMETERS application message format

The parameter IDs are defined in appendix A.

In event of error in carrying out of the COMMAND_READPARAMETERS command by the device, the L7ErrorCode byte and the L7ErrorParam byte of the message are positioned at one of the following values according to the first error detected by the device:

L7ErrorCode	Cause	L7ErrorParam
1	Number of the unsupported parameter	Faulty ParamId value
2	Attempt to read a “write only” parameter	Faulty ParamId value
3	Response too long to be transmitted	Length of the response to be transmitted
4 to 255	Reserved errors	Reserved

Table 13: Encoding the error causes of the READPARAMETERS command

Regarding L7ErrorCode 3 in this case, each device can support the transmission of application level frames of a given maximum length, according to its internal design, which in turn depends on the application messages to be transmitted. This length is defined by the MANUFACTURER and is accessible by reading the parameter L7TRANSMIT_LENGTH_MAX (see 5.1).

In event of a read request generating a response from the device to the gateway that is larger in size than the maximum size supported, the device must return the error code “Response too long to be transmitted” in its response.

4.4.2.4 RESPONSE_WRITEPARAMETERS

The format of the application part of the RESPONSE_WRITEPARAMETERS message returned in response to a COMMAND_WRITEPARAMETERS message, is as follows:

Byte	Description	Comments
1	L7ResponseId	Fixed value $20 ID RESPONSE_WRITEPARAMETERS
2	L7ErrorCode	00 if correct execution, see below
3	L7SwVersion	Current software version of the device
4		(MSBs first)
5	L7Rssi	Reception RSSI by the device of the corresponding COMMAND message Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)

Table 14:RESPONSE_WRITEPARAMETERS application message format

The parameter IDs are defined in appendix A.

In event of error in carrying out of the COMMAND_WRITEPARAMETERS command by the device, the L7ErrorCode byte and the L7ErrorParam byte of the message are positioned at one of the following values according to the first error detected by the device:

L7ErrorCode	Cause	L7ErrorParam
1	Number of the unsupported parameter	Faulty ParamId value
2	Attempt to write a “read only” parameter	Faulty ParamId value
3	Attempt to assign an illegal value to a parameter	Faulty ParamId value
4 to 255	Reserved errors	Reserved

Table 15 : Encoding the error causes of the WRITEPARAMETERS command

4.4.2.5 RESPONSE_WRITEKEY

The format of the application part of the RESPONSE_WRITEKEY message returned in response to a COMMAND_WRITEKEY message, is as follows:

Byte	Description	Comments
1	L7ResponseId	Fixed value $28 ID RESPONSE_WRITEKEY
2	L7ErrorCode	00 if correct execution, see below

3 4	L7SwVersion	Current software version of the device (MSBs first)
5	L7Rssi	Reception RSSI by the device of the corresponding COMMAND message Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)

Table 16 : RESPONSE_WRITEKEY application message format

In event of error in carrying out of the COMMAND_WRITEKEY command by the device, the L7ErrorCode byte and the L7ErrorParam byte of the message are positioned at one of the following values according to the first error detected by the device:

L7ErrorCode	Cause	L7ErrorParam
1	Incorrect frame length	Real length of frame
2	Illegal value of a parameter	Number of faulty field (0 for the first, etc)
3	Kchg encryption key not used	L7Ctrl.L7KeySel value of message received
4 to 255	Reserved errors	Reserved

Table 17 : Encoding the error causes of the COMMAND_WRITEKEY command

4.4.2.6 RESPONSE_ANNDOWNLOAD

The format of the application part of the RESPONSE_ANNDOWNLOAD message returned in response to a COMMAND_ANNDOWNLOAD message, is as follows:

Byte	Description	Comments
1	L7ResponseId	Fixed value $30 ID RESPONSE_ANNDOWNLOAD
2	L7ErrorCode	00 if correct execution, see below
3 4	L7SwVersion	Current software version of the device (MSBs first)
5	L7Rssi	Reception RSSI by the device of the corresponding COMMAND message Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm)

Table 18 : RESPONSE_ANNDOWNLOAD application message format

In event of error in carrying out of the COMMAND_ANNDOWNLOAD command by the device, L7ErrorCode byte and the L7ErrorParam byte of the message are positioned at one of the following values according to the first error detected by the device:

L7ErrorCode	Cause	L7ErrorParam
1	Illegal value of a parameter	Number of faulty field (0 for the first, etc)
2	Incorrect frame length	Real length of frame
3	Incorrect initial software version	LSB of the current software version
4	Incorrect hardware version of the device	LSB of the current hardware version
5	Kchg encryption key not used	L7Ctrl.L7KeySel value of the message received

6	Incorrect broadcasting start day	Current day minus day of first broadcast (signed integer in complement on two, bounded at -128/+127)
7	Operation refused as an update is in progress on the local interface	LSB of the software version currently being received via the local interface
8	Target software version	LSB of current software
9	Target version already downloaded, waiting for update	LSB of current software
10	Diffusion time out of broadcasting window	Local time proposed from 0 to 23
11 to 255	Reserved errors	Reserved

Table 19 : Encoding the error causes of the ANNDOWNLOAD command

4.4.2.7 RESPONSE_EXECINSTPING

The format of the application part of the RESPONSE_EXECINSTPING message returned in response to a COMMAND_EXECINSTPING message, is as follows:

Byte	Description	Comments
1	L7ResponseId	Fixed value $40 ID RESPONSE_EXECINSTPING
2	L7ErrorCode	00 if correct execution, see below
3 4	L7SwVersion	Current software version of the device (MSBs first)
5	L7Rssi	Reception RSSI by the device of the corresponding COMMAND message Value in steps of 0.5dBm, from 0 (-147.5dBm) to 255 (-20dBm), rounded to the nearest value or to the lower value in event of equality.
6	L7NbPong	Number of different INSTPONG messages received in response to the last connectivity test
7 to 15	L7Pong1	Response 1 received for the last connectivity test (most significant L7RssiDown). Formed by concatenation of the following fields: • L7ConcentId (6 bytes coded in BCD) • L7ModemId (1 byte) • L7RSSIUp (1 byte) • L7RssiDown (1 byte)
16 to 24	L7Pong2	Response 2 received for the last connectivity test (next most significant L7RssiDown). Formed by concatenation of the following fields: • L7ConcentId (6 bytes coded in BCD) • L7ModemId (1 byte) • L7RSSIUp (1 byte) • L7RssiDown (1 byte)

25 to 33	L7Pong3	Response 3 received for the last connectivity test (next most significant L7RssiDown). Formed by concatenation of the following fields: • L7ConcentId (6 bytes coded in BCD) • L7ModemId (1 byte) • L7RSSIUp (1 byte) • L7RssiDown (1 byte)

Table 20 : RESPONSE_EXECINSTPING application message format

If less than 3 responses are found, the L7PongX fields not used (L7Pong3 if 2 responses, L7Pong2 and L7Pong3 if 1 response, all if 0 responses) are initialised to zero by the device. If there are exactly 3 responses, the RESPONSE_EXECINSTPING message will give the Head-End system all the responses. If more than 3 responses are found, the Head-End system can read the next responses by carrying out a device memory read command (see 5.1).

In event of error in carrying out of the COMMAND_EXECINSTPING command by the device, the L7ErrorCode byte and the L7ErrorParam byte of the message are positioned at one of the following values according to the first error detected by the device:

L7ErrorCode	Cause	L7ErrorParam
1 to 255	Reserved errors	Reserved

Table 21 : Encoding the error causes of the WRITEPARAMETERS command

4.5 APP-DOWNLOAD application layer (optional)

The APP-DOWNLOAD application layer is used for the transmission of data blocks during software download. These data blocks have a fixed size defined by the LLC- DOWNLOAD layer (see WIZE – 01 LAN Protocol document, section 4.2), and free content. Structuring and use of the content of these blocks is the responsibility of the MANUFACTURER of the device in question, according to the conditions defined in chapter 4.2 of the WIZE-01 LAN Protocol document.

5 Appendix A: DATA LAN parameter dictionary

5.1 LAN parameters

The ID range $00 to $5F is reserved for LAN protocol management parameters. In this range, the following parameters must be supported by all the devices:

Id	Parameter name	Description	Size (bytes)	Mode	L/R	Coding	Default value
01	**VERS_HW_TRX**	Hardware version number of the device (or transceiver for a remote module)	2	R	L/R	Byte 1 : Version, Byte 2 : Revision	As per MANUFACTURER
02	**VERS_FW_TRX**	Software version number run by the device (or transceiver for a remote module)	2	R	L/R	Byte 1 : Version, Byte 2 : Revision	As per MANUFACTURER
03	**DATEHOUR_LAST_UPDATE**	Date/time of the last successful firmware download	4	R	L/R	EPOCH encoded on 32 bits and corresponding to the number of seconds since 1st January 2013 at 00:00: MSBs first (big endian)	00000000
04	**L6APP**	Primary application of the device as defined by WIZE association used for information to the Head End System; see WIZE -01- LAN protocol §1.8	1	R	L/R	set in factory as per Wize Alliance definition, ex:00h for Water Metering	00h
08	**RF_UPLINK_CHANNEL**	Frequency channel to be used for all uplink message transmissions	1	R/W	L/R	See Regional Parameters document in the corresponding frequency band for parameter value.	See Regional Parameters document.
09	**RF_DOWNLINK_CHANNEL**	Frequency channel to be used for all message receptions (except firmware download)	1	R/W	L/R	See Regional Parameters document in the corresponding frequency band for parameter value.	See Regional Parameters document.
0A	**RF_UPLINK_MOD**	Modulation to be used for all uplink message transmissions	1	R/W	L/R	See Regional Parameters document in the corresponding frequency band for parameter value.	See Regional Parameters document.
0B	**RF_DOWNLINK_MOD**	Modulation to be used for all message receptions (except firmware download)	1	R/W	L/R	See Regional Parameters document in the corresponding frequency band for parameter value.	See Regional Parameters document.
10	**TX_POWER**	Transceiver nominal transmission power	1	R/W	L/R	See Regional Parameters document in the corresponding frequency band for parameter value.	See Regional Parameters document.

11	**TX_DELAY_FULLPOWER**	Maximum time between two COMMAND messages before the device automatically returns to maximum transmission power	2	R/W	L/R	Number of days, from 1 to 65535 Value 0000:function disabled Byte 1: MSBs Byte 2: LSBs	100 (three months and 10 days)
12	**TX_FREQ_OFFSET**	Absolute correction of transmission frequency	2	R/W	L/R	In Hertz, from -32768 (-32.768KHz) to +32767 (+32.767KHz). Signed number on 16 bits encoded in 2-complement : Byte 1: MSBs Byte 2: LSBs Note: the device can round off this value, provided that the accuracy requirements specified in chapter 5 are complied with).	Calibrated in factory
18	**EXCH_RX_DELAY**	Fixed wait time after transmission of a DATA message by the device and before opening the COMMAND message listening window	1	R/W	L/R	In seconds, from 1 (1s) to 255 (255s)	5 (5s)
19	**EXCH_RX_LENGTH**	Duration of the COMMAND message listening window by the device	1	R/W	L/R	In multiples of 5 milliseconds, from 0 (reception disabled) to 255 (1.27s)	8 (40ms)
1A	**EXCH_RESPONSE_DELAY**	Time between reception of a COMMAND message by the device and transmission of the corresponding RESPONSE message	1	R/W	L/R	In seconds, from 0 (0s) to 255	In seconds, from 0(0s) to EXCH_RESPONSE_DELAY_MIN
1B	**EXCH_RESPONSE_DELAY_MIN**	Minimum value accepted for the EXCH_RESPONSE_DELAY parameter (defined by the device MANUFACTURER)	1	R	L/R	In seconds, from 0 (0s) to 255 (255s)	As per MANUFACTURER
1C	**L7TRANSMIT_LENGTH_MAX**	Maximum length of application messages that can be sent by the device (fixed value defined by SUEZ)	1	R	L/R	In bytes, from 40 to 100	As per MANUFACTURER Target value: 80
1D	**L7RECEIVE_LENGTH_MAX**	Maximum length of application messages that can be received by the device (Fixed value defined by SUEZ)	1	R	L/R	En bytes, de 50 à 100	As per MANUFACTURER Taget value: 100
20	**CLOCK_CURRENT_EPOC**	Current time of device	4	R/W	L/R	EPOCH encoded on 32 bits and corresponding to the number of seconds since 1st January 2013 at 00:00: MSB first (big endian) Programmed in factory	N/A
21	**CLOCK_OFFSET_CORRECTION**	Relative correction (time delta) to be applied to the device clock once only to correct its absolute drift	2	W	L/R	All values from -32768s to +32767s. Signed number on 16 bits encoded in complement on 2: Byte 1: MSBs Byte 2: LSBs This parameter is a virtual parameter: each writing corrects the current time in relative manner	00h

22	**CLOCK_DRIFT_CORRECTION**	Correction of device clock frequency	2	R/W	L/R	Number S of seconds to add to or subtract from the current time every D days: Byte 1: Number of seconds S (signed integer from -128 to +127, complement on two) Byte 2: Number of days D (from 1 to 255)	00h, 01h
28	**CIPH_CURRENT_KEY**	Current key number	1	R/W	L/R	00: No encryption (not accepted through APP-ADMIN request) 01 to CIPH_KEY_COUNT: Kenc key number enabled Other values: reserved	Value configured when manufacturing the devices: 01h
29	**CIPH_KEY_COUNT**	Number of encryption keys available in the device	1	R	L/R	unsigned int	14
2A	**L6NETWID**	Value used by the device to initialize the L6NetwId field of any upstream messages	1	R	L/R	00 to 255	As per operator
30	**PING_RX_DELAY**	Fixed waiting time after transmission of an INSTPING message by the device and before opening the INSTPONG message listening window	1	R/W	L/R	In seconds, from 1 (1s) to PING_RX_DELAY_MIN	10 (10s) or PING_RX_DELAY_MIN if this parameter is greater than 10
31	**PING_RX_LENGTH**	Duration of the INSTPONG message listening window by the device	1	R/W	L/R	In seconds, from 1 (1s) to PING_RX_LENGTH_MAX	3
32	**PING_RX_DELAY_MIN**	Minimum value of the PING_RX_DELAY parameter	1	R	L/R	In seconds, from 1 (1s) to 255 (255s)	as per MANUFACTURER
33	**PING_RX_LENGTH_MAX**	Maximum value of the PING_RX_LENGTH parameter	1	R	L/R	In seconds, from 1 (1s) to 255 (255s)	as per MANUFACTURER
34	**PING_LAST_EPOCH**	Execution time of the last connectivity test (INSTPING/INSTPONG)	4	R	L/R	EPOCH	00:00
35	**PING_NBFOUND**	Number of different INSTPONG messages received in response to the last connectivity test	1	R	L/R	From 00 to FF	0
36	**PING_REPLY1**	Response 1 received for the last connectivity test (Bigest L7RssiDown)	9	R	L/R	Concatenation of the following fields: L7ConcentId (6 bytes) L7ModemId (1 byte) L7RssiUpstream (1 byte) L7RssiDownstream (1 byte)	0
37	**PING_REPLY2**	…	9	R	L/R	Idem Ping_reply1	0
38	**PING_REPLY3**	…	9	R	L/R	Idem Ping_reply1	0
39	**PING_REPLY4**	…	9	R	L/R	Idem Ping_reply1	0
3A	**PING_REPLY5**	…	9	R	L/R	Idem Ping_reply1	0
3B	**PING_REPLY6**	…	9	R	L/R	Idem Ping_reply1	0
3C	**PING_REPLY7**	…	9	R	L/R	Idem Ping_reply1	0
3D	**PING_REPLY8**	Response 8 received for the last connectivity test, (Weakest L7RssiDown)	9	R	L/R	Idem Ping_reply1	0

3E	**EXECPING_PERIOD E**	Periodic time of execping sending by the device, in months	1	R/ W	L/R	Unsigned int, 0 = deactivated	6

Table 22 : LAN parameters

Other parameters related to Business layers are described in corresponding documents.

6 Informative Appendix B: L6App table as available to publishing date of this document

Updated table available near Wize Alliance, please send an email to contact@wize-alliance.com, with the subject “New L6App ID request”

L6App ID	Application Layer	Description	specifications
$00	APP-WTR	Remote reading of water meters	See..
$01	APP-GAS	Remote reading of gas meters (alternativement $11)	See WIZE - 04 APP_METER-GAS Application Layers - v1.1
$02		*Generic frame (si confirmé)*	
$03		*Shut-off valve (si confirmé)*	
$04 to $1F		Reserved for Metering applications prefered	
$20 to $8F		Reserved for future Use, available from the Wize Alliance that manages their allocation	
$90		M-Bus Full frame	
$91		M-Bus Compact frame	
$92		M-Bus Format frame	
$A0 to $BF		Manufacturer specific frames (maps M-Bus CI-fields)	
$C0 to $FB		Reserved for future Use, available from the Wize Alliance that manages their allocation	
$FC	APP-INSTALL	Identifies Installation Layer; this code is optional as Installation Layer is also notified in C-FIELD (see Wize – 01 LAN Protocol document, see §2.1 Format of LLC- Exchange frames)	See WIZE – 03 Common Application Layer
$FD	APP-ADMIN	Identifies device administration Layer; this code is optional as Administraive Layer is also notified in C-FIELD (see Wize – 01 LAN Protocol document, see §2.1 Format of LLC- Exchange frames)	See WIZE – 03 Common Application Layer
$FE	APP-OPEN	Can be used freely for experimental devices or proprietary application-level protocols.	
$FF	APP-EXT	Reserved for future use as an extension value if multiple bytes are needed	

Table 23:L6App table

END OF DOCUMENT