LCRTLS 定位基站与定位引擎之间数据通讯

重要说明：本文档是贵阳联创智能网络科技有限公司内部资料，请注意保密。

本文所述的“数据包”、“消息”，是同义词，有时会因为上下文或表达方式的原因而使用不同的词，但它们是同一个意思。

定位基站的基本任务是把接收到的标签的 UWB 定位数据包组装后，转发给定位引擎，再由定位引擎计算标签的坐标。

1. 消息类型定义

```
#define NET_MSG_TYPE_TAG_RANGE                  (2)  // Anchor 发给 RTLE. Anchor 在收到 Tag 的 TDOA 测距包后，向 RTLE 报告收到的测距包. 因为每个基站可能会与多个时钟源同步时钟，所以每一个 Tag 测距包会生成多条本消息

#define NET_MSG_TYPE_NOTIFY_ANCHOR_POE_INFO          (10) // Anchor 定期发给 RTLE，通知 Anchor 信息

#define NET_MSG_TYPE_ALIVE                      (13) // 心跳包，用于通知对方自己还活着，没其它意义
```

2. 消息结构

2.1. 测距消息

```
typedef struct {
    uint16_t pack_sign;         // 0, 2: 55 AA
    uint16_t pack_size;         // 2, 2: 包大小(字节)
    uint8_t pack_sum;           // 4, 1: 包字节和，用于校验

    uint8_t msg_type;           // 5, 1: 包类型: NET_MSG_TYPE_TAG_RANGE
    uint8_t seq;                // 6, 1:
    uint8_t anchor_addr[8];     // 7, 8:
    int64_t pure_standard_ts64; // 15, 8:
    uint64_t message_serial_no; // 23, 8:
    uint32_t ldo1_voltage;
    uint32_t ldo2_voltage;

    uint8_t  cs_addr[8];            // 39, 8:
    uint8_t  tag_addr[8];           // 47, 8:
    uint16_t power_voltage;         // 55, 2:
    uint16_t battery_voltage;       // 57, 2:
    uint16_t lighteness;            // 59, 2:
    uint8_t  switch_status;         // 61, 1:
```

```
    int16_t first_path_power_level;
    int16_t receive_signal_power_level;

    int16_t imu_aacx;                // 66, 2:
    int16_t imu_aacy;                // 68, 2:
    int16_t imu_aacz;                // 70, 2:
    int32_t imu_roll;                // 72, 4:
    int32_t imu_pitch;               // 76, 4:
    int32_t imu_yaw;                 // 80, 4:
    int16_t imu_temp;                // 84, 2:
    uint8_t imu_silent;              // 86, 1:
    uint8_t imu_signal;              // 87, 1:

    uint8_t uwb_module_no;
} TagRangePacket;
```

说明:

- 有几个重要的字段:msg_type 是消息类型, anchor_addr 是基站的 Id, message_serial_no 是消息的序号，cs_addr 是时钟源的 Id，tag_addr 是标签的 Id。
- 对于标签来说，它定期发送 UWB 定位消息，我们称为测距消息。tag_addr 是标签 ID
- 对于某一个 tag，每条测距消息的 message_serial_no 会是递增的
- anchor_addr 是收到测距消息的基站的 ID
- cs_addr 是此基站时钟同步的时钟源的 ID
- 如果基站与多个时钟源同步时钟，基站收到 UWB 测距消息时，会组装出多条相似的网络测距消息，对应不同的时钟源

2.2. 基站信息

```
typedef struct {
    uint16_t pack_sign;                         // 0, 2: 55 AA
    uint16_t pack_size;                         //  2, 2: 包大小(字节)
    uint8_t pack_sum;                           //  4, 1: 包字节和, 用于校验
    uint8_t msg_type;                           //           5,     1:     包    类    型    :
NET_MSG_TYPE_NOTIFY_ANCHOR_POE_INFO
    uint8_t seq;                                //  6, 1:

    uint8_t hw_model;                           //  7, 1: 硬件型号
    uint8_t hw_ver_major;                       //  8, 1: 硬件版本
    uint8_t hw_ver_minor;                       //  9, 1: 硬件版本
    uint8_t fw_ver_major;                       // 10, 1: Firmware version major
    uint8_t fw_ver_minor;                       // 11, 1: Firmware version minor

    uint8_t serial_no[14];                      // 11+1, 14: Anchor 的 SN, 20170317090909
    uint8_t anchor_id[8];                       // 25+1, 8: EUI64
    uint8_t anchor_type;                        // 33, 1: 1=普通基站(ANCHOR_TYPE_ANCHOR), 2=时钟
源(ANCHOR_TYPE_CLOCK_SOURCE)
    uint16_t clock_sync_interval;               // 34, 2: 时钟同步间隔(ms)
    uint16_t pan_id;                            // 36, 2: UWB 的 PAN ID
    uint8_t uwb_channel;                        // 38, 1: 超宽带频道: 1,2,3,4,5,7
    uint8_t uwb_data_rate;                      //  39,  1:  超宽带数据传输速度:  0=110K,  1=850K,
2=6.8M
    uint8_t receive_clock_message_from_same_panid_only;      // 40, 1: 只接收来自相同 PAN ID 的
时钟信号
    uint8_t receive_range_message_from_same_panid_only;      // 41, 1:   只接收来自相同 PAN ID
的测距信号
```

```
    uint8_t  anchor_name[66];                  // 42, 66: 基站名字(只是助记符而已), 最长 64 个字
符或 32 个汉字
    uint8_t  rtle_auto_discover;          //108, 1: RTLE 自动发现. 0=使用指定的 RTLE, 1=使用自动
发现的 RTLE
    uint8_t  rtle_ip[4];                       //109, 4: R4TLE IP
    uint16_t  rtle_port;                       //113, 2: RTLE 的 TCP 端口

    uint8_t  rj45_mac[6];                      //115, 6: MAC 地址
    uint8_t  rj45_auto_get_ip;                 //121, 1: 自动获取 ip 地址
    uint8_t  rj45_ip[4];                       //122, 4: 本地 IP
    uint8_t  rj45_subnet[4];              //126, 4: 子网掩码
    uint8_t  rj45_gateway[4];                  //130, 4: 网关

    uint8_t  dns[4];                           //134, 4: DNS,
    uint16_t  webserver_port;                  //138, 2: 侦听的 WebServer 端口 (80), 用于配置
Anchor, 80
    uint16_t  tcpserver_port;                  //140, 2: 侦听的 TCP 端口, 用于发送数据, 1200

    uint8_t  uwb_module_num;              //142, 1: UWB 模块数量

    uint16_t  packets_num_per_sec;             //142, 2: 每秒收到多少个包
    uint16_t  cs_packets_num_per_sec;     //144, 2: 每秒收到多少个包
    uint16_t  range_packets_num_per_sec;  //146+1, 2: 每秒收到多少个包
} NOTIFY_ANCHOR_POE_INFO;
```

说明：基站信息会被基站定期发送给定位引擎，定位引擎收到后，检查是否有变化，更新自己内部保存的基站信息。其中有些字段是冗余信息，或者根据不同类型的基站，有些字段不一定有意义。

如果做负载均衡，此消息应该分发给所有定位引擎。

2.3. 心跳包

```
typedef struct {   // 6 Bytes
    uint16_t  pack_sign;                  // 0, 消息头: 0x55, 0xAA
    uint16_t  pack_size;                  // 2, 消息的长度, 包括 pack_sign
    uint8_t  pack_sum;                    // 4, 包字节和, 用于校验
    uint8_t  msg_type;                    // 5, 消息类型
} NET_MESSAGE_ALIVE;
```

说明：心跳包是最小数据包。在收到基站发来的测距消息和基站信息后，应该立即回复这个心跳包给基站，通知基站这个 TCP 连接是正常工作中的。否则基站隔 10 秒收不到任何消息，它会认为这个 TCP 连接有故障，它会主动关闭 TCP 连接，再重新连接到引擎。

3. 其他说明

如果遇到其他类型的数据包，可以直接忽略之。

检验和字段 pack_sum 是从下一字节开始到包结束的全部字节和的低 8 位。代码大致如下：

```
    pmsg->pack_sum = 0;
```

```
uint8_t *p = (uint8_t *)pmsg;
p += 5;
for (int16_t i=5; i<pmsg->pack_size; i++) {
     pmsg->pack_sum += (*p);
     p++;
}
```