

**Stateless [RFC5444](#)-based Dynamic Link Exchange Protocol (DLEP)
draft-rogge-stateless-rfc5444-dlep-00**

Abstract

This document provides material for the discussion in the MANET WG about the Dynamic Link Exchange Protocol (DLEP). This document reflects the authors' thoughts about how a stateless DLEP protocol compliant with [RFC5444](#) could look like.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 9, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	4
2.	Terminology	5
3.	Assumptions	6
4.	Protocol Overview and Functioning	7
4.1.	Routers and Radios	7
4.2.	Information Base Overview	8
4.3.	Signaling Overview	9
4.3.1.	Automatic Radio Interface Discovery	10
4.3.2.	Setup Radio	10
4.3.3.	Interface Updates	10
4.3.4.	Local Destinations	11
4.3.5.	Neighbor Updates	11
5.	Protocol Parameters	12
5.1.	Port Number and Multicast Address	12
5.2.	DLEP-Radio Parameters	12
5.3.	DLEP-Router Parameters	13
6.	Local Radio Information Base	15
6.1.	Local Interface Set	15
6.2.	Interface Neighbor Set	15
6.3.	Lost Neighbor Set	16
6.4.	Interface Data Set	16
6.5.	Neighbor Data Set	17
7.	Configuration Information Base	18
7.1.	Interface Configuration Set	18
7.2.	Local Destination Set	18
8.	Local Router Information Base	20
8.1.	Radio Interface Configuration Set	20
8.2.	Destination Configuration Set	20
9.	Network Information Base	22
9.1.	Discovered Interface Set	22
9.2.	Network Local Destination Set	22
9.3.	Network Interface Data Set	23
9.4.	Network Neighbor Set	24
9.5.	Network Neighbor Data Set	24
10.	Local Radio Information Base Changes	26
10.1.	New Radio Interface Neighbor	26
10.2.	Lost Radio Interface Neighbor	27
10.3.	Radio Interface Status	28
11.	Packets and Messages	29
11.1.	DLEP Messages	29
11.1.1.	DLEP Message Orders	30
11.2.	Message TLVs	30
11.3.	Address Block TLVs	31
12.	Interface Update Order	36
12.1.	Interface Update Order Generation	36
12.2.	Interface Update Order Processing	37

13.	Local Destination Order	39
13.1.	Local Destination Order Generation	39
13.2.	Local Destination Order Processing	40
14.	Neighbor Update Order	42
14.1.	Neighbor Update Order Generation	42
14.2.	Neighbor Update Order Discarding	44
14.3.	Neighbor Update Order Processing	44
15.	Setup Radio Order	46
15.1.	Setup Radio Order Generation	46
15.2.	Setup Radio Order Processing	47
16.	Setup Destination Order	48
16.1.	Setup Destination Order Generation	48
16.2.	Setup Destination Order Processing	49
17.	IANA Considerations	51
17.1.	Expert Review: Evaluation Guidelines	51
17.2.	Message Types	51
17.3.	Message-Type-Specific TLV Type Registries	51
17.4.	Message TLV Types	52
17.5.	Address Block TLV Types	52
18.	Further work	53
19.	Security Considerations	54
20.	References	55
20.1.	Normative References	55
20.2.	Informative References	55
Appendix A.	RFC5444 -DLEP Message Examples	56
A.1.	Interface Update Order Example	56
A.2.	Neighbor Update Order Example	59
A.3.	Setup Radio Order Example	61
A.4.	Setup Destination Order	62
A.5.	Local Destination Order Example	65
Author's Address	66

1. Introduction

This document is not intended to go forward on its own, but provides input for the discussion in the MANET WG about DLEP, in particular for (1) allowing to use preexisting [RFC5444](#) parsers and generators with DLEP, and (2) to simplify the protocol by proposing a "stateless" protocol specification of DLEP. It is the opinion of the author of this document that delivering this input for the discussion may help to clarify some proposals that have been made on the MANET mailing list.

Dynamic Link Exchange Protocol (DLEP, as defined in [[dlep02](#)]) is a proposal for a cross-layer protocol between a layer-2 entity like a radio and a layer-3 router to transport layer-2 metric, statistic and status data from the radio to the router. In addition, it allows the router to control and configure aspects of the radio, such as radio status, channel or link speed.

DLEP does not communicate via radio links, it is only used locally. Therefore DLEP does not need to focus on payload efficiency as other protocols of the MANET working group.

A DLEP-capable radio works as a transparent bridge for the data-plane. DLEP is used to transport meta-information like link metrics, detected neighbors and configuration options between radio and router via a separate control plane.

This document does not discuss the Link Characteristic configuration and Credit Granting parts of [[dlep02](#)], it rather concentrates on a [[RFC5444](#)] compatible packet format, so a standard compliant generic parser/generator code can be used for this.

Instead of copying the functionality of the [[dlep02](#)] state machine and session handling, this document tries to reproduce the functionality with a stateless design similar to the mechanisms in [[RFC6130](#)] and [[olsrv2](#)].

If this approach is adopted by the Working Group, parts of this document could be folded back into the [draft-ietf-manet-dlep](#).

2. Terminology

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document are to be interpreted as described in [\[RFC2119\]](#).

The terminology introduced in [\[RFC5444\]](#) and [\[RFC5497\]](#), including the terms "packet", "message", "Address Block", "TLV Block", "TLV", "address", "address prefix", and "address object" are to be interpreted as described therein.

Additionally, this document uses the following terminology and notational conventions:

Order - This specification uses a single [\[RFC5444\]](#) Message Type to transmit several different kinds of information. The Order of the Message specifies the type of information that is contained in the Message.

DLEP-router - a router which runs DLEP, also called Server in [\[dlep02\]](#).

DLEP-radio - a radio (or modem) which runs DLEP with one or more external layer-2 interfaces (e.g. WLAN interfaces). It also has a connection to attach a DLEP-router. A DLEP-radio is also called Client in [\[dlep02\]](#).

Radio Interface - an external interface of a DLEP-radio.

Radio Interface ID - an unique 6 octet identifier for an radio interface of a DLEP-radio has. A DLEP-radio MAY use MAC-addresses as interface IDs. There MAY be special IDs for defining the receiving part of a multicast or broadcast transmission.

Destination - A MAC-address of a device attached a DLEP-radio that can be reached through the radio interface of the DLEP-radio. This can be a unicast, multicast or broadcast MAC.

Neighbor - A Destination that can be reached through a remote radio interface.

Network - In the context of this document, a network is a group of layer-2 radio devices talking to each other.

3. Assumptions

This protocol keeps most assumptions as defined in [[dlep02](#)].

[dlep02] and this specification assume that participating DLEP-radios appear to the DLEP-routers as a transparent bridge - specifically, the assumption is that the destination MAC address for data-plane traffic in any frame emitted by the DLEP-router should be the MAC address of the next-hop router or end-device, and not the MAC address of any of the intervening devices (like DLEP-radios).

[dlep02] and this specification assume that security on the protocol (e.g. authentication of DLEP-radio/router, encryption of data- and/or control-plane traffic) is dealt with by the underlying transport mechanism for the [[RFC5444](#)] packets. As an alternative this specification would suggest using the signature support for [[RFC5444](#)] itself (as defined in [[RFC6622](#)]).

[dlep02] and this specification assume that the DLEP-radios will be only connected to a single DLEP-router, not multiple devices. It might be investigate if this assumption can be relaxed later.

This specification assumes that the DLEP-radio is capable to determine the Radio Interface IDs of the other side of the radio communication from the Ethernet destination MAC-address of packets incoming from the local DLEP-router. It also assumes that a DLEP-radio is capable to determine the source MAC-address of packets incoming through the DLEP-radios interface.

This specification assumes that the control-plane of the DLEP-radio, which is used by this protocol, is separated by the data-plane from each other by some mechanism. Protocol traffic between DLEP-router and DLEP-radio will not interfere with data forwarded by the radio.

[4.](#) Protocol Overview and Functioning

The objectives of this protocol are for each DLEP-radio to:

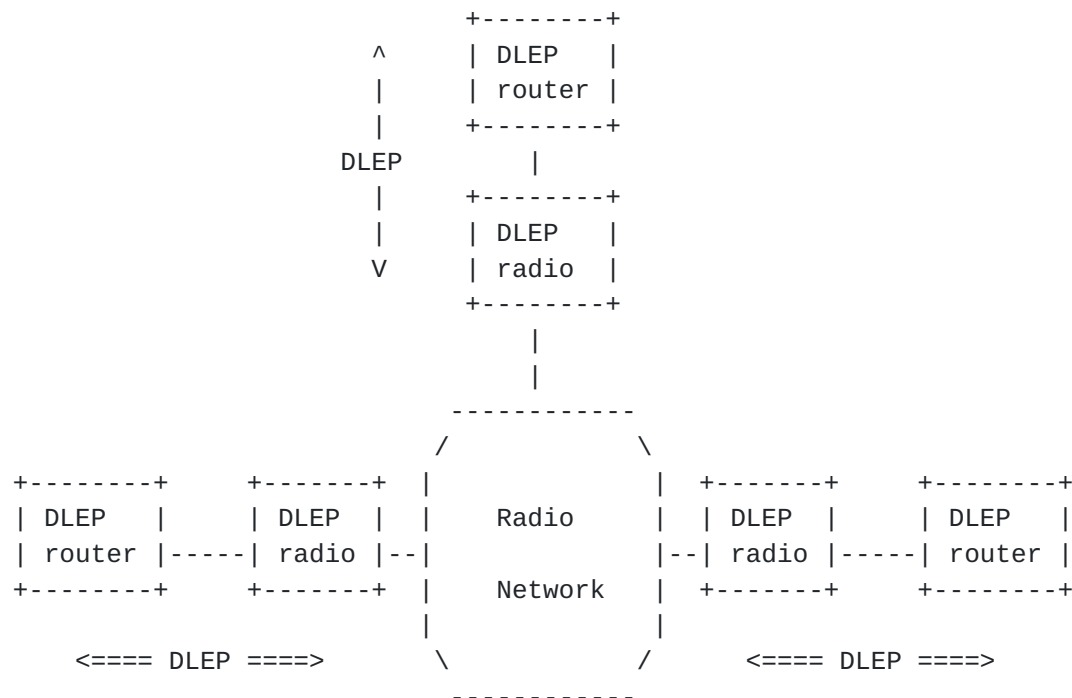
- o Announce its presence to the attached DLEP-router.
- o Provide each DLEP-router with Neighbor information present in its radio.
- o Provide each DLEP-router with network metrics and statistics.
- o Provide each DLEP-router with Neighbor radio link metrics and statistics.

The objectives of this protocol are for each DLEP-router to:

- o Automatically discover the presence of DLEP-radio interfaces in the local network.
- o Get a list of all known Neighbors of each DLEP-radio interface.
- o Get a list of network metrics and statistics.
- o Get a list of Neighbor metrics and statistics.
- o Being able to configure the radios local interface.

[4.1.](#) Routers and Radios

This specification describes two different kinds of protocol instances, DLEP-radios and DLEP-routers.



Example of a radio network with three DLEP-radios and -routers.

Figure 1

DLEP-radios are typically data-link layer forwarding devices. These devices are connected to a local router via a short distance and high speed link, often an Ethernet connection. They contain enough resources to provide extra services to the router, and they are often embedded computer systems on their own.

DLEP-radios should split the connection between radio and router into separate control- and data-planes, to be able to bridge the data-plane directly to a radio interface without being influenced by the protocol traffic between DLEP-radio and the DLEP-router on the control-plane.

DLEP-routers can detect and configure DLEP-radios via the described protocol. The router can use the protocol to gather interface and Neighbor information and access link-layer metrics and statistics in a hardware independent way, even with the radio hardware not built into the router.

4.2. Information Base Overview

Both DLEP-radio and DLEP-router maintain the protocol state using Information Bases, described in the following section. Each Information Base consists of a number of Protocol Sets. Each

Protocol Set contains a number of Protocol Tuples.

An implementation of this protocol may maintain this information in the indicated form, or in any other organization that offers access to this information. In particular, note that it is not necessary to remove Protocol Tuples from Protocol Sets at the exact time indicated, only to behave as if the Protocol Tuples were removed at that time.

The Local Radio Information Base is maintained in the DLEP-radio and included in Interface Update and Neighbor Update Orders to the DLEP-router. Information in the Network Information Base on the DLEP-router is determined by received Interface Update and Neighbor Update Orders from the DLEP-radio. Such information has a limited duration in which it is considered valid. This duration is determined from the VALIDITY_TIME TLV in the Order in which the information is received, which in turn is set by the DLEP-radio that originated the Order, using its corresponding DLEP-radio parameter INTERFACE_UPDATE_HOLD_TIME and NEIGHBOR_UPDATE_HOLD_TIME.

The Local Router Information Base is maintained in the DLEP-router and included in Setup Radio and Setup Destination Orders to the DLEP-radio. Information in the Configuration Information Base on the DLEP-radio originates from Setup Radio and Setup Destination Orders from the DLEP-router. Such information has a limited duration in which it is considered valid. This duration is determined from the VALIDITY_TIME TLV in the both Orders in which the information is received, which in turn is set by the DLEP-router that originated the Orders, using its corresponding DLEP-router parameter SETUP_RADIO_HOLD_TIME and SETUP_DESTINATION_HOLD_TIME.

Information in the Configuration Information Base on the DLEP-radio is included in the Local Destination Order to the DLEP-router. Such information has a limited duration in which it is considered valid. This duration is determined from the VALIDITY_TIME TLV in the Local Destination Order in which the information is received, which in turn is set by the DLEP-router that originated the Local Destination Order, using its corresponding DLEP-radio parameter LOCAL_DESTINATION_HOLD_TIME.

4.3. Signaling Overview

This protocol contains a signaling mechanism for maintaining the Configuration Information Base and the Network Information Base.

[4.3.1.](#) Automatic Radio Interface Discovery

As in [[dlep02](#)], this specification contains an automatic discovery mechanism, that allows a DLEP-router to detect the presence of one or multiple DLEP-radio interfaces.

To enable the DLEP-router to do this, this specification uses a single Order, the Interface Update Order.

This Order combines the Peer Discovery, Peer Offer, Peer Update (ACK), Peer Termination (ACK), and Heartbeat messages of [[dlep02](#)]. Because of a validity time based mechanism, this obsoletes the internal state machine of [[dlep02](#)].

For the content of the Interface Update Order, see [Section 4.3.3](#)

[4.3.2.](#) Setup Radio

This specification allows the DLEP-router to configure DLEP-radio interface settings. In this document, there are two Orders to configure the DLEP-radio, the Setup Radio Order and the Setup Destination Order.

The Setup Radio Order allows the DLEP-router to set the status of the DLEP-radio interface to UP or DOWN.

The Setup Destination Order adds a local Destination to a DLEP-radio interface, with IP-prefixes attached to this Destination (if known). This data is used by some radios to transport the MAC-address and IP prefixes of their attached Destinations to each other within the layer-2 protocol, to avoid an Address Resolution Protocol (ARP) exchange over the radio.

[4.3.3.](#) Interface Updates

DLEP-radios transmit a list of its interfaces and data about the interface's status, description and network metrics and statistics with the Interface Update Order.

The Order contains the Radio Interface ID that is described in the Order, the status of the interface (up or down) and an optional textual description of this radios interface (e.g. Company XYZ Software Defined Radio ABC line 1). It also contains interface specific metrics and statistics.

This data will be sent by DLEP-radios in regular intervals, at least once every INTERFACE_UPDATE_INTERVAL.

4.3.4. Local Destinations

DLEP-radios transmit the currently configured Destination of the local router back to the router to acknowledge the configuration with the Local Destination Order.

The Order contains the MAC-address of the locally configured Destination, the its local Radio Interface ID and the IP-prefixes of the Destination (if known).

This data will be sent by DLEP-radios in Local Destination Orders in regular intervals, at least once every LOCAL_DESTINATION_INTERVAL.

4.3.5. Neighbor Updates

Modern IP-capable radios are able to collect lots of PHY- and link-layer data about their connections to other radios in range and the Destinations behind them.

The Neighbor Update Order contains the known configuration (MAC- and IP-addresses), status and attributes (e.g. link statistics and metrics) of a radio connection between the local DLEP-radio interface and a Neighbor.

This data will be sent by DLEP-radios in Neighbor Update Orders in regular intervals, at least once every NEIGHBOR_UPDATE_INTERVAL.

5. Protocol Parameters

5.1. Port Number and Multicast Address

While [[dlep02](#)] is likely to be designed to be transport independent, DLEP over UDP will be a common use case, so DLEP MUST specify a default UDP port number and a Linklocal Multicast address to be used. If an existing port and Multicast address can be reused or not is out of scope for this specification.

5.2. DLEP-Radio Parameters

There are several parameters that MUST be set on a DLEP-radio.

INTERFACE_UPDATE_INTERVAL:

The maximum time between the transmission of two successive Interface Update Orders for the same DLEP-radio interface, possibly modified by jitter as specified in [[RFC5148](#)].

LOCAL_DESTINATION_INTERVAL:

The maximum time between the transmission of two successive Local Destination Orders for the same DLEP-radio interface, possibly modified by jitter as specified in [[RFC5148](#)].

NEIGHBOR_UPDATE_INTERVAL:

The maximum time between the transmission of two successive Neighbor Update Orders for the same DLEP-radio interface, possibly modified by jitter as specified in [[RFC5148](#)].

INTERFACE_UPDATE_HOLD_TIME:

Used as the Value in the VALIDITY_TIME Message TLV included in all Interface Update Orders. It is then used by the DLEP-router receiving such an Order to indicate the validity of the information taken from that Order and recorded in the receiving DLEP-router's Network Information Base.

LOCAL_DESTINATION_HOLD_TIME:

Used as the Value in the VALIDITY_TIME Message TLV included in all Local Destination Orders. It is then used by the DLEP-router receiving such an Order to indicate the validity of the information taken from that Order and recorded in the receiving DLEP-router's Network Information Base.

NEIGHBOR_UPDATE_HOLD_TIME:

Used as the Value in the VALIDITY_TIME Message TLV included in all Neighbor Update Orders. It is then used by the DLEP-router receiving such an Order to indicate the validity of the information taken from that Order and recorded in the receiving DLEP-router's Network Information Base.

DEFAULT_INTERFACE_STATE:

Defines the default status of the interfaces of a DLEP-radio. It MUST be UP or DOWN.

The following constraints apply to these DLEP-radio parameters:

- o INTERFACE_UPDATE_INTERVAL > 0
- o LOCAL_DESTINATION_INTERVAL > 0
- o NEIGHBOR_UPDATE_INTERVAL > 0
- o INTERFACE_UPDATE_HOLD_TIME > INTERFACE_UPDATE_INTERVAL
- o LOCAL_DESTINATION_HOLD_TIME > LOCAL_DESTINATION_INTERVAL
- o NEIGHBOR_UPDATE_HOLD_TIME > NEIGHBOR_UPDATE_INTERVAL

[5.3.](#) DLEP-Router Parameters

There are several parameters that MUST be set on a DLEP-router.

SETUP_RADIO_INTERVAL:

The maximum time between the transmission of two successive Setup Radio Orders, possibly modified by jitter as specified in [\[RFC5148\]](#).

SETUP_LOCAL_DESTINATION_INTERVAL:

The maximum time between the transmission of two successive Setup Destination Orders, possibly modified by jitter as specified in [\[RFC5148\]](#).

SETUP_RADIO_HOLD_TIME:

Used as the Value in the VALIDITY_TIME Message TLV included in all Setup Radio Orders. It is then used by the DLEP-radio receiving such an Order to indicate the validity of the information taken

from that Order and recorded in the receiving DLEP-radio's Information Bases.

SETUP_LOCAL_DESTINATION_HOLD_TIME:

Used as the Value in the VALIDITY_TIME Message TLV included in all Setup Destination Orders. It is then used by the DLEP-radio receiving such an Order to indicate the validity of the information taken from that Order and recorded in the receiving DLEP-radio's Information Bases.

The following constraints apply to these DLEP-router parameters:

- o SETUP_RADIO_INTERVAL > 0.
- o SETUP_RADIO_HOLD_TIME > SETUP_RADIO_INTERVAL.
- o SETUP_LOCAL_DESTINATION_INTERVAL > 0.
- o SETUP_LOCAL_DESTINATION_HOLD_TIME > SETUP_LOCAL_DESTINATION_INTERVAL.

6. Local Radio Information Base

The DLEP-radio maintains a Local Radio Information Base that records a list of locally defined or collected data about the DLEP-radio's interfaces and its connected neighbors.

The Local Radio Information Base is not modified by signaling. If a DLEP-radio's interface configuration changes, then the Local Radio Information Base **MUST** reflect these changes. This **MAY** also result in signaling to advertise these changes.

6.1. Local Interface Set

A DLEP-radio's Local Interface Set records the existing local interfaces, their default status and their description. It consists of Local Interface Tuples, one per existing local interface:

(LI_radio_interface_id, LI_default_status, LI_description)

where:

LI_radio_interface_id is the Radio Interface ID of the local radio interface.

LI_default_status is the default status of the radio interface if not configured by the DLEP-router.

LI_description is a textual description of the radio interface with a maximum of 80 octets without ending zero byte. It **MAY** be **NONE**.

LI_radio_interface_id is the unique key of this set, there **MUST NOT** be two tuples with the same unique key in the set.

6.2. Interface Neighbor Set

A DLEP-radios's Interface Neighbor Set records the known other DLEP-radios in communication range and the addresses of the neighbors reachable through this radios. It consists of Interface Neighbor Tuples, one per known neighbor of a local interface:

(IN_radio_interface_id, IN_neighbor_interface_id, IN_mac_address, IN_ipv4_prefixes, IN_ipv6_prefixes)

where:

IN_radio_interface_id is a six octet unique Radio Interface ID of the local interface for this neighbor.

IN_neighbor_interface_id is a Radio Interface ID of the neighbors DLEP-radio's interface, it MAY be NONE (or a special ID) for a broadcast or multicast neighbor.

IN_mac_address is the six octet MAC address of the neighbor.

IN_ipv4_prefixes is a list of known IPv4 prefixes of this neighbor. It MAY be empty.

IN_ipv6_prefixes is a list of known IPv6 prefixes of this neighbor. It MAY be empty.

(IN_radio_interface_id, IN_neighbor_interface_id, IN_mac_address) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

6.3. Lost Neighbor Set

When a radio interface's neighbor becomes unreachable, a DLEP-radio's Lost Neighbor Set records this for at least INTERFACE_UPDATE_HOLD_TIME. It consists of Lost Neighbor Tuples, one per lost neighbor of a local radio interface:

(LN_radio_interface_id, LN_neighbor_interface_id, LN_mac_address)

where:

LN_radio_interface_id is a six octet unique Radio Interface ID of the local interface for the lost neighbor.

LN_neighbor_interface_id is a six octet unique Radio Interface ID of the lost neighbors DLEP-radio's interface, it MAY be NONE for a broadcast or multicast neighbor.

LN_mac_address is the six octet MAC address of the lost neighbor.

There MUST NOT be two tuples with the same data in the set.

6.4. Interface Data Set

A DLEP-radio's Interface Data Set records attributes (e.g. statistics or metrics) of its local radio interfaces and their attached networks. It consists of Interface Data Tuples:

(ID_radio_interface_id, ID_data_type, ID_data_value)

where:

ID_radio_interface_id is a Radio Interface ID of a local radio interface.

ID_data_type is the type of data stored. It MUST be an extension type of the NETWORK_DATA Address TLV.

ID_data_value is the data corresponding to this tuple's key. Its format is defined in the NETWORK_DATA Address TLV values.

(ID_radio_interface_id, ID_data_type) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

6.5. Neighbor Data Set

A DLEP-radio's Neighbor Data Set records attributes (e.g. statistics or metrics) of a connection to a Neighbor. It consists of Neighbor Data Tuples:

(ND_radio_interface_id, ND_neighbor_interface_id, ND_mac_address, ND_data_type, ND_data_value)

where:

ND_radio_interface_id is a Radio Interface ID of the local interface for this neighbor.

ND_neighbor_interface_id is a Radio Interface ID of the neighbors DLEP-radio's interface, it MAY be NONE for a broadcast or multicast neighbor.

ND_mac_address is the six octet MAC address of the neighbor.

ND_data_type is the type of data stored. It MUST be an extension type of the NEIGHBOR_DATA Address TLV.

ND_data_value is the data corresponding to this tuples key. Its format is defined in the NEIGHBOR_DATA Address TLV values.

(ND_radio_interface_id, ND_neighbor_interface_id, ND_mac_address, ND_data_type) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

7. Configuration Information Base

The DLEP-radio maintains a Configuration Information Base that records the configuration settings of the DLEP-router for each interface.

The Configuration Information Base is modified by Setup Radio and Setup Destination Orders from the DLEP-router. If a Configuration Information Base changes, then the local configuration of the interface **MUST** reflect this change. This **MAY** also result in signaling to advertise these changes.

7.1. Interface Configuration Set

A DLEP-radio's Interface Configuration Set records the status of the local DLEP-radio interfaces as configured by the DLEP-router. It consists of Interface Configuration Tuples, one for each local interface:

(IC_radio_interface_id, IC_radio_status, IC_time)

where:

IC_radio_interface_id is a Radio Interface ID of a local radio interface.

IC_radio_status is the status of the local radio interface. It **MUST** be "up" or "down".

IC_time is the time when this tuple **MUST** be removed.

IC_radio_interface_id is the unique key of this set, there **MUST NOT** be two tuples with the same unique key in the set.

7.2. Local Destination Set

A DLEP-radio's Local Destination Set records the known local destinations of local radio interfaces set by the connected DLEP-router. It consists of Local Destination Tuples:

(LD_radio_interface_id, LD_mac_address, LD_ipv4_prefixes,
LD_ipv6_prefixes, LD_time)

where:

LD_radio_interface_id is a Radio Interface ID of a local radio interface.

LD_mac_address is the MAC address of the Destination.

LD_ipv4_prefixes is a list of IPv4 prefixes of the Destination.
It MAY be empty.

LD_ipv6_prefixes is a list of IPv6 prefixes of the Destination.
It MAY be empty.

LD_time is the time when this tuple MUST be removed.

(LD_radio_interface_id, LD_mac_address) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

8. Local Router Information Base

The DLEP-router maintains a Local Router Information Base that records the radio interface status and the local Destinations for a DLEP-radio.

The Local Router Information Base is not modified by signaling. If a DLEP-router's configuration changes, then the Local Router Information Base **MUST** reflect these changes. This **MAY** also result in signaling to advertise these changes.

8.1. Radio Interface Configuration Set

A DLEP-router's Radio Interface Configuration Set records the radio interface status, up or down, to be configured on a DLEP-radio. It consists of Radio Interface Configuration Tuples, up to one for each DLEP-radio interface connected to the DLEP-router:

(RIC_radio_interface_id, RIC_status)

where:

RIC_radio_interface_id is the Radio Interface ID of an attached DLEP-radio.

RIC_status is the desired status of the DLEP-radio interface.
MUST be UP or DOWN.

RIC_radio_interface_id is the unique key of this set, there **MUST NOT** be two tuples with the same unique key in the set.

8.2. Destination Configuration Set

A DLEP-router's Destination Configuration Set records the local Destinations to be configured on a DLEP-radio interface. It consists of Destination Configuration Tuples, up to one for each Destination of a DLEP-radio interface connected to the DLEP-router:

(DC_radio_interface_id, DC_mac_address, DC_ipv4_prefixes,
DC_ipv6_prefixes)

where:

DC_radio_interface_id is the Radio Interface ID of an attached DLEP-radio interface.

DC_mac_address is the MAC-address of Destination for the DLEP-radio interface.

DC_ipv4_prefixes is a list of IPv4 prefixes of Destination for the DLEP-radio interface. MAY be empty.

DC_ipv6_prefixes is a list of IPv6 prefixes of Destination for the DLEP-radio interface. MAY be empty.

(DC_radio_interface_id, DC_mac_address) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

9. Network Information Base

A DLEP-router maintains a Network Information Base that records all collected information transmitted by the attached DLEP-radios. This consists both of information about the settings of the local interfaces of the DLEP-radios, metrics and statistics about the network attached to the local interfaces and metrics and statistics about neighbors.

The Network Information Base is modified by Interface Update, Local Destination, and Neighbor Update Orders from the DLEP-radio. If the Network Information Base changes, this MAY trigger changes in the DLEP-router.

9.1. Discovered Interface Set

A DLEP-router's Discovered Interface Set records the attached Radio Interface IDs, the status and description of the DLEP-radio interfaces. It consists of Discovered Interface Tuples, one for each discovered DLEP-radio interface:

(DI_radio_interface_id, DI_radio_status, DI_description, DI_time)

where:

DI_radio_interface_id is the Radio Interface ID of the discovered DLEP-radio interface.

DI_radio_status is the current status of the interface. MUST be UP or DOWN.

DI_description is a textual description of the radio interface of up to 80 characters without zero byte ending.

DI_time is the time when this tuple MUST be removed.

DI_radio_interface_id is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

9.2. Network Local Destination Set

A DLEP-router's Network Local Destination Set records the DLEP-radio interface's configured local Destinations. It consists of Network Local Destination Tuples, one for each DLEP-radio interface's local Destination:

(NLD_radio_interface_id, NLD_mac_address, NLD_ipv4_prefixes,
NLD_ipv6_prefixes, NLD_time)

where:

NLD_radio_interface_id is the Radio Interface ID of the local Destination's radio interface.

NLD_mac_address is the six octet MAC-address of the local Destination.

NLD_ipv4_prefixes is a list of IPv4 prefixes configured for the local Destination, MAY be empty.

NLD_ipv6_prefixes is a list of IPv6 prefixes configured for the local Destination, MAY be empty.

NLD_time is the time when this tuple MUST be removed.

(NLD_radio_interface_id, NLD_mac_address) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

[9.3.](#) Network Interface Data Set

A DLEP-router's Network Interface Data Set records a list of attributes known about a DLEP-radio's interface (e.g. interface statistics and radio attributes). It consists of Network Interface Data Tuples:

(NID_radio_interface_id, NID_data_type, NID_data_value, NID_time)

where:

NID_radio_interface_id is the Radio Interface ID of the DLEP-radio's interface.

NID_data_type is the type of data stored. It MUST be an extension type of the NETWORK_DATA Address TLV.

NID_data_value is the data corresponding to this tuples key. Its format is defined in the NETWORK_DATA Address TLV values.

NID_time is the time when this tuple MUST be removed.

(NID_radio_interface_id, NID_data_type) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

9.4. Network Neighbor Set

A DLEP-router's Network Neighbor Set records the known configuration and status of each neighbor of a DLEP-radio interface. It consists of Network Neighbor Tuples, one for each neighbor or a radio interface:

```
(NN_radio_interface_id, NN_neighbor_interface_id, NN_mac_address,  
NN_status, NN_ipv4_prefixes, NN_ipv6_prefixes, NN_time)
```

where:

NN_radio_interface_id is the Radio Interface ID of the DLEP-radio interface.

NN_neighbor_interface_id is the Radio Interface ID of the neighbors DLEP-radio interface.

NN_mac_address is the MAC-address of the neighbor.

NN_status is the status of the neighbors radio interface. MUST be UP or DOWN.

NN_ipv4_prefixes is a list of IPv4 prefixes of the neighbor. MAY be EMPTY.

NN_ipv6_prefixes is a list of IPv6 prefixes of the neighbor. MAY be EMPTY.

(NN_radio_interface_id, NN_neighbor_interface_id, NN_mac_address) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

9.5. Network Neighbor Data Set

A DLEP-router's Network Neighbor Data Set records a list of attributes known about a DLEP-radio interface's neighbor (e.g. link statistics and metrics). It consists of Network Neighbor Data Tuples:

```
(NND_radio_interface_id, NND_neighbor_interface_id,  
NND_mac_address, NND_data_type, NND_data_value, NND_time)
```

where:

NND_radio_interface_id is the Radio Interface ID of the DLEP-radio interface.

NND_neighbor_interface_id is the Radio Interface ID of the neighbors DLEP-radio interface.

NND_mac_address is the MAC-address of the neighbor.

NND_data_type is the type of data stored. It MUST be an extension type of the NEIGHBOR_DATA Address TLV.

NND_data_value is the data corresponding to this tuples key. Its format is defined in the NEIGHBOR_DATA Address TLV values.

NND_time is the time when this tuple MUST be removed.

(NND_radio_interface_id, NND_neighbor_interface_id, NND_mac_address, NND_data_type) is the unique key of this set, there MUST NOT be two tuples with the same unique key in the set.

10. Local Radio Information Base Changes

The Local Radio Information Base MUST be updated in response to changes in the DLEP-radio's local interface configuration.

A DLEP-radio MAY transmit Interface Update and Neighbor Update Orders in response to these changes.

10.1. New Radio Interface Neighbor

If a new Neighbor is discovered on a DLEP-radio interface, this MUST result in changes to the Local Radio Information Base

Define

- o neighbor_local_interface_id := the local Radio Interface ID of the former neighbor.
- o neighbor_remote_interface_id := the Radio Interface ID of the neighbors DLEP-radio interface, MAY be NONE or a special ID for multicast and broadcast Neighbors.
- o neighbor_mac_address := the MAC-address of the neighbor.
- o neighbor_ipv4_prefixes := the list of known IPv4 prefixes of the neighbor.
- o neighbor_ipv6_prefixes := the list of known IPv6 prefixes of the neighbor.

Then change the information base as follows:

1. Remove all Lost Neighbor Tuple with:
 - * LN_radio_interface_id = neighbor_local_interface_id AND
 - * LN_neighbor_interface_id = neighbor_remote_interface_id AND
 - * LN_mac_address = neighbor_mac_address.
2. Add an Interface Neighbor Tuple with:
 - * IN_radio_interface_id := neighbor_local_interface_id.
 - * IN_neighbor_interface_id := neighbor_remote_interface_id.
 - * IN_mac_address := neighbor_mac_address.

- * IN_ipv4_prefixes := neighbor_ipv4_prefixes.

- * IN_ipv6_prefixes := neighbor_ipv6_prefixes.

The change MAY also trigger a Neighbor Update Order.

10.2. Lost Radio Interface Neighbor

If an existing neighbor is lost on a DLEP-radio interface, this MUST result in change of the Local Radio Information Base.

Define

- o neighbor_local_interface_id := the local Radio Interface ID of the former neighbor.
- o neighbor_remote_interface_id := the Radio Interface ID of the former neighbor's radio interface.
- o neighbor_mac_address := the MAC-address of the neighbor.

Then change the information base as follows:

1. Remove an existing Interface Neighbor Tuple with:

- * IN_radio_interface_id = neighbor_local_interface_id AND
- * IN_neighbor_interface_id = neighbor_remote_interface_id AND
- * IN_mac_address = neighbor_mac_address.

2. Remove all Neighbor Data Tuples with:

- * ND_radio_interface_id = neighbor_local_interface_id AND
- * ND_neighbor_interface_id = neighbor_remote_interface_id AND
- * ND_mac_address = neighbor_mac_address.

3. Add a Lost Neighbor Tuple with:

- * LN_radio_interface_id := NI_radio_interface_id.
- * LN_neighbor_interface_id := NI_neighbor_interface_id.
- * LN_mac_address := NI_mac_address.

* LN_time := current + NEIGHBOR_UPDATE_HOLD_TIME.

The change MAY also trigger an Interface Update Order.

[10.3.](#) Radio Interface Status

The DLEP-radio interface status is determined by two sources, the default radio interface status and the status configured by a DLEP-router. Determine the actual status as follows:

Define radio_interface_id as the ID of the radio interface which status will be determined.

If there is an Interface Configuration Tuple with

- o IC_radio_interface_id = radio_interface_id,

set the radio status to IC_radio_status.

Otherwise, find the Local Interface Tuple with

- o LI_radio_interface_id = radio_interface_id

and set the radio interface status to LI_default_status.

11. Packets and Messages

The packet and message format used by this protocol is defined in [[RFC5444](#)], which is used with the following considerations:

- o This protocol specifies one Message Type, the DLEP message.
- o A DLEP message MAY use any combination of Message Header options specified in [[RFC5444](#)].
- o DLEP messages MUST NOT be forwarded, i.e., a <msg-hop-limit>, if present, MUST have the value 1.
- o DLEP messages MAY be included in multi-message packets as specified in [[RFC5444](#)].
- o Received DLEP messages MUST be parsed in accordance with [[RFC5444](#)]. A DLEP message that is not in conformance with [[RFC5444](#)] MUST be discarded without being processed.
- o This protocol specifies five new Address Block TLVs and one new Message TLV.
- o This protocol uses one Message TLVs defined in [[RFC5497](#)], the VALIDITY_TIME TLV.

This specified protocol defines and owns the DLEP Message Type (see [Section 17](#)). Thus, as specified in [[RFC5444](#)] this protocol generates and transmits all DLEP messages, receives all DLEP messages and is responsible for determining whether and how each DLEP message is to be processed.

11.1. DLEP Messages

A DLEP Message MUST contain:

- o An address length of 6, meaning msg-addr-length (as defined in [[RFC5444](#)]) must be 5.
- o Exactly one Message TLV with Type = VALIDITY_TIME as defined in [[RFC5497](#)].
- o Exactly one Message TLV with Type = ORDER.

Each Address in a DLEP message MUST:

- o have zero or one TLV with Type = ADD_ADDRESS and the same Extension Type.

- o have zero or one TLV with Type = NETWORK_DATA and the same Extension Type.
- o have zero or one TLV with Type = NEIGHBOR_DATA and the same Extension Type.
- o have zero or one TLV with Type = DESCRIPTION.

DLEP Messages that do not obey this rules MUST be discarded without further processing.

A DLEP Message MAY contain:

- o Other Message TLVs.
- o One or more Address Blocks, each with an associated Address Block TLV Block, which MAY contain other Address Block TLVs.

[11.1.1.](#) DLEP Message Orders

This protocol uses several messages with different semantics. Instead of allocating a Message-Type for each of them, this specification use the ORDER Message TLV to define an extension type for the single Message-Type it uses. (see Table 2)

[11.2.](#) Message TLVs

The ORDER Message TLV MUST be used in all DLEP Messages. A message MUST NOT contain more than one ORDER TLV.

Type	Value	Value
	Length	
ORDER	1	Specifies the ORDER of a DLEP Message, which in turn defines the context for the rest of the Message.

Table 1: ORDER TLV definition

There are five types of ORDER defined in this document.

ORDER value	Originator	Description
INTERFACE_UPDATE	DLEP-radio	Updates the data of one or more radio interfaces.
LOCAL_DESTINATION	DLEP-radio	Publishes the locally set Destinations of the radio interface.
NEIGHBOR_UPDATE	DLEP-radio	Updates the data of one or more neighbors of a radio interface.
SETUP_RADIO	DLEP-router	Configures a the interface settings of one or more radio interfaces.
SETUP_DESTINATION	DLEP-router	Adds local Destinations to one or more DLEP-radio interfaces.

Table 2: Types of ORDERS in DLEP Messages

[11.3.](#) Address Block TLVs

The RADIOIF_STATUS TLV is used in all three Orders. An Address of a DLEP Message MUST NOT contain more than one RADIOIF_STATUS TLV.

Type	Value Length	Value
RADIOIF_STATUS	1	Specifies that the interface is UP or DOWN.

Table 3: RADIOIF_STATUS TLV definition

The DESCRIPTION TLV is used in the in the Interface Update Order. An Address of an Interface Update Order MUST NOT contain more than one DESCRIPTION TLV.

Type	Value	Value
	Length	
DESCRIPTION	1-80	Specifies a human readable ASCII identifier for a DLEP-radio interface without added zero byte.

Table 4: DESCRIPTION TLV definition

The ADD_ADDRESS TLV is used in the Local Destination, the Neighbor Update and the Setup Destination Order. An Address of a DLEP Message MUST NOT contain multiple ADD_ADDRESS TLVs with the same Extension Types.

Type	Ext-Type	Value	Value
		Length	
ADD_ADDRESS	MAC	x*6 octets	A list of MAC-addresses attached to the address object.
ADD_ADDRESS	IPv4_PREFIX	x*5 octets	A list of IPv4 prefixes attached to the address object. Every IPv4 address is followed by a one octet prefix length (0-32) in the list.
ADD_ADDRESS	IPv6_PREFIX	x*17 octets	A list of IPv6 prefixes attached to the address object. Every IPv4 address is followed by a one octet prefix length (0-128) in the list.

Table 5: ADD_ADDRESS TLV definition

The NETWORK_DATA TLV is used in the Interface Update Order. An Interface Update Order Address MUST NOT contain more than one NETWORK_DATA TLV with the same Type Extension.

Type	Ext-Type	Value Length	Value
NETWORK_DATA	NETWORK_ID	1-16	Binary identifier of a radio network (e.g. BSSID).
NETWORK_DATA	NETWORK_DESCR	1-80	ASCII identifier of a radio network without added zero byte (e.g. SSID)
NETWORK_DATA	SUPPORTED_RATES	x*8	Array of supported data rates of the network an interface is attached to as an array of 8 octet unsigned integers in bit/s.
NETWORK_DATA	RESOURCES	2	The first octet contains an estimate of the resources left, between 0 (no resources left) and 100 (all resources left). The second octet is 0 if the radio has limited resources or 1 if the resources are unlimited and/or the resources are currently being recharged by an external source.
NETWORK_DATA	LAST_ACTIVE	4	Time since the last data was sent or received over a radio interface as an unsigned integer in milliseconds.
NETWORK_DATA	FREQUENCY	8	Mid frequency of a radio interface channel as an unsigned integer in Hertz.

NETWORK_DATA	BANDWIDTH	8	Amount of spectrum (frequency range) of a radio interface channel as an unsigned integer in Hertz.
+-----+-----+-----+-----+			

Table 6: NETWORK_DATA TLV definition

The NEIGHBOR_DATA TLV is used in the Neighbor Update Order. An Network Update Order Address MUST NOT contain more than one NEIGHBOR_DATA TLV with the same Type Extension.

Type	Ext-Type	Value Length	Value
NEIGHBOR_DATA	RELATIVE_LQ	1	Estimate of the quality of a link between 0 (worst) and 100 (best).
NEIGHBOR_DATA	MAXIMUM_DATARATE	8+8	Maximum possible link speed as 8 octet unsigned integer in bits/s. First value is receiving speed, second is transmitting speed.
NEIGHBOR_DATA	CURRENT_DATARATE	8+8	Current link speed as 8 octet unsigned integer in bits/s. First value is receiving speed, second is transmitting speed.
NEIGHBOR_DATA	TRAFFIC	8+8	Number of bytes exchanged with a neighbor since link went up as 8 octet unsigned integer in bytes. First value is received bytes, second is transmitted bytes.

NEIGHBOR_DATA	PACKETS	8+8	Number of IP packets exchanged with a neighbor since link went up as 8 octet unsigned integer in bytes. First value is received packets, second is transmitted packets.
NEIGHBOR_DATA	FRAMES	8+8	Number of link-layer frames exchanged with a neighbor since link went up as 8 octet unsigned integer in bytes. First value is received frames, second is transmitted frames.
NEIGHBOR_DATA	TX_RETRIES	8	Number of link-layer retransmissions of the same IP packet since link went up as unsigned integer.
NEIGHBOR_DATA	TX_FAILS	8	Number of permanent link-layer transmission failures of an IP packet since the link went up as unsigned integer.
NEIGHBOR_DATA	LAST_ACTIVE	4	Time since the last data was exchanged as an unsigned integer in milliseconds.

Table 7: NEIGHBOR_DATA TLV definition

12. Interface Update Order

The Interface Update Order is sent by DLEP-radios in regular intervals, at least once every `INTERFACE_UPDATE_INTERVAL`.

An Interface Update Order contains the description, status and attributes (e.g. interface metrics and statistics) of one or more DLEP-radio interfaces. This allows the DLEP-router to receive the whole data generated from the radio about each interface of the DLEP-radio in an atomic way.

The Interface Update Order **MUST** be generated by the DLEP-radio periodically. While it is allowed to split the Interface Updates for multiple DLEP-radio interfaces into more than one Interface Update Order, each Local Interface Tuple **MUST** be sent once in an Interface Update Order within `INTERFACE_UPDATE_INTERVAL`.

The Interface Update Order includes a list of Radio Interface IDs as address objects.

For each address, the Interface Update Order **MUST** contain the `RADIOIF_STATUS` Address TLV, which specifies if the radio interface is up or down.

For each address, it **MAY** contain one `DESCRIPTION` TLV with a textual description of the radio interface.

For each address, it **MAY** contain one `NETWORK_DATA` Address TLV for each Extension-Type, which will specify the known statistics and metrics of the radio interface.

12.1. Interface Update Order Generation

Each DLEP-radio **MUST** generate Interface Update Orders according to the specification in this section.

Define `interface_set` as a subset of the Local Interface Set, which **MUST** contain at least one Local Interface Tuple.

Each Interface Update Order **MUST** contain the following additional elements:

- o A Message TLV with Type := `ORDER` and Value := `INTERFACE_UPDATE`.
- o A Message TLV with Type := `VALIDITY_TIME` (as specified in [[RFC5497](#)]) with Value := `INTERFACE_UPDATE_HOLD_TIME` (encoded as specified in [[RFC5497](#)]).

- o For each Local Interface Tuple in `interface_set`, add `LI_radio_interface_id` as an address object (as defined in [\[RFC5444\]](#)) and apply the following steps:
 - 1. If `LI_description` is not `NONE`, add an Address TLV with Type := `DESCRIPTION` and Value := `LI_description`.
 - 2. Add an Address TLV with Type := `RADIOIF_STATUS`. If there is an Interface Configuration Tuple with `IC_radio_interface_id` = `LI_radio_interface_id`, set Value := `IC_radio_status`. Otherwise, set Value := `LI_default_status`.
 - 3. For each Interface Data Tuple with `ID_radio_interface_id` = `LI_radio_interface_id`, add an Address TLV with Type := `NETWORK_DATA`, Extension Type := `ID_data_type` and Value := `ID_data_value`.

12.2. Interface Update Order Processing

An Interface Update Order is processed on the DLEP-router as follows:

- 1. Define `validity_time` as the Value of the Message TLV with Type = `VALIDITY_TIME`.
- 2. For each address object `interface_id` that has an Address TLV with Type = `RADIOIF_STATUS`,
 - 1. Define `interface_status` as the Value of the Address TLV with Type = `RADIOIF_STATUS`.
 - 2. If there is a Discovered Interface Tuple with `DI_radio_interface_id` = `interface_id`, remove it (there should be a maximum of one such tuples).
 - 3. Add a Discovered Interface Tuple with:
 - + `DI_radio_interface_id` := `interface_id`.
 - + `DI_radio_status` := `interface_status`.
 - + `DI_description` := `EMPTY`.
 - + `DI_time` := `current_time` + `validity_time`.
 - 4. If there is an Address TLV with Type = `DESCRIPTION`, set `DI_description` := TLV-Value.

5. Remove all Network Interface Data Tuples with
NID_radio_interface_id = interface_id.
6. For each Address TLV with Type = NETWORK_DATA, add a Network
Interface Data Tuple with:
 - + NID_radio_interface_id := interface_id.
 - + NID_data_type := TLV Extension Type.
 - + NID_data_value := TLV Value.
 - + NID_time := current_time + validity_time.

13. Local Destination Order

If the Local Destination Set is not empty, the Local Destination Order MUST be sent by DLEP-radios in regular intervals, at least once every LOCAL_DESTINATION_INTERVAL.

A Local Destination Order contains the IP-prefixes of a DLEP-radio interface's locally configured Destination. This allows the DLEP-router to check the configuration of the DLEP-radio interfaces.

While it is allowed to split the Local Destination Order for multiple destinations into more than one Local Destination Order, each Local Destination Tuple must be contained once in a Local Destination Order within LOCAL_DESTINATION_INTERVAL.

The Local Destination Order includes a list of local Destination MAC-addresses as address objects.

For each address, the Local Destination Order MUST contain one ADD_ADDRESS Address TLV of Extension-Type := MAC with a single address. This specifies the Radio Interface ID of the local Destination.

For each address, it MAY contain one ADD_ADDRESS Address TLV of Extension-Type := IPV4_PREFIX with one or more prefix. This specifies the IP-prefixes of the local Destination.

For each address, it MAY contain one ADD_ADDRESS Address TLV of Extension-Type := IPV6_PREFIX with one or more prefix. This specifies the IP-prefixes of the local Destination.

13.1. Local Destination Order Generation

Each DLEP-radio MUST generate Local Destination Orders according to the specification in this section.

Define destination_set a subset of the Local Destination Set, which MUST contain at least one Local Destination Tuple.

Each Local Destination Order MUST contain the following additional elements:

- o A Message TLV with Type := ORDER and Value := LOCAL_DESTINATION.
- o A Message TLV with Type := VALIDITY_TIME (as specified in [[RFC5497](#)]) with Value := LOCAL_DESTINATION_HOLD_TIME (encoded as specified in [[RFC5497](#)]).

- o For each Local Destination Tuple in `destination_set`, add `LD_mac_address` as an address object (as defined in [[RFC5444](#)]) and apply the following steps:
 - 1. Add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `MAC` and Value := `LD_radio_interface_id`.
 - 2. If `LD_ipv4_prefixes` is not empty, add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `IPV4_PREFIX` and Value := `LD_ipv4_prefixes`.
 - 3. If `LD_ipv6_prefixes` is not empty, add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `IPV6_PREFIX` and Value := `LD_ipv6_prefixes`.

[13.2.](#) Local Destination Order Processing

A Local Destination Order is processed in the DLEP-router as follows:

- 1. Define `validity_time` as the Value of the Message TLV with Type = `VALIDITY_TIME`.
- 2. For each address object `destination_mac` that has an Address TLV with Type = `ADD_ADDRESS` and Extension Type = `MAC`,
 - 1. Define `interface_id` as the Value of the Address TLV with Type = `ADD_ADDRESS` and Extension Type = `MAC`.
 - 2. If there is a Network Local Destination Tuple with `NLD_radio_interface_id` = `interface_id` and `NLD_mac_address` = `destination_mac`, remove it (there should be a maximum of one such tuples).
- 3. Add a Network Local Destination Tuple with:
 - + `NLD_radio_interface_id` := `interface_id`.
 - + `NLD_mac_address` := `destination_mac`.
 - + `NLD_ipv4_prefixes` := `EMPTY`.
 - + `NLD_ipv6_prefixes` := `EMPTY`.
 - + `NLD_time` := `current_time` + `validity_time`.
- 4. If there is an Address TLV with Type = `ADD_ADDRESS` and Extension Type = `IPV4_PREFIX`, set `NLD_ipv4_prefixes`:= TLV-Value.

5. If there is an Address TLV with Type = ADD_ADDRESS and Extension Type = IPV6_PREFIX, set NLD_ipv6_prefixes:= TLV-Value.

14. Neighbor Update Order

If either the Interface Neighbor Set or the Lost Neighbor Set are not empty, the Neighbor Update Order MUST be sent by DLEP-radios in regular intervals, at least once every NEIGHBOR_UPDATE_INTERVAL.

A Neighbor Update Order contains the prefixes, status, and attributes (e.g. radio metrics and statistics) of one or more Neighbors of a single DLEP-radio interface. This allows the DLEP-router to receive the whole data about each Neighbor of the DLEP-radio interface in an atomic way.

While it is allowed to split the Neighbor Update Order for multiple Neighbors into more than one Neighbor Update Order, each Neighbor Interface Tuple and Lost Neighbor Tuple must be contained once in an Neighbor Update Order within NEIGHBOR_UPDATE_INTERVAL.

The Neighbor Update Order includes the Radio Interface ID of the local radio interface as the originator address.

The Neighbor Update Order includes a list of Neighbor MAC-addresses as address objects.

For each address, the Neighbor Update Order MUST contain the RADIOIF_STATUS Address TLV, which specifies if the neighbors radio interface is UP or DOWN.

For each address, it MAY contain one ADD_ADDRESS Address TLV of Extension-Type := MAC with a single address. This specifies the neighbor's Radio Interface ID.

For each address, it MAY contain one ADD_ADDRESS Address TLV of Extension-Type := IPV4_PREFIX with one or more prefix. This specifies the IP-prefixes of the neighbor.

For each address, it MAY contain one ADD_ADDRESS Address TLV of Extension-Type := IPV6_PREFIX with one or more prefix. This specifies the IP-prefixes of the neighbor.

For each address, it MAY contain one NETWORK_DATA Address TLV for each Extension-Type, which will specify the known statistics and metrics of the neighbor.

14.1. Neighbor Update Order Generation

Each DLEP-radio MUST generate Neighbor Update Orders according to the specification in this section, each for a single DLEP-radio interface, which means for a single Local Interface Tuple.

Define `neighbor_set` a subset of the Interface Neighbor Set with all tuples `IN_radio_interface_id = LI_radio_interface_id`.

Define `lost_neighbor_set` a subset of the Lost Neighbor Set with all tuples `LN_radio_interface_id = LI_radio_interface_id`.

`neighbor_set` and `lost_neighbor_set` MUST NOT be both empty.

Each Neighbor Update Order MUST contain the following additional elements:

- o `LI_radio_interface_id` as the originator address.
- o A Message TLV with Type := ORDER and Value := NEIGHBOR_UPDATE.
- o A Message TLV with Type := VALIDITY_TIME (as specified in [[RFC5497](#)]) with Value := NEIGHBOR_UPDATE_HOLD_TIME (encoded as specified in [[RFC5497](#)]).
- o For each Interface Neighbor Tuple in `neighbor_set`, add `IN_mac_address` as an address object (as defined in [[RFC5444](#)]) and apply the following steps:
 1. Add an Address TLV with Type := RADIOIF_STATUS and Value := UP.
 2. If `IN_neighbor_interface_id` != NONE, add an Address TLV with Type := ADD_ADDRESS, Type Extension := MAC and Value := `IN_neighbor_interface_id`.
 3. If `IN_ipv4_prefixes` is not EMPTY, add an Address TLV with Type := ADD_ADDRESS, Type Extension := IPV4_PREFIX and Value := `IN_ipv4_prefixes`.
 4. If `IN_ipv6_prefixes` is not EMPTY, add an Address TLV with Type := ADD_ADDRESS, Type Extension := IPV6_PREFIX and Value := `IN_ipv6_prefixes`.
- o For each Lost Neighbor Tuple in `lost_neighbor_set`, add `LN_mac_address` as an address object and apply the following steps:
 1. Add an Address TLV with Type := RADIOIF_STATUS and Value := DOWN.
 2. If `LN_neighbor_interface_id` != NONE, add an Address TLV with Type := ADD_ADDRESS, Type Extension := MAC and Value := `LN_neighbor_interface_id`.

14.2. Neighbor Update Order Discarding

If a Neighbor Update Order has no Originator Address, it MUST be dropped without further processing.

14.3. Neighbor Update Order Processing

A Neighbor Update Order is processed as follows:

1. Define `interface_id` as the Originator Address of the Neighbor Update Order.
2. Define `validity_time` as the Value of the Message TLV with Type = `VALIDITY_TIME`.
3. For each address object `neighbor_mac` that has an Address TLV with Type = `RADIOIF_STATUS`:
 1. Define `neighbor_status` as the Value of the Address TLV with Type = `RADIOIF_STATUS`.
 2. If there is an Address TLV with Type = `ADD_ADDRESS` and Extension Type = `MAC`, define `neighbor_radio_id` as the Value of the Address TLV. If not, define `neighbor_radio_id` as `NONE`.
 3. If there is a Network Neighbor Tuple with `NN_radio_interface_id = interface_id` AND `NN_neighbor_interface_id = neighbor_radio_id` AND `NN_mac_address = neighbor_mac`, remove it (there should be a maximum of one).
4. Add a Network Neighbor Tuple with:
 - + `NN_radio_interface_id := interface_id`.
 - + `NN_neighbor_interface_id := neighbor_radio_id`.
 - + `NN_status := neighbor_status`.
 - + `NN_mac_address := neighbor_mac`.
 - + `NN_ipv4_prefixes := EMPTY`.
 - + `NN_ipv6_prefixes := EMPTY`.
 - + `NN_time := current_time + validity_time`.

5. If there is an Address TLV with Type = ADD_ADDRESS and Extension Type = IPV4_PREFIX, set NN_ipv4_prefixes := TLV-Value.
6. If there is an Address TLV with Type = ADD_ADDRESS and Extension Type = IPV6_PREFIX, set NN_ipv6_prefixes := TLV-Value.
7. Remove all Network Neighbor Data Tuples with
NND_radio_interface_id = interface_id AND
NND_neighbor_interface_id = neighbor_radio_id AND
NND_mac_address = neighbor_mac.
8. For each Address TLV with Type = NEIGHBOR_DATA, add a Network Neighbor Data Tuple with:
 - + NND_radio_interface_id := interface_id.
 - + NND_neighbor_interface_id = neighbor_radio_id.
 - + NND_mac_address := neighbor_mac.
 - + NND_data_type := TLV Extension Type.
 - + NND_data_value := TLV Value.
 - + NND_time := current_time + validity_time.

[15.](#) Setup Radio Order

If the Radio Interface Configuration Set is not empty, the Setup Radio Order MUST be sent by DLEP-routers in regular intervals, at least once every SETUP_RADIO_INTERVAL.

A Setup Radio Order contains the status of DLEP-radio interfaces to be configured.

While it is allowed to split the Setup Radio Order for multiple radio interfaces into more than one Setup Radio Order, each Radio Interface Configuration Tuple must be contained once in a Setup Radio Order within SETUP_RADIO_INTERVAL.

The Setup Radio Order includes a list of Radio Interface ID's as address objects.

For each address, the Setup Radio Order MUST contain one RADIOIF_STATUS Address TLV to specify the radio interface status.

[15.1.](#) Setup Radio Order Generation

Each DLEP-router MUST generate Setup Radio Orders according to the specification in this section.

Define `interface_set` a subset of the Radio Interface Configuration Set, which MUST contain at least one Radio Interface Configuration Tuple.

Each Setup Radio Order MUST contain the following additional elements:

- o A Message TLV with Type := ORDER and Value := SETUP_RADIO.
- o A Message TLV with Type := VALIDITY_TIME (as specified in [\[RFC5497\]](#)) with Value := SETUP_RADIO_HOLD_TIME (encoded as specified in [\[RFC5497\]](#)).
- o For each Radio Interface Configuration Tuple in `interface_set`, add `RIC_radio_interface_id` as an address object (as defined in [\[RFC5444\]](#)) and apply the following steps:
 1. Add an Address TLV with Type := RADIOIF_STATUS and Value := `RIC_status`.

15.2. Setup Radio Order Processing

A Setup Radio is processed in the DLEP-router as follows:

1. Define `validity_time` as the Value of the Message TLV with Type = `VALIDITY_TIME`.
2. For each address object `interface_id` that has an Address TLV with Type = `RADIOIF_STATUS`,
 1. Define `interface_status` as the Value of the Address TLV with Type = `RADIOIF_STATUS`.
 2. If there is an Interface Configuration Tuple with `IC_radio_interface_id = interface_id`, remove it (there should be a maximum of one such tuples).
3. Add an Interface Configuration Tuple with:
 - + `IC_radio_interface_id := interface_id`.
 - + `IC_radio_status := interface_status`.
 - + `IC_time := current_time + validity_time`.

16. Setup Destination Order

If the Destination Configuration Set is not empty, the Setup Destination Order MUST be sent by DLEP-routers in regular intervals, at least once every `SETUP_DESTINATION_INTERVAL`.

A Setup Destination Order contains the IP-prefixes and Radio Interface ID of the local Destinations to be configured on a DLEP-radio interface.

While it is allowed to split the Setup Destination Order for multiple destinations into more than one Setup Destination Order, each Destination Configuration Tuple MUST be contained once in a Setup Destination Order within `SETUP_DESTINATION_INTERVAL`.

The Setup Destination Order includes a list of to be configured local Destination MAC-addresses as address objects.

For each address, the Setup Destination Order MUST contain one `ADD_ADDRESS` Address TLV of Extension-Type := MAC with a single address. This specifies the Radio Interface ID of the local Destination.

For each address, it MAY contain one `ADD_ADDRESS` Address TLV of Extension-Type := `IPV4_PREFIX` with one or more prefix. This specifies the IP-prefixes of the local Destination.

For each address, it MAY contain one `ADD_ADDRESS` Address TLV of Extension-Type := `IPV6_PREFIX` with one or more prefix. This specifies the IP-prefixes of the local Destination.

16.1. Setup Destination Order Generation

Each DLEP-router MUST generate Setup Destination Orders according to the specification in this section.

Define `destination_set` a subset of the Destination Configuration Set, which MUST contain at least one Destination Configuration Tuple.

Each Setup Destination Order MUST contain the following additional elements:

- o A Message TLV with Type := `ORDER` and Value := `SETUP_DESTINATION`.
- o A Message TLV with Type := `VALIDITY_TIME` (as specified in [[RFC5497](#)]) with Value := `SETUP_DESTINATION_HOLD_TIME` (encoded as specified in [[RFC5497](#)]).

- o For each Destination Configuration Tuple in `destination_set`, add `DC_mac_address` as an address object (as defined in [[RFC5444](#)]) and apply the following steps:
 - 1. Add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `MAC` and Value := `DC_radio_interface_id`.
 - 2. If `DC_ipv4_prefixes` is not empty, add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `IPV4_PREFIX` and Value := `DC_ipv4_prefixes`.
 - 3. If `DC_ipv6_prefixes` is not empty, add an Address TLV with Type := `ADD_ADDRESS`, Extension Type := `IPV6_PREFIX` and Value := `DC_ipv6_prefixes`.

[16.2](#). Setup Destination Order Processing

A Setup Destination Order is processed in the DLEP-radio as follows:

- 1. Define `validity_time` as the Value of the Message TLV with Type = `VALIDITY_TIME`.
- 2. For each address object `destination_mac` that has an Address TLV with Type = `ADD_ADDRESS` and Extension Type = `MAC`,
 - 1. Define `interface_id` as the Value of the Address TLV with Type = `ADD_ADDRESS` and Extension Type = `MAC`.
 - 2. If there is a Local Destination Tuple with `LD_radio_interface_id` = `interface_id` and `LD_mac_address` = `destination_mac`, remove it (there should be a maximum of one tuple).
 - 3. Add a Local Destination Tuple with:
 - + `LD_radio_interface_id` := `interface_id`.
 - + `LD_mac_address` := `destination_mac`.
 - + `LD_ipv4_prefixes` := `EMPTY`.
 - + `LD_ipv6_prefixes` := `EMPTY`.
 - + `LD_time` := `current_time` + `validity_time`.
 - 4. If there is an Address TLV with Type = `ADD_ADDRESS` and Extension Type = `IPV4_PREFIX`, set `LD_ipv4_prefixes` := TLV-Value.

5. If there is an Address TLV with Type = ADD_ADDRESS and Extension Type = IPV6_PREFIX, set LD_ipv6_prefixes:= TLV-Value.

17. IANA Considerations

This specification defines one Message Type, which MUST be allocated from the "Message Types" repository of [\[RFC5444\]](#), 1 Message TLV Types, which MUST be allocated from the "Message TLV Types" repository of [\[RFC5444\]](#), and 5 Address Block TLV Types, which MUST be allocated from the "Address Block TLV Types" repository of [\[RFC5444\]](#).

17.1. Expert Review: Evaluation Guidelines

For the registries where an Expert Review is required, the designated expert SHOULD take the same general recommendations into consideration as are specified by [\[RFC5444\]](#).

17.2. Message Types

This specification defines one Message Type, to be allocated from the 0-223 range of the "Message Types" namespace defined in [\[RFC5444\]](#), as specified in Table 8.

+-----+-----+	
Type	Description
+-----+-----+	
TBD1	DLEP: Dynamic link exchange protocol
+-----+-----+	

Table 8: Message type assignment.

17.3. Message-Type-Specific TLV Type Registries

IANA is requested to create a registry for Message-Type-specific Message TLVs for DLEP messages, in accordance with [Section 6.2.1 of \[RFC5444\]](#), and with initial assignments and allocation policies as specified in Table 9.

+-----+-----+-----+		
Type	Description	Allocation Policy
+-----+-----+-----+		
128-223	Unassigned	Expert Review
+-----+-----+-----+		

Table 9: DLEP Message-Type-specific Message TLV Type

IANA is requested to create a registry for Message-Type-specific Address Block TLVs for DLEP messages, in accordance with [Section 6.2.1 of \[RFC5444\]](#), and with initial assignments and allocation policies as specified in Table 10.

+-----+-----+-----+		
Type	Description	Allocation Policy
+-----+-----+-----+		
128-223	Unassigned	Expert Review
+-----+-----+-----+		

Table 10: DLEP Message-Type-specific Address Block TLV Types

[17.4.](#) Message TLV Types

TODO: allocate ORDER (message specific?)

[17.5.](#) Address Block TLV Types

TODO: allocate RADIOIF_STATUS (message specific?)

TODO: allocate DESCRIPTION (message specific?)

TODO: allocate ADD_ADDRESS (message specific?)

TODO: allocate NETWORK_DATA (message specific?)

TODO: allocate NEIGHBOR_DATA (message specific?)

18. Further work

There are two aspects of [[dlep02](#)] that have not been part of this document:

- o Configuration of the radio channel.
- o Managing of traffic credit tokens.

The author of this documents has the opinion that there is also demand for a discussion on the following questions:

- o What is the exact difference between Latency and Expected-Forwarding-Delay and how shall both of them be defined?
- o What are the full consequences of multicast MAC-addresses as destinations?
- o Is it possible to allow multiple devices (DLEP-routers and end-devices) connected with an Ethernet Switch to a DLEP-radio?

19. Security Considerations

Currently, this protocol does not specify any special security measures.

[20.](#) References

[20.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March 1997.
- [RFC5148] Clausen, T., Dearlove, C., and B. Adamson, "Jitter Considerations in Mobile Ad Hoc Networks (MANETs)", [RFC 5148](#), February 2008.
- [RFC5444] Clausen, T., Dearlove, C., Dean, J., and C. Adjih, "Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format", [RFC 5444](#), February 2009.
- [RFC5497] Clausen, T. and C. Dearlove, "Representing Multi-Value Time in Mobile Ad Hoc Networks (MANETs)", [RFC 5497](#), March 2009.
- [RFC6130] Clausen, T., Dearlove, C., and J. Dean, "Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)", [RFC 6130](#), April 2011.
- [RFC6622] Clausen, T. and U. Herberg, "Integrity Check Value and Timestamp TLV Definitions for Mobile Ad Hoc Networks (MANETs)", [RFC 6622](#), July 2012.

[20.2.](#) Informative References

- [dlep02] Ratliff, S., Berry, B., Harrison, G., Jury, S., and D. Satterwhite, "Dynamic Link Exchange Protocol (DLEP)", [draft-ietf-manet-dlep-02](#) (work in progress), February 2012.
- [olsrv2] Clausen, T., Dearlove, C., Jacquet, P., and U. Herberg, "The Optimized Link State Routing Protocol version 2", [draft-ietf-manet-olsrv2-15](#) (work in progress), September 2009.

[Appendix A](#). [RFC5444](#)-DLEP Message Examples

DLEP messages are instances of [\[RFC5444\]](#) messages, and this protocol supports any combination of message header options and address encodings, enabled by [\[RFC5444\]](#) that convey the required information. As a consequence, there is no single way to represent how all DLEP messages look.

[A.1](#). Interface Update Order Example

The Interface Update Order is used by DLEP-radios to announce their radio interfaces. The status of the interface, an optional description and a number of optional network based attributes can be attached to each radio interface.

In the following example that is depicted in Figure 2, we see a simple Interface Update Order of a DLEP-radio with a single interface which is up and there are no further attributes set.

The DLEP message's four bit Message Flags (MF) field has value 0, indicating that no optional message header fields are present. Its four bit Message Address Length (MAL) field has value 5, indicating addresses in the message have a length of six octets, here being Radio Interface IDs. The overall message length is 28 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and ORDER. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value Length of 1 octet. The Value included in the first TLV is a time code (as defined in [\[RFC5497\]](#)) representing the parameters INTERFACE_UPDATE_HOLD_TIME. The Value included in the second TLV is a Order ID, respectively INTERFACE_UPDATE (see [Section 11.2](#)).

The message has a single Address Block containing one address. The Address Block Flags octet (ABF) value 0 indicates an no address Head, no address Tail, and no address prefixes. The Address Block is followed by the full Radio Interface ID.

The following Address Block TLV Block (content length 4 octets) includes one Address Block TLV. The TLV is a RADIOIF_STATUS Address Block TLV with Flags octet (ATLVF) value 16, which indicates that a single address, with index 0 (implicit) belongs to an interface with the status UP (the one octet Value UP indicates this).


```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      DLEP      | MF=0 | MAL=5 |      Message Length = 28      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Message TLV Block Length = 8 | VALIDITY_TIME | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | Value (Time) |      ORDER      | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | INTERF.UPD. | Num Addrs = 1 |      ABF = 0      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 1 Byte 1 | RIID 1 Byte 2 | RIID 1 Byte 3 | RIID 1 Byte 4 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 1 Byte 5 | RIID 1 Byte 6 | Address TLV Block Length = 4 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RADIOIF_STATUS| ATLVF = 16 | Value Len = 1 | VALUE (UP) |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Example of a simple Interface Update Order.

Figure 2

The second example (see Figure 3) is a more complex Interface Update Order of a DLEP-radio with two interface (first up, second down), each with a 4 character description (both different) and the supported data-rate (both have the same data rate). The two Radio Interface IDs have the common header HEAD_1/HEAD_2/HEAD_3.

The DLEP message's four bit Message Flags (MF) field has value 0, indicating that no optional message header fields are present. Its four bit Message Address Length (MAL) field has value 5, indicating addresses in the message have a length of six octets, here being Radio Interface IDs. The overall message length is 56 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and ORDER. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value Length of 1 octet. The Value included in the first TLV is a time code (as defined in [\[RFC5497\]](#)) representing the parameters INTERFACE_UPDATE_HOLD_TIME. The Value included in the second TLV is a Order ID, respectively INTERFACE_UPDATE (see [Section 11.2](#)).

The message has a single Address Block containing two address. The Address Block Flags octet (ABF) value 128, indicates an address Head, but no address Tail and Prefixes. The Head Length of 3 octets indicates address Mid sections of 3 octets each.

The following Address Block TLV Block (content length 28 octets) includes three Address Block TLVs. The first TLV is a RADIOIF_STATUS Address Block TLV with Flags octet (ATLVF) value 20, which indicates that it contains a value for each of the addresses (UP/DOWN). The second TLV is a DESCRIPTION Address Block TLV with Flags octet (ATLVF) value 20 too, and a four byte value with the descriptions for each address. The last TLV is a NETWORK_DATA TLV with Flags octet (ATLVF) value 144, which indicates an TLV extension type and a single value for both addresses.

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      DLEP      | MF=0 | MAL=5 |      Message Length = 56      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Message TLV Block Length = 8 | VALIDITY_TIME | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | Value (Time) |      ORDER      | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | INTERF.UPD. | Num Addrs = 2 | ABF = 128 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Head L. = 3 | HEAD 1 | HEAD 2 | HEAD 3 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 1 Byte 4 | RIID 1 Byte 5 | RIID 1 Byte 6 | RIID 2 Byte 4 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 2 Byte 5 | RIID 2 Byte 6 | Address TLV Block Length = 28 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RADIOIF_STATUS| ATLVF = 20 | Value Len = 2 | VALUE (UP) |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| VALUE (DOWN) | DESCRIPTION | ATLVF = 20 | Value Len = 8 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     Description Interface 1      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     Description Interface 2      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| NETWORK_DATA | ATLVF = 144 | SUPP. RATES | Value Len = 8 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     Common Supported Data Rate Part 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|                                     Common Supported Data Rate Part 2
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Example of a simple Interface Update Order.

Figure 3

[A.2.](#) Neighbor Update Order Example

The Neighbor Update Order is used by DLEP-radios to announce neighbors of their radio interfaces. The status of the neighbor, the neighbors IP addresses and a number of metrics and statistics can be attached to each neighbor.

In the following example that is depicted in Figure 4, we see a Neighbor Update Order of a DLEP-radio with a four neighbors, three of them up and one down. All of them have their radio interface_id, the three active neighbors have a relative link quality and the second neighbor also has an IP address set. The four Neighbor MAC-addresses have the common header HEAD_1/HEAD_2/HEAD_3.

The DLEP message's four bit Message Flags (MF) field has value 128, indicating that an originator address message header field is present. Its four bit Message Address Length (MAL) field has value 5, indicating addresses in the message have a length of six octets, here being MAC-addresses. The overall message length is 83 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and ORDER. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value Length of 1 octet. The Value included in the first TLV is a time code (as defined in [\[RFC5497\]](#)) representing the parameters NEIGHBOR_UPDATE_HOLD_TIME. The Value included in the second TLV is a Order ID, respectively NEIGHBOR_UPDATE (see [Section 11.2](#)).

The message has a single Address Block containing three addresses. The Address Block Flags octet (ABF) value 128 indicates an address Head, but no address Tail and prefixes. The Head Length of 3 octets indicates address Mid sections of 3 octets each.

The following Address Block TLV Block (content length 42 octets) includes three Address Block TLVs. The first TLV is a RADIOIF_STATUS Address Block TLV with Flags octet (ATLVF) value 20, which indicates that each address has a value of its own (two times UP, once DOWN). The second TLV is an ADD_ADDRESS Address Block TLV with Flags octet (ATLVF) value 148, which indicates a TLV Type Extension (MAC) and a value of its own. Its followed by three neighbor Radio Interface IDs, each six octets. The third TLV is a NEIGHBOR_DATA Address Block TLV with Flag octet (ATLVF) value 180, which indicates a TLV Type Extension (RELATIVE_LQ) and a value for each of the first two addresses. The last TLV is an ADD_ADDRESS Address Block TLV with Flags octet (ATLVF) 208, which indicates a TLV Type Extension (IPV4_PREFIX) for a the second address.


```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      DLEP      | MF=128| MAL=5 |      Message Length = 83      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Local Radio Interface ID Part 1      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Local R. Interface ID Part 2 | Message TLV Block Length = 8 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| VALIDITY_TIME | MTLVF = 16 | Value Len = 1 | Value (Time) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      ORDER      | MTLVF = 16 | Value Len = 1 | NEIGHB.UPD. |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Num Addrs = 3 | ABF = 128 | Head L. = 3 |      HEAD 1      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      HEAD 2      |      HEAD 3      | MAC 1 Byte 4 | MAC 1 Byte 5 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MAC 1 Byte 6 | MAC 2 Byte 4 | MAC 2 Byte 5 | MAC 2 Byte 6 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MAC 3 Byte 4 | MAC 3 Byte 5 | MAC 3 Byte 6 | Address TLV - |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Block L. = 42 | RADIOIF_STATUS| MTLVF = 20 | Value Len = 3 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| VALUE (UP) | VALUE (UP) | VALUE (DOWN) | ADD_ADDRESS |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ATLVF = 148 | MAC | Value L. = 18 | N1 RIID B.1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N1 RIID B.2 | N1 RIID B.3 | N1 RIID B.4 | N1 RIID B.5 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N1 RIID B.6 | N2 RIID B.1 | N2 RIID B.2 | N2 RIID B.3 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N2 RIID B.4 | N2 RIID B.5 | N2 RIID B.6 | N3 RIID B.1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N3 RIID B.2 | N3 RIID B.3 | N3 RIID B.4 | N3 RIID B.5 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| N3 RIID B.6 | NEIGHBOR_DATA | ATLVF = 180 | RELATIVE_LQ |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| index-start=0 | index-end=1 | Value Len = 2 | REL_LQ 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| REL_LQ 2 | ADD_ADDRESS | ATLVF = 208 | IPV4_PREFIX |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| index-start=1 | Value Len = 5 | IPaddr Byte 1 | IPaddr Byte 2 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPaddr Byte 3 | IPaddr Byte 4 | IPaddr Prefix |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Example of an Neighbor Update Order.

Figure 4

[A.3.](#) Setup Radio Order Example

The Setup Router Order is used by DLEP-routers to set the status of the DLEP-radios interfaces.

In the following example that is depicted in Figure 5, we see a Setup Router Order for a DLEP-radio with two interfaces. Both of them are configured to status UP.

The DLEP message's four bit Message Flags (MF) field has value 0, indicating that no optional message header fields are present. Its four bit Message Address Length (MAL) field has value 5, indicating addresses in the message have a length of six octets, here being Radio Interface IDs. The overall message length is 32 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and ORDER. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value Length of 1 octet. The Value included in the first TLV is a time code (as defined in [\[RFC5497\]](#)) representing the parameters SETUP_RADIO_HOLD_TIME. The Value included in the second TLV is a Order ID, respectively SETUP_RADIO (see [Section 11.2](#)).

The message has a single Address Block containing two addresses. The Address Block Flags octet (ABF) value 128 indicates an address Head, but no address Tail and prefixes. The Head Length of 3 octets indicates address Mid sections of 3 octets each.

The following Address Block TLV Block (content length 4 octets) includes one Address Block TLV. The TLV is a RADIOIF_STATUS Address Block TLV with Flags octet (ATLVF) value 16, which indicates the same value (UP) for both addresses.


```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|      DLEP      | MF=0 | MAL=5 |      Message Length = 32      |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Message TLV Block Length = 8 | VALIDITY_TIME | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | Value (Time) |      ORDER      | MTLVF = 16 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Value Len = 1 | SETUP_RADIO | Num Addrs = 2 |      ABF = 128   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| Head L. = 3 |      HEAD 1   |      HEAD 2   |      HEAD 3   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 1 Byte 4 | RIID 1 Byte 5 | RIID 1 Byte 6 | RIID 2 Byte 4 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RIID 2 Byte 5 | RIID 2 Byte 6 | Address TLV Block Length = 4 |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| RADIOIF_STATUS|      ATLVF = 16 | Value Len = 1 |      VALUE (UP)  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Example of an Setup Router Order.

Figure 5

[A.4.](#) Setup Destination Order

The Setup Destination Order is used by DLEP-routers to set its local IP addresses for a destination on the DLEP-radios interfaces.

In the following example that is depicted in Figure 6, we see a Setup Destination Order for a DLEP-radio with one interfaces. The router configures a single MAC-address with one IPv4 address and two IPv6 addresses for this interface.

The DLEP message's four bit Message Flags (MF) field has value 0, indicating that no optional message header fields are present. Its four bit Message Address Length (MAL) field has value 5, indicating addresses in the message have a length of six octets, here being MAC addresses. The overall message length is 81 octets.

The message contains a Message TLV Block with content length 8 octets containing two Message TLVs, of types VALIDITY_TIME and ORDER. Each uses a Message TLV with Flags octet (MTLVF) value 16, indicating that each has a Value, and each has a Value Length of 1 octet. The Value included in the first TLV is a time code (as defined in [\[RFC5497\]](#)) representing the parameters SETUP_DESTINATION_HOLD_TIME. The Value included in the second TLV is a Order ID, respectively SETUP_DESTINATION (see [Section 11.2](#)).

The message has a single Address Block containing one addresses. The Address Block Flags octet (ABF) value 0 indicates an no address Head, no address Tail, and no address prefixes. The Address Block is followed by the full MAC-address of the Destination.

The following Address Block TLV Block (content length 57 octets) includes three Address Block TLV. All TLVs are ADD_ADDRESS Address Block TLVs with Flags octet (ATLVF) value 144, which indicates a TLV Type Extension and a value. The first TLV (IPV4_PREFIX Type Extension) contains a single IPv4 address with prefix, the second one (IPV6_PREFIX Type Extension) contains two IPv6 addresses with prefix.


```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      DLEP      | MF=0 | MAL=5 |      Message Length = 81      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Message TLV Block Length = 8 | VALIDITY_TIME | MTLVF = 16 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Value Len = 1 | Value (Time) |      ORDER      | MTLVF = 16 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Value Len = 1 | SETUP_DEST. | Num Addrs = 1 |      ABF = 0      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MAC 1 Byte 1 | MAC 1 Byte 2 | MAC 1 Byte 3 | MAC 1 Byte 4 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MAC 1 Byte 5 | MAC 1 Byte 6 | Address TLV Block Length = 57 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ADD_ADDRESS | ATLVF = 144 |      MAC      | Value Len = 6 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| RIID Byte 1 | RIID Byte 2 | RIID Byte 3 | RIID Byte 4 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| RIID Byte 5 | RIID Byte 6 | ADD_ADDRESS | ATLVF = 144 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPV4_PREFIX | Value Len = 5 | IPv4 Byte 1 | IPv4 Byte 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv4 Byte 1 | IPv4 Byte 1 | Prefix (32) | ADD_ADDRESS |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ATLVF = 144 | IPV6_PREFIX | Value L. = 34 | IPv6 1 Byte 1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 1 Byte 2 | IPv6 1 Byte 3 | IPv6 1 Byte 4 | IPv6 1 Byte 5 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 1 Byte 6 | IPv6 1 Byte 7 | IPv6 1 Byte 8 | IPv6 1 Byte 9 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 1 B. 10 | IPv6 1 B. 11 | IPv6 1 B. 12 | IPv6 1 B. 13 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 1 B. 14 | IPv6 1 B. 15 | IPv6 1 B. 16 | IPV6 1 Prefix |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 2 Byte 1 | IPv6 2 Byte 2 | IPv6 2 Byte 3 | IPv6 2 Byte 4 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 2 Byte 5 | IPv6 2 Byte 6 | IPv6 2 Byte 7 | IPv6 2 Byte 8 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 2 Byte 9 | IPv6 2 B. 10 | IPv6 2 B. 11 | IPv6 2 B. 12 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 2 B. 13 | IPv6 2 B. 14 | IPv6 2 B. 15 | IPv6 2 B. 16 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv6 2 Prefix |
+---+---+---+---+---+

```

Example of an Setup Destination Order.

Figure 6

[A.5.](#) Local Destination Order Example

The Local Destination Order looks exactly like the Setup Destination Order, only that it is sent from the DLEP-radio to the DLEP-router.

Author's Address

Henning Rogge
Fraunhofer FKIE
Neuenahrer Strasse 20
Wachtberg 53343
Germany

Phone: +49 228 9435-961

Email: henning.rogge@fkie.fraunhofer.de

URI: <http://fkie.fraunhofer.de/>