



DELIVERABLE D 4.2_PROT

Description of the LocON protocols

Dissemination level: public report

**Author(s): Hendrik Millner, Peter Gulden, Maarten Weyn,
Marc Faßbinder, Augusto Casaca**

Work package: WP 4

WP Leader: Fraunhofer

**Partners: Symeo, CIT, CEA-LETI, INOV, Artesis,
Fraunhofer IIS**

Delivery date: 01/04/2009

Short abstract of the deliverable: Deliverable D4.2 stipulates all matters concerning communication of LocON-aware devices with the LocON framework. This includes the incorporation of the TCP/IP protocol as a fully transparent transport protocol. The report describes the data stream between the components and its structure, as well as data packet composition, available packet types, and packet validation.

TABLE OF CONTENTS

Table of contents	2
Table of figures.....	3
Introduction	4
1 Definition of Terms	6
2 Incorporation of the transport and network layer for intranode communication	6
2.1 Detailed description of the TCP/IP stack for LocON	7
2.2 Description of RTP	10
3 LocON Message Types	11
4 Packet Framing.....	12
4.1 LocON Packet Fields.....	12
4.2 Message Data Area.....	13
4.3 Packet Signature.....	13
5 Message Types and Packet Structures.....	13
5.1 LocON position message	13
5.2 LocON extended data message (Message Type 11)	15
5.3 LocON Configuration Message (Message Type 31)	16
5.4 LocON Configuration Response Message (Message Type 32).....	17
5.5 LocON Configuration Read Request (Message Type 33)	17
5.6 LocON Configuration Read Response Message (Message Type 34).....	18

- 5.7 Ping message for monitoring issues (Message Type 41) 19
- 5.8 Ping message response (Message Type 42) 19
- 5.9 Events 20
- 5.10 Update Request (firmware, configuration files) (Message Type 51)..... 20
- 5.11 Update Request Response (Message Type 52) 21
- 5.12 Scaling of High-level and intranode fusion 21
- 6 Multi-Platform Communication 22
- 7 References..... 22

TABLE OF FIGURES

- Figure 1: LocON protocol and datastream in the general project context 4
- Figure 2: Throughput in dependence of the message size 9
- Figure 3: IP Header with TOS Field (Bit 8-16) 10
- Figure 4: RTP header 11

INTRODUCTION

The purpose of this report is the stipulation of the means of communication between the LocON framework LocON-aware sensor nodes and central RTLS connecting to the platform. This document should be referred to when implementing an interface to the LocON framework. It standardizes the network layer, introduces to different means for the transportation and application layer and provides the messages incorporated in the LocON standard protocol. It does not include the physical and link layers because it is envisioned that LOCON messages will be exchanged using WIFI, Ethernet and other communication standards.

Fig. 1 illustrates the importance of the protocol and data stream as central communication mean between the LocON platform and the diverse RTLS in the field and the layers treated in this document.

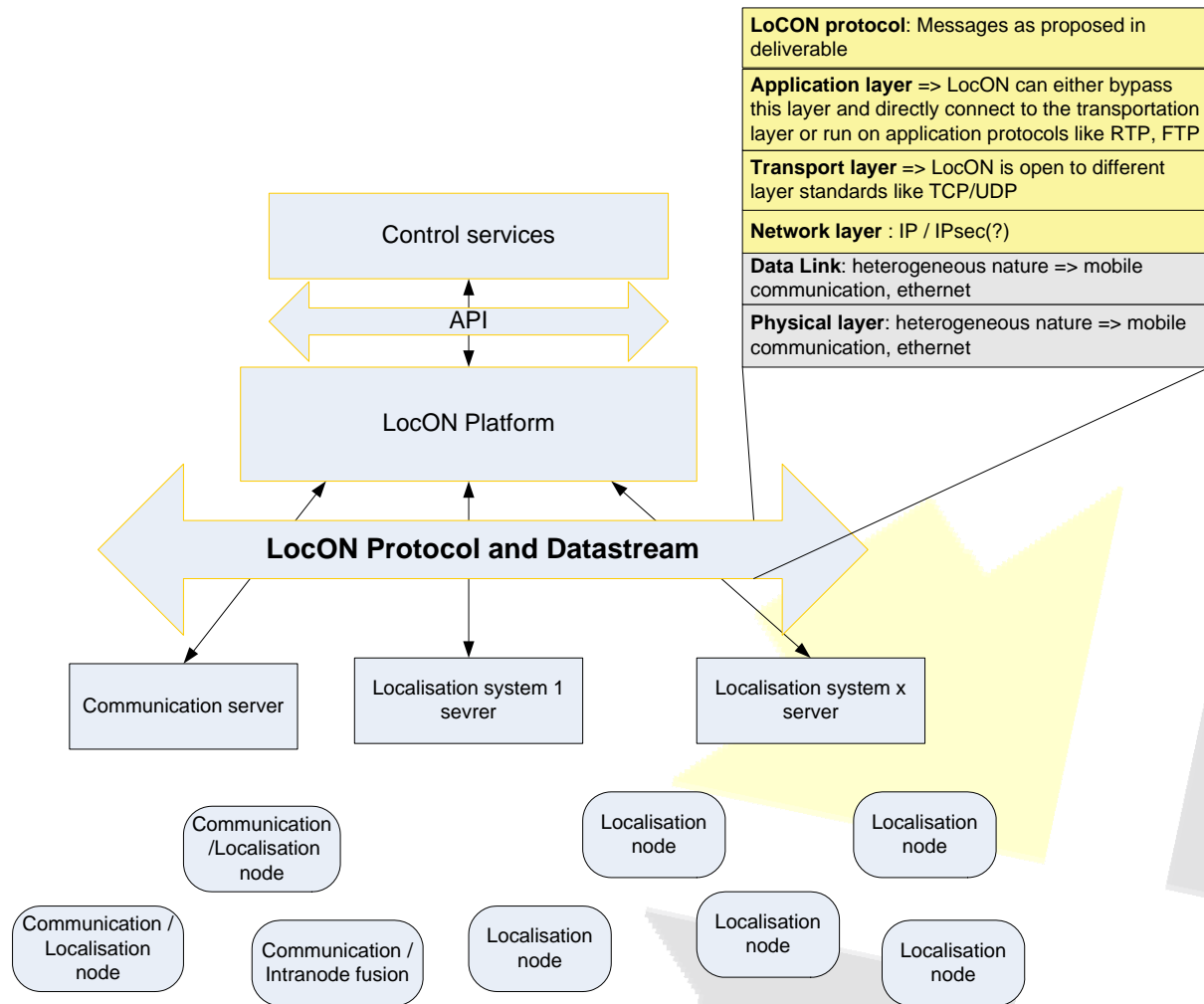


Figure 1: LocON protocol and datastream in the general project context

Several different types of information need to be exchanged. The nodes in the field transmit position information and additional information to the central LocON platform, provide configuration information, accept configuration updates and receive information about firmware updates from the central platform.

The security aspects of the position information in airports and the heterogeneous nature of the physical layer for the information bring up the requirement of messages signatures and encryption for some use cases. Encryption introduces a strong computational overhead and requires more bandwidth. Thus encryption is limited to use cases where the physical layer is considered insecure, e.g. GPRS communication or unknown Ethernet connection. Consequently encryption is an optional feature of the LocON platform, selectable by the end-user. Due to the complexity of the topic a separate document on the LocON encryption and signing methods is provided along with this deliverable.

The proposed protocol reflects the current state of the project and is based on knowledge gathered so far. During the completion of the project the protocol will be open for amendments, changes and additions.

1 DEFINITION OF TERMS

RTLS	Real Time Localization System
TCP/IP	Transmission Control Protocol/Internet Protocol
RTP	Real-Time Transport Protocol
UDP	User Datagram Protocol
FTP	File Transfer Protocol
ECEF	Earth Centered, Earth Fixed coordinate system (global coordinate system)
LLA	Latitude, Longitude, Altitude coordinate system (global coordinate system)
QoS	Quality of Service
GPRS	General Packet Radio Service
WiFi	not an acronym, artificial brand name invented for the Wireless Ethernet Compatibility Alliance
UMTS	Universal Mobile Telecommunications System
EPC	Electronic Product Code
TOS	Type of Service Field (TCP)
AN	Acknowledge Number (TCP)
SN	Sequence Number (TCP)
ISN	Initial Sequence Number (TCP)
SA	Selective Acknowledgement (TCP)
SYN	Synchronize (TCP)
SYN-ACK	Synchronize Acknowledge (TCP)
RFU	Reserved for Future Use (TCP)
NIC	Network Interface Controller (TCP)
V	Version (RTP)
P	Padding (RTP)
X	Extension (RTP)
CC	CSRC count (RTP)
M	Marker (RTP)
PT	Payload Type (RTP)
SN	Sequence Number (RTP)
SSRC	Synchronisation Source (RTP)
CSRC	Contribution Source (RTP)

2 INCORPORATION OF THE TRANSPORT AND NETWORK LAYER FOR INTRANODE COMMUNICATION

The communication layer of the LocON standard should be able to handle several different physical and link layers in a completely transparent way. The platform should not require knowledge about the data channel (Ethernet, WIFI, GPRS, UMTS...). The devices should be addressed only by their unique ID, no routing information etc., which should come from either platform or device. The transparent nature of the protocol in principle allows for different transportation layers as well as long as the transport layer is encapsulated by the gateway. On the network level, the addressing and communication will be IP based while different transportation and application protocols are foreseen.

Several different transport and application protocols have been considered, namely TCP, UDP and RTP/UDP in IP networks. Each of the protocols has individual advantages and disadvantages. TCP avoids loss of packets and detects packet loss. On the one hand this is important for configuration updates where the platform must know if the data packet has been received or not. On the other hand the mechanism of resending data packets until an acknowledgement is received introduces latency in the control application where only current packets are required. Problems have also been reported using a wireless physical layer [4,5]. When the link quality is poor, the loss of packages is sometimes interpreted as overflow, further slowing down the bandwidth [4,5]. Other problems might occur at handover when the handover-time exceeds the timeout limit of the TCP connection [5]. UDP handles this by transmitting data packets without acknowledgement mechanisms. While this is acceptable for positioning messages for control applications, this is not acceptable for positioning messages for tracking applications where the complete track is required for offline evaluation. Of course, UDP is also not recommended for configuration messages where the platform must know if the device received the configuration change command or not. This can be solved by using the application protocol RTP, implemented on top of UDP and providing request for retransmission mechanisms. Additionally acknowledgement mechanisms are implemented for control and command messages [1]. This introduces the cost of using acknowledgement of the transmitted position information.

The LocON protocol is designed to run over either TCP/IP or RTP/UDP/IP. As LocON will use different types of interfaces (air-based and cable-based interfaces) the exact transport protocol to be used is dependent on the interface and localization system under consideration. Consequently the message types defined later in this document are designed independently of the transport and application layer.

2.1 DETAILED DESCRIPTION OF THE TCP/IP STACK FOR LOCON

This chapter describes the TCP/IP protocol, according to the LocON requirements in more detail.

2.1.1 TCP HANDSHAKING

In order to establish a TCP connection, two communication partners in following states are required:

- Passive open: the server is listening on a port for incoming connections.
- Active open: the client is trying to establish a connection to the IP-address and the port from the server that is in a passive open state.

In the following, three steps are described, to establish a TCP connection:

- Active open client sends SYN to the passive open (listening) server.
- The Server responds with SYN-ACK an ISN (Initial Sequence Number).
- The Client acknowledges the SYN-ACK with the ISN from the server.

These three steps complete the handshaking process, it is called “3 way handshaking”.

2.1.2 TCPIP PAKET SIZE

The “Maximum Transfer Unit” is limited to 1500 bytes. Besides the TCP-header (20 Bytes) and the IP-header (20 Bytes) a maximum net packet size of 1460 bytes is left. That is maximum size of a “LocON Message”, so that it is assured that one LocON message is sent in one TCP packet. Thus a segmentation of the messages is avoided.

2.1.3 DATA TRANSFER

The following topics distinguish the connection based on TCP protocol from the streaming protocols UDP and RTP/UDP.

2.1.3.1 Ordered Data Transfer

For the ordered data transfer, the following fields in the TCP header are used:

- Sequence Number (SN): With that number each byte of the data is identified. For transmission of each byte the SN is incremented. The first SN is transmitted during the handshaking process (Initial Sequence Number, ISN).
- Acknowledgement Number (AN): The acknowledgement number informs the transmitter, which next sequence number is expected.
- Window Size: The maximum number of bytes (MTU-2*header) the transceiver is allowed to send, while waiting for an acknowledgement from the receiver.

If the maximum number of bytes is defined the data can be transferred reliably and in order. The data will be retransmitted, if an error during the transmission is asserted.

2.1.3.2 Error Free Data Transfer

The sequence number and acknowledgement number is used to identify and avoid duplicated packets. With the checksum field included in the tcp header, the correctness of the packets can be guaranteed.

2.1.4 TCP-PERFORMANCE

According to [2] and assuming that the message is smaller than 1460 bytes, the theoretical performance of a TCP/IP connection is calculated as follows

$$tp = \frac{ms}{ms \cdot 78 \cdot 84} \cdot \frac{tp_{nic}}{byte}$$

- tp: Throughput in Bytes
- ms: Message Size in Bytes
- tp(NIC): Throughput of the Network Interface Controller in Bytes

Figure 2 shows the dependency between throughput and message size for a 100 MBit/s Ethernet connection and for the message size is smaller than 1460 bytes:

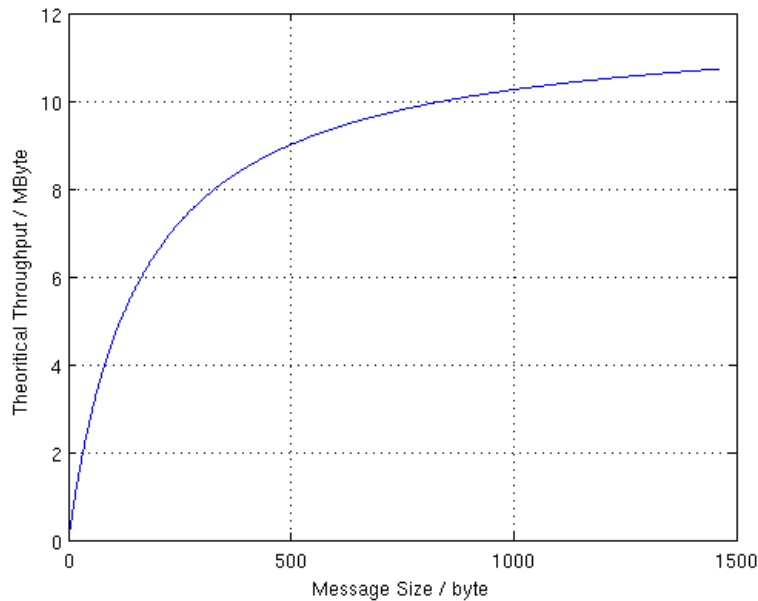


Figure 2: Throughput in dependence of the message size

2.1.5 IMPROVING THE PERFORMANCE OF TCP/IP

To improve the performance of TCP/IP, especially when there is heavy network load, several methods and algorithms exist:

- Slow start: Is used to avoid sending more data than the network is capable of transmitting.
- Fast retransmit/Fast Recovery: These methods are used to handle packet losses in a faster way.
- Selective Acknowledgement: SA are used to get more information about the dataflow from the receiver.
- Urgent mode: With that mode TCP provides a possibility to inform the communication partner that there are “urgent” data in the data stream at the moment. It is the responsibility of the receiver to decide, how to handle these data.

2.1.6 QUALITY OF SERVICE

The *Quality of Service* represents a set of techniques, necessary to control bandwidth reservation, latency and traffic prioritization. In the LocON project the following ones should be investigated in deliverable D4.3 in depth:

- Bandwidth Reservation
- Latency Management
- Traffic Prioritization
- Traffic Shaping
- Network Congestion Avoidance

2.1.7 ToS / DIFFERENTIATED SERVICE FIELD IN THE IP PROTOCOL

The *Type of Service* Field (ToS) is a one Byte long flag-field used by routers and switches to prioritize data. The ToS-byte is structured into three groups: the “Precedence Field”, the “ToS-Field” and one reserved bit¹. The first three bits are used for the precedence and the next four bits for the ToS-Field.

¹ See RFC 1349

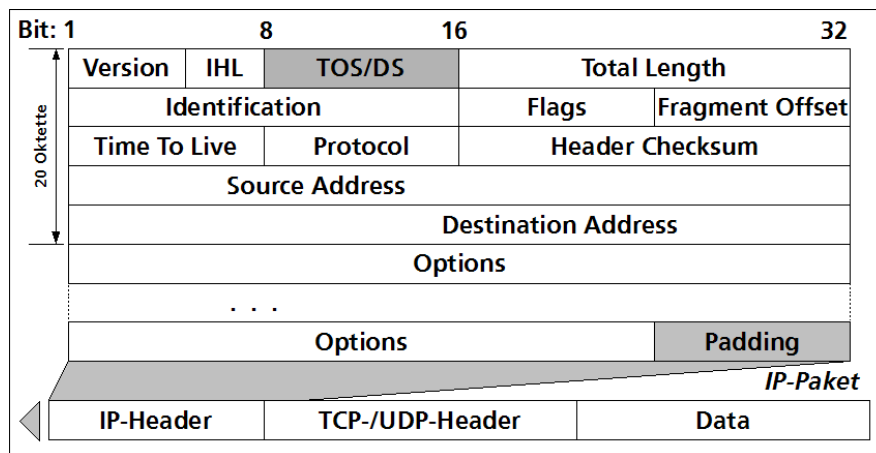


Figure 3: IP Header with TOS Field (Bit 8-16)

The TOS field(Bit 3-7), it must not set more than one bit:

- Bit 3: 0 = Normal Delay, 1 = Low Delay
- Bit4: 0 = Normal Throughput, 1= High Throughput
- Bit5: 0 = Normal Reliability, 1 = High Reliability
- Bits6-7: RFU

2.1.8 QoS METHODS

- *DiffServ*: Differentiated Services (see RFC 2474, RFC 2475). *DiffServ* is a QoS-methodology to prioritize IP packets. This method differs from the others allowing the sender to assign the priority. *DiffServ* uses the first six bits from the TOS byte in the IP header.
- *IntServ*: Integrated Services. "The idea of *IntServ* is that every router in the system implements *IntServ*, and every application that requires some kind of guarantees has to make an individual reservation". In contrast to *DiffServ*, resources will be allocated for every single connection and not for different classes. The disadvantage of *IntServ*, that it is not scalable for larger networks.

2.2 DESCRIPTION OF RTP

The Real Time Transport Protocol (RTP) is used for the transmission of streaming real-time information in an IP network. It is an application protocol on top of UDP. The figure below identifies the fields in the RTP header. In the case of Locon the header extension is not required.

different locating systems position messages are split into standard position messages for simple devices and extended data messages for more complex systems relaying more information to the general platform. Extended data messages can further handle non-recurrent, position independent information like e.g. that the device is overheating. All these messages are transmitted from the RTLS to the LocON platform. The LocON platform on the other hand sends configuration messages to the RTLS (e.g. update rate adjustment) and provides firmware update commands to the nodes in the field. However, the firmware itself is handled outside the LocON protocol, the platform just provides the RTLS with the requirement for an update and the link to the update location. The RTLS can then use any mechanism, including non-transparent methods like FTP to receive the update.

To summarize the messages, there are the following messages types from the RTLS to the LocON platform:

- LocON Position Message
- LocON Extended Data Message
- LocON configuration responses
- LocON update response
- LocON ping response

and these types of messages from the LocON platform to the RTLS

- LocON Configuration requests
- LocON firmware update and specific configuration files update requests
- LocON ping requests

The messages will be explained in detail in the corresponding sections of chapter 4.

4 PACKET FRAMING

Packets are sent binary coded using the full 8 bit alphabet. Framed packets have the following characteristics:

- Frames are either single data packets frames or consist of multiple data packets of the different types.
- Each packet contains a signature for means of certification and validation of its contents and the sender identity. This prevents from a modification of packet data and from faking an identity.
- The framing structure allows for an implementation of data encryption, if this is seen regarded necessary. Encryption is not planned to be implemented during the initial LocON project.
- The protocol byte order is Big-Endian. Thus the value 256 is represented by the two bytes [0x01, 0x00].

The packet structure is as follows:

ProtocolVersion	DataLength	Data Area	Signature
1 byte	2 bytes (uint16)	variable length	externally defined

4.1 LOC ON PACKET FIELDS

The message header contains the following fields:

- *ProtocolVersion* Version of protocol (see 4.1.1)
- *DataLength* Length of upcoming data field (see 4.1.2)
- *DataArea* Actual data field with payload (see 4.2)
- *Signature* Signature to validate packet contents and sender ID (see 4.3)

4.1.1 PROTOCOL VERSION

ProtocolVersion is an unsigned 8 bit integer value determining the version number of the protocol implementation. The version number of the protocol as it is described in this document is 0 (zero). This value ensures compatibility after future protocol changes.

4.1.2 DATA LENGTH

The field *DataLength* is an unsigned 16 bit integer value, specifying the length in bytes of the data area it is followed by.

4.2 MESSAGE DATA AREA

The data area can contain several different messages, e.g. multiple position messages from different technologies attached to one device node, or combinations of position messages and extended data. The data area has the following structure:

- *MsgType* Message type of upcoming message (see 4.2.1).
- *MessageData* Data associated with current message type (see 4.2.2).

Header/ previous message	MsgType	MessageData	Footer/ next message
	1 byte	variable length	

4.2.1 MESSAGE TYPE

The message type is an unsigned 8 bit integer value describing the structure and contents of the data area of the message. Following message types represent valid LocON messages:

Message Type No.	Description
1	Position message with position expressed in ECEF
2	Position message with position expressed in LLA
3	Position message with position expressed in local Cartesian coordinates
11	Extended data message
31	Configuration message
32	Configuration response message
33	Configuration read request
34	Configuration read request response
41	Ping message
42	Ping message response
51	Firmware/ configuration update request
52	Firmware/ configuration update request response

4.2.2 MESSAGE DATA

This area of the message holds the actual information associated with the current message type. Its length and structure depends on are message dependent. The structures of valid packet types are described in chapter 5.

4.3 PACKET SIGNATURE

Packet signatures and security aspects have been moved to the second document with compound treatment of the complex issue.

5 MESSAGE TYPES AND PACKET STRUCTURES

5.1 LOC ON POSITION MESSAGE

There are the message type 1, 2 and 3. The data area of this message contains the following fields:

- *Device ID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *Timestamp* The timestamp of the location data (see 5.1.3)
- *Coordinates 1, 2, 3* Computed coordinates of the device (see 5.1.4)
- *QoL* Quality of Location value (see 5.1.5) (a value which conveys the quality of the localization to the LocON platform)

MsgType	DeviceID length	DeviceID	Timestamp	Coordinate 1	Coordinate 2	Coordinate 3	QoL	Footer/next MsgType
	1 byte	variable length	4 bytes (uint32)	4 bytes (int64)	4 bytes (int64)	4 bytes (int64)	2 bytes (uint16)	

5.1.1 DEVICE ID LENGTH

Since there are multiple technologies available, each with its own unique addresses (device ID) and address lengths, LocON supports variable device id lengths. The field *DeviceID length* states the length in bytes of the upcoming field *DeviceID*. Only the lower 4 bits (0b0000xxxx) are used for the value, enabling a device ID length from 0 to 15 bytes. The upper 4 bits are reserved for future use.

5.1.2 DEVICE ID

The *DeviceID* is an assumptive unique identifier of the sending localization device. It is generated by the manufacturer of the device. Upon address collisions the LocON platform can increase the address length by one and add a random number to make the address unique within the LocON framework. The largest device ID is probably be an EPCGlobal number which needs 13 bytes.

5.1.3 TIMESTAMP

The timestamp is an unsigned 32 bit integer value with the following format: hhmmsssttt, where

- hh represents hour of day,
- mm represents minute of hour,
- ss represents second of minute, and
- ttt represents milliseconds of second.

It is generated at the time the position of the device is computed. The timestamp is an unsigned 32 bit integer value.

5.1.4 DEVICE COORDINATES

The contents of the three coordinates are dependent on the message type of the packet. Coordinate transformations between all available coordinate systems is done by the LocON platform to ensure the correctness of implementation. The LocON platform also manages reference points for local coordinate systems (left- or right-handed coordinate system, orientation matrix, specification in T4.4 and WP3). Planned coordinates are:

5.1.4.1 LocON platform Message type 1

The position is expressed in ECEF coordinate system.

Coordinate 1 contains the X-coordinate, coordinate 2 the Y-coordinate and coordinate 3 the Z-coordinate of the device position in the ECEF coordinate system. All three values are signed 32 bit integer values and scaled in millimeters.

5.1.4.2 Message type 2

The Position is expressed in LLA coordinate system.

Coordinate 1 contains the latitude of the device position, coordinate 2 the longitude and coordinate 3 the altitude. All three values are signed 64 bit integer values. Latitude and longitude are scaled in 10^8 , thus we have a resolution of roughly one millimeter. Altitude is scaled in millimeters.

5.1.4.3 Message type 3

The Position is expressed in local Cartesian coordinate system.

Coordinate 1 contains the X-coordinate, coordinate 2 the Y-coordinate and coordinate 3 the Z-coordinate of the device position in a local, reference-dependant coordinate system. All three values are signed 32 bit integer values and scaled in millimeters.

Coordinate transformations between all available coordinate systems is done by the LocON platform to ensure the correctness of implementation. The LocON platform also manages reference points for local coordinate systems (left- or right-handed coordinate system, orientation matrix, specification in T4.4 and WP3).

5.1.5 QoL

The quality of location value gives a standardized hint upon the accuracy of the current position. The value is an unsigned 16 bit integer value and scaled in millimeter. It contains the square root of the (measured, estimated or synthesized) variance of the device position as a scalar value (standard degression/ deviation).

5.2 LOC ON EXTENDED DATA MESSAGE (MESSAGE TYPE 11)

Extended data messages transport data other than the device position, like velocity, bearing, or sensor measurements, like temperature, button pressed events.

The message contains the following fields:

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *Timestamp* Timestamp of the location data (see 5.1.3)
- *DataLength* Length of upcoming data field (see 5.2.1)
- *DataArea* An accumulation of data fields with extended information (see 5.2.2)

MsgType	DeviceID length	DeviceID	Timestamp	DataLength	DataArea	Footer/ next MsgType
	1 byte	variable length	4 bytes (uint32)	1 byte	variable length	

5.2.1 DATA LENGTH

The field *DataLength* is an unsigned 8 bit integer value determining the length in bytes of the upcoming data area.

5.2.2 DATA AREA

The data area contains one or more data fields of the following structure:

- *DataTypeID* Data type identifier (see 5.2.2.1)
- *DataValue* Corresponding data (see 5.2.2.2)

Header/ previous data field	DataTypeID	DataValue	Footer/ next data field
	1 byte	variable length	

The LocON platform will log this data and provide a datamining service to the upper layers (business logic, fusion, host applications).

5.2.2.1 Data Type ID

DataTypeID is an unsigned 8 bit integer value determining the type of the data field. It can be used to identify the information in the upcoming field *DataValue* and thus is needed to decode the information stored within it.

5.2.2.2 Data Value

DataValue contains the corresponding data. Its length and structure depend on the data type ID. As an example, a *DataTypeID* value of 1 could represent the temperature of the device, with a 2 byte field *DataValue* representing a signed 16 bit integer. These correspondences have to be supplied by an external information file. The nature and structure of the external information files will be specified during task 4.4 of the LocON project.

5.3 LOC ON CONFIGURATION MESSAGE (MESSAGE TYPE 31)

Configuration messages enable the LocON platform to modify certain parts of the configuration of the LocON devices and their localization subcomponents. Additionally the scaling between intranode and high-level fusion will be controlled through these messages.

The message contains the following fields:

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *DataLength* Length of upcoming data area (see 5.2.1)
- *DataArea* An accumulation of configuration IDs and values (see 5.3.1)

MsgType	DeviceID length	DeviceID	DataLength	DataArea	Footer/ next MsgType
	1 byte	variable length	1 byte	variable length	

5.3.1 DATA AREA

The data area contains one or more configuration IDs and values of the following structure:

- *ConfigurationID* Configuration property to write to (see 5.3.1.1)
- *ConfigurationValue* Corresponding value to write to configuration (see 5.3.1.2)

Header/ previous data field	ConfigurationID	ConfigurationValue	Footer/ next data field
	1 byte	variable length	

5.3.1.1 Configuration ID

ConfigurationID is an unsigned 8 bit integer value determining the configuration property of the device to modify. The association of IDs with properties must be supplied within an external information file. Configuration properties can be read-only or subject to special authorization. Authorization can be obtained by using a specific certificate to sign the message in its signature part.

5.3.1.2 Configuration Value

ConfigurationValue contains the corresponding value to write to the configuration property. Its length and structure depends on the configuration ID. For example, a *ConfigurationID* value of 1 could represent the measurement delay of the device, with a 2 byte field *ConfigurationValue* representing an unsigned 16 bit integer, holding the delay value in milliseconds. These correspondences have to be supplied by an external information file.

5.4 LOC ON CONFIGURATION RESPONSE MESSAGE (MESSAGE TYPE 32)

The response message will provide feedback on the configuration modification request(s).

The message contains the following fields:

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *DataLength* Length of upcoming data area (see 5.2.1)
- *DataArea* An accumulation of configuration IDs and return codes (see 5.4.1)

MsgType	DeviceID length	DeviceID	DataLength	DataArea	Footer/ next MsgType
	1 byte	variable length	1 byte	variable length	

5.4.1 DATA AREA

The *DataArea* contains one or more configuration IDs and return codes of the following structure:

- *ConfigurationID* Configuration property which was to be written to (see 5.3.1.1)
- *ReturnCode* Corresponding return code with result of the modification request (see 5.4.1.1)

Header/ previous data field	ConfigurationID	ReturnCode	Footer/ next data field
	1 byte	1 byte	

5.4.1.1 Return Code

ReturnCode is an unsigned 8 bit integer value specifying the action taken by the device upon the request to change the corresponding configuration ID. A return code of 0 (zero) represents the successful modification of the specified configuration ID, other return codes are error values. Return codes in the range from 32 to 255 can be used for device specific error states.

Return code	Description
0	Successful configuration modification
1	Unknown configuration ID
2	Denied attempt to modify read-only value
3	Denied due to insufficient authorization

5.5 LOC ON CONFIGURATION READ REQUEST (MESSAGE TYPE 33)

Configuration read requests enable the LocON platform to read the values of certain configuration properties of LocON devices and their localization subcomponents.

The message contains the following fields:

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *DataLength* Length of upcoming data area (see 5.2.1)
- *DataArea* An accumulation of configuration IDs to be read (see 5.5.1)

MsgType	DeviceID length	DeviceID	DataLength	DataArea	Footer/ next MsgType
	1 byte	variable length	1 byte	variable length	

5.5.1 DATA AREA

The data area contains one or more configuration IDs of the following structure:

- *ConfigurationID* Configuration property to read value from (see 5.5.1.1)

Header/ previous data field	ConfigurationID	Footer/ next data field
	1 byte	

5.5.1.1 Configuration ID

ConfigurationID is an unsigned 8 bit integer value determining the configuration property of the device to be read. The association of IDs with properties must be supplied by an external information file. Configuration property read requests can be subject to special authorization. Authorization can be obtained by using a specific certificate to sign the message in its signature part.

5.6 LOCON CONFIGURATION READ RESPONSE MESSAGE (MESSAGE TYPE 34)

The response message will provide the values of the requested configuration properties.

The message contains the following fields:

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *DataLength* Length of upcoming data area (see 5.2.1)
- *DataArea* An accumulation of configuration IDs, read configuration values and return codes (see 5.6.1)

MsgType	DeviceID length	DeviceID	DataLength	DataArea	Footer/ next MsgType
	1 byte	variable length	1 byte	variable length	

5.6.1 DATA AREA

The data area contains one or more configuration IDs, read configuration values and return codes of the following structure:

- *ConfigurationID* Configuration property which was to be read from (see 5.5.1.1)
- *ConfigurationValue* Corresponding value to write to configuration (see 5.6.1.1)
- *ReturnCode* Corresponding return code with result of the modification request (see 5.6.1.2)

Header/ previous data field	ConfigurationID	ReturnCode	ConfigurationValue	Footer/ next data field
	1 byte	1 byte	variable length	

5.6.1.1 Configuration Value

ConfigurationValue contains the corresponding configuration property value if the read request was successful. Its length and structure depend on the configuration ID. The correspondences of configuration ID and configuration value structure have to be supplied by an external information file. In case of an error, the returned configuration value consists of null bytes (0x00).

5.6.1.2 Return Code

ReturnCode is an unsigned 8 bit integer value specifying the action taken by the device upon the request to read the corresponding configuration ID. A return code of 0 (zero) represents the successful extraction of the specified configuration ID's value, other return codes are error values:

Return code	Description
0	Successfully read configuration property
1	Unknown configuration ID
2	Denied due to insufficient authorization

5.7 PING MESSAGE FOR MONITORING ISSUES (MESSAGE TYPE 41)

Ping messages are small messages to "ping" the device and check its state.

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *Timestamp* Timestamp of packet (see 5.1.3)
- *SequenceNumber* A request number which should be sent back from the device (see 5.7.1)

MsgType	DeviceID length	DeviceID	Timestamp	SequenceNumber	Footer/ next MsgType
	1 byte	variable length	4 bytes (uint32)	1 byte	

5.7.1 SEQUENCE NUMBER

SequenceNumber is an unsigned 8 bit integer value, which identifies one ping request. This value should be sent back by the device in order to finish the ping request.

5.8 PING MESSAGE RESPONSE (MESSAGE TYPE 42)

The response message will contain the timestamp applied by the device, the same cycle counter as in the corresponding ping message and two 16 bit integer values to give feedback on the system state, e.g. low power, error state, error number.

- *DeviceID length* Length of upcoming device ID (see 5.1.1)
- *DeviceID* The System ID and the Device ID make the device unique (see 5.1.2)
- *Timestamp* Timestamp of packet applied by the device (see 5.1.3)
- *TimeDelta* Time difference between the timestamp of the corresponding ping message and the device's timestamp (see 5.8.1) of the device
- *SequenceNumber* The request number of the corresponding ping message (see 5.8.2)
- *SystemState* State of system (see 5.8.3)

MsgType	DeviceID length	DeviceID	Timestamp	TimeDelta	SequenceNumber	SystemState	Footer/ next MsgType
	1 byte	variable length	4 bytes (uint32)	2 bytes (int16)	1 byte	2 bytes (uint16)	

5.8.1 TIME DELTA

TimeDelta is a signed 16 bit integer value which represents the difference in milliseconds between the timestamp of the received ping request and the device time. Positive values indicate that the received timestamp is in the past, relative to the device clock. If the framework clock and the device clock are synchronized, this value should be positive and represent the communication latency.

5.8.2 SEQUENCE NUMBER

SequenceNumber is an unsigned 8 bit integer value which is sent back as it was received. It is used to identify the ping reply to the ping request.

5.8.3 SYSTEM STATE

SystemState is an unsigned 16 bit integer value containing the device state. A device state of 0 (zero) means that everything is in order, different states can be used to give information about an erroneous state, e.g. low battery power, sensor fault, etc.

5.9 EVENTS

Events will not be implemented within the scope of the project. Extended data messages can serve the purpose of informing the platform of specific system states.

5.10 UPDATE REQUEST (FIRMWARE, CONFIGURATION FILES) (MESSAGE TYPE 51)

Currently the consensus is that firmware updates are triggered by the LocON platform. This will be clarified in T4.4. After receiving the update request the device then connects to FTP via MW->RTLS update.

The update request message requests the device to update its firmware or its own proprietary config file. The digital certificate of the LocON platform must be deposited on the device and authorized to update the specified object.

■ <i>DeviceID length</i>	Length of upcoming device ID (see 5.1.1)
■ <i>DeviceID</i>	The System ID and the Device ID make the device unique (see 5.1.2)
■ <i>UpdateType</i>	Type of object to be updated (see 5.10.1)
■ <i>LinkLength</i>	Length of upcoming link (see 5.10.2)
■ <i>Link</i>	TCP/IP link towards updated object (see 5.10.3)

MsgType	DeviceID length	DeviceID	UpdateType	LinkLength	Link	Footer/ next MsgType
	1 byte	variable length	1 byte	1 byte	variable length	

5.10.1 UPDATE TYPE

The *UpdateType* is an unsigned 8 bit integer value to determine the object to be updated. A value of 0 (zero) implies an update of the device firmware, a value of 1 an update of the device configuration. Further object IDs are possible.

Update type	Description
0	Device firmware
1	Device configuration

5.10.2 LINK LENGTH

LinkLength is an unsigned 8 bit integer value stating the length in bytes of the upcoming string Link.

5.10.3 LINK

Link is a string consisting of unsigned 8 bit integer values representing the characters. It contains the URL of the new firmware or configuration for the device or object. This can be an FTP link like

ftp://10.0.0.3/device1/firmware.bin for example. The update process may break the transparency of the TCP/IP layer for LocON usage.

5.11 UPDATE REQUEST RESPONSE (MESSAGE TYPE 52)

Message type 52

Upon finishing an update process or after occurrence of an error during the process, a response message is sent to the LocON platform:

■	<i>DeviceID length</i>	Length of upcoming device ID (see 5.1.1)
■	<i>DeviceID</i>	The System ID and the Device ID make the device unique (see 5.1.2)
■	<i>UpdateType</i>	Type of object which was to be updated (see 5.10.1)
■	<i>ReturnCode</i>	Return code of performed operation (see 5.11.1)
■	<i>NewVersion</i>	Version of object after update (attempt) (see 5.11.2)

MsgType	DeviceID length	DeviceID	UpdateType	ReturnCode	NewVersion	Footer/ next MsgType
	1 byte	variable length	1 byte	1 byte	4 bytes (uint32)	

5.11.1 RETURN CODE

ReturnCode is an unsigned 8 bit integer value offering information about the outcome of the update request. A value of 0 (zero) states a successful update, other values are error codes.

Return code	Description
0	Successfully updated object
1	Object update failed
2	Not authorized to update
3	URL not reachable
4	URL contains no valid update for this object

5.11.2 NEW VERSION

NewVersion is an unsigned 32 bit integer value indicating the version of the object after the update process. In case the update failed with an error, *NewVersion* contains the version of the object from before the update attempt.

5.12 SCALING OF HIGH-LEVEL AND INTRANODE FUSION

A unique feature of the project is the scaling of intranode and high-level fusion. Local nodes incorporating several different RTLS like GPS and LPR or inertia sensors handle fusion in a local way to reduce the data traffic. The high-level fusion performance though increases when more sensor information is used. Consequently the amount of information transmitted must be adapted according to available bandwidth and the needs of the high-level data fusion.

The scaling will be done using extended data messages and configuration messages. The exact messages will be defined depending on the requirement of the high-level fusion in the progress of the project.

Currently, three states of communication are planned from the intranode to the high-level fusion:

1. Standard: Only position messages with the object position are transmitted
2. Extended: Extended data messages with more information about the fusion process are transmitted along with the position. Additional information could be various states of the internal Kalman filter for example
3. Raw: All raw data is sent along with the position using extended data messages.

The platform uses the message 31, configuration message, to change between the three types of information provided according to the estimated bandwidth and the required input of the high-level data fusion. Of course the exact content to be transmitted largely depends on the nature of the RTLS parts present in the mobile unit and must be exchanged between the high-level data fusion and the node when the node connects to the platform for the first time. Details will be worked out in WP 5 and 6.

6 MULTI-PLATFORM COMMUNICATION

Projecting success of the LocON approach, multiple LocON platforms might need to interface. For example, the LocON platform of an airport might interchange information with a LocON platform at an intermodal terminal on freight leaving the airport heading for the intermodal terminal. Defining messages for the multi-platform communication should be handled only after the platform is more defined and we have more implementation and field experience with the single LocON platform.

7 REFERENCES

- [1] RTP Protocol- rfc1889, Network Working Group, Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, January 1996
- [2] TCP/IP Illustrated, Pearson Education Inc, W.Richard Stevens
- [3] Technik der IP-Netze, Carl Hanser Verlag, Anatol Badach, Erwin Hoffmann
- [4] TCP Performance Issues over Wireless Links, George Xylomenos et al, IEEE COMMUNICATIONS MAGAZINE, VOLUME 39, NUMBER 4, 2001, PP. 52–58
- [5] Open Issues on TCP for Mobile Computing, Vassilis Tsaoussidis, Ibrahim Matta