StageLBarraud

De DrakkarWiki.

Sommaire

Naming

- LID = location-ID
- UID = permanent ID of a node, unique in the network - could be the mac address.

Ideas

- The "LID" (location-ID) of a node represents the route from the sink in a compact way.
- When node A requests a route to node B, both nodes must recall the LID of each other. If one of the nodes changes its LID, it must then notify the other node, which allows to avoid flooding the network when such a change occurs.
- Use this at level 2.
- "LID" leases are temporary (time interval expressed as a number of beacons). The parent may re-address LIDs after this period ends. To avoid this the child must re-request an address before the period ends; in this request it specifies how many childs and connections it has, itself, in order for the parent to judge if re-addressing it has a low impact; if possible, the parent will reaffect the same address (specified in LIDsrc).
- When a node receives its LID, if this LID has the max length it doesn't emit beacons (cannot accept children)

Advantage compared to Zigbee

- No node number limitation (in fact, 2^255)
- No need to pre-configure node for specific network organisation
- Manage network organisation change
- No risk of misrouting packet

Messages definition

Common patterns

```
                               +---------+---------+
LIDxxx (variable bit-length) = | LIDsize | LIDcode |
                               +---------+---------+

LIDsize = bit-size of LIDcode as a 8 bits unsigned integer
LIDcode = bit-code of the Location IDentifier (= bit-coded route from the sink)
```

```
          +------+---------+---------+----------+----------+---------+-----------+
Header =  | Type | SrcType | DstType | [LIDsrc] | [LIDdst] | NextHop | [padding] |
          +------+---------+---------+----------+----------+---------+-----------+
Type = Message type - 4 bits
    0-> Beacon
    1-> Child Attachement New Request
    2-> Child Attachement Renewal Request
    3-> Child Attachement Response
    4-> Child LID Assignment
    5-> Communication Request
    6-> Communication Response
    7-> Communication Discarding
    8-> Communication Failure
    9-> Unreachability
   10-> Location Change
   11-> Data
   12-> reserved
   13-> reserved
   14-> reserved
   15-> reserved
SrcType = source type - 1bit:
    0-> unassigned
    1-> LIDsrc specified
DstType = destination type - 3bits:
    0-> unassigned
    1-> LIDdst specified
    2-> forward flooding
    3-> subtree flooding
    4-> reserved
    5-> reserved
    6-> reserved
    7-> reserved
LIDsrc = LID of the message source (only present if SrcType = 1)
LIDdst = LID of the message destination (only present if DstType = 1)
NextHop = Designation of close target
padding = zeroes until next byte
```

```
          +------+----------+-------+
NextHop = | Mode | [Prefix] | [Lid] |
          +------+----------+-------+

Mode = NextHop mode addressing - 2 bits
    00-> Source Prefix Mode
    01-> Destination Prefix Mode
    10-> Specified LID Mode
    11-> Broadcast (beacons)
Prefix = Only if source prefix mode or destination prefix mode,
         bit-size of the suffix who, added to the target node come to source LID (respectively destinati
Lid = Only if Specified LID mode.
      It's the LID of the target node.
```

```
      +-----+-----+
STW = | SNC | SCC |
      +-----+-----+

SNC = Sub-tree node count (8 bits unsigned integer)
SCC = Sub-tree communication count (8 bits unsigned integer)
```

Beacon

```
+--------+------+------+-----+
| Header | Rank | ALSC | STW |
+--------+------+------+-----+
```

```
Header = Type and Path Specification (SrcType = 1; DstType = 0; type = TYPE_BEACON)
```

```
Rank = Integer rank from the PAN coordinator + Flag disconnected (true/false)? (8 bits unsigned integer
ALSC = Available LID slots count (8 bits unsigned integer)
STW = Sub-tree weight (see below)
```

Child Attachment New Request

```
+--------+--------+-----+
| Header | UIDreq | STW |
+--------+--------+-----+
```

```
Header = Type and Path Specification (SrcType = 0, DstType = 1, LIDdst = parent node; Type = TYPE_CHILD_
```

```
UIDreq = UID of child
STW = Sub-tree weight
(remove : Flags | * Force: true / false -always force- )
```

Child Attachment Renewal Request

```
+--------+-----+
| Header | STW |
+--------+-----+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1, LIDdst = parent node; Type = TYPE_CHILD_
LIDsrc = current LID (in case of renewal)
LIDdst = parent node
```

```
STW = Sub-tree weight (see beacon message)
```

Child Attachment Response

```
+--------+--------+--------+---------+
| Header | UIDreq | LIDnew | ExpTime |
+--------+--------+--------+---------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 0,  Type = TYPE_CHILD_ATTACHMENT_RESPONSE)
LIDsrc = sender
LIDdst = <null>
```

```
LIDnew = LID assigned
ExpTime = LIDnew expiry time expressed as a number of beacon intervals (8 bits unsigned integer)
```

Child LID Assignment

```
+--------+--------+---------+
| Header | LIDnew | ExpTime |
+--------+--------+---------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1, LIDdst = child old LID, Type = TYPE_CHILI
LIDsrc = sender
LIDdst = child old LID
```

```
LIDnew = LID assigned
ExpTime = LIDnew expiry time expressed as a number of beacon intervals (8 bits unsigned integer)
```

Communication Request

```
+--------+--------+-----------+
| Header | UIDreq | [Payload] |
+--------+--------+-----------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1 or 2, Type = TYPE_COMMUNICATION_ESTABLISH
LIDsrc = sender
LIDdst = specific LID or forward-flooding
```

```
UIDreq = UID for which we want the LID
Payload = optional payload
```

Communication Response

```
+--------+--------+
| Header | UIDreq |
+--------+--------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1, Type = TYPE_LOCATOR_RESPONSE)
LIDsrc = sender (= requested LID)
LIDdst = requester
```

```
UIDreq = UID for which the LID was requested
```

Communication Discarding

```
+--------+-----------+
| Header | [Payload] |
+--------+-----------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1, Type = TYPE_COMMUNICATION_DISCARDING)
LIDsrc = sender
LIDdst = communication peer
```

```
Payload = optional payload
```

Communication Failure

```
+--------+
| Header |
+--------+
```

```
PathSpec = Path specification (SrcType = 1, DstType = 1, Type = TYPE_COMMUNICATION_FAILURE)
LIDsrc = sender
LIDdst = communication peer
```

Unreachability

```
+--------+---------+
| Header | LIDfail |
+--------+---------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = 1, Type = TYPE_UNREACHABILITY)
LIDsrc = sender
LIDdst = communication peer
```

```
LIDfail = LID requested but unavailable
```

Location Change

```
+--------+--------+
| Header | LIDold |
+--------+--------+
Header = Type and Path specification (SrcType = 1, LIDsrc = new LID, DstType = 1 or 3, Type = TYPE_LOCAT
LIDsrc = sender (= new LID)
LIDdst = specific LID or subtree-flooding
```

```
LIDold = previous LID of the sender
```

Data

```
+--------+---------+
| Header | Payload |
+--------+---------+
```

```
Header = Type and Path specification (SrcType = 1, DstType = user choice, Type = TYPE_DATA)
LIDsrc = sender
LIDdst = receiver LID (or any type of flooding)
```

Communication phases

Child Attachment

- The child sends a "Child attachment new request" message (force = false)
- The parent may choose to discard the message

Communication ending

We cannot express the expiry time of a communication channel as a number of beacon intervals because the number of hops between the 2 nodes may vary and therefore affect this expiry time. That is why we introduce a new message used to discard the communication channel. This is the responsability of the communication initiator to send this message.

We must also handle un-notified expired LIDs. This can occur when a node A left the network and could not send the communication discarding message to node B. After the LID of A is expired, either:

- The LID of A was reaffected to node C. Node C may therefore receive unexpected messages from B (i.e. node B is not registered in its communication table); in this case, node C should send a communication failure message to node B.
- The LID of A was not reaffected. The parent node of A, let's call it P, may

therefore receive messages from B with an unexpected LIDdst; in this case, node P should send an unreachability message to node B.

Stored informations

All of the LIDs who should be stored are in a allocated static memory block. They can be found with LIDP (LID pointer).

```
        +-----------+------+-----+
LIDP => | mem_index | size | NoP |
        +-----------+------+-----+

mem_index = index in the memory block
size = size of the LID pointed
NoP = Number of Pointer who point to this LIDP
```

All of the LIDP are stored in a static table.

Personal information

- A Node keeps some informations about itself :
  - personal LIDP
  - parent LID-length (with this and personal LID, it's possible to know the parent LID)
  - personal STW

Tables

It also store some tables

- Children Table

```
+----------------------------+
|      Children Table        |
+--------------+-------------+
| Child1 ILIDP | Child1 STW |
| -   -   -   -|-   -   -  -|
| Child2 ILIDP | Child2 STW |
| -   -   -   -|-   -   -  -|
|  ...         | ...        |
+--------------+-------------+
```

- Initiated Com Table

```
+---------------------------+
|    Initiated Com Table    |
+--------------+------------+
| Peer1 ILIDP  | Peer1 UID  |
| -   -   -   -|-   -   -  -|
| Peer2 ILIDP  | Peer2 UID  |
| -   -   -   -|-   -   -  -|
|  ...         | ...        |
+--------------+------------+
```

- Requested Com Table

```
+---------------------------+
|   Requested Com Table     |
+-------------+-------------+
|        Peer1 ILIDP        |
| -  -  -   -   -   -   -  -|
|        Peer2 ILIDP        |
| -  -  -   -   -   -   -  -|
|           .....           |
+-------------+-------------+
```

```
ILIDP = Index of LIDP
```

Récupérée de « https://intranet-drakkar.imag.fr/wiki/index.php/StageLBarraud »

- Dernière modification de cette page le 15 juillet 2013 à 12:53.