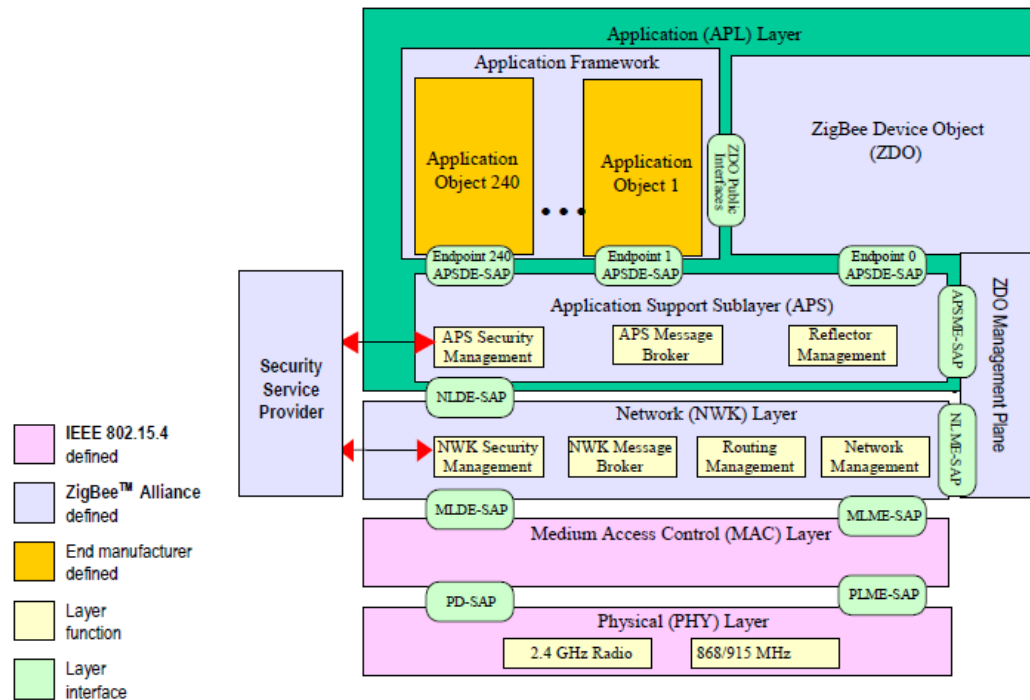


*Please use this document together with the lab 5 note

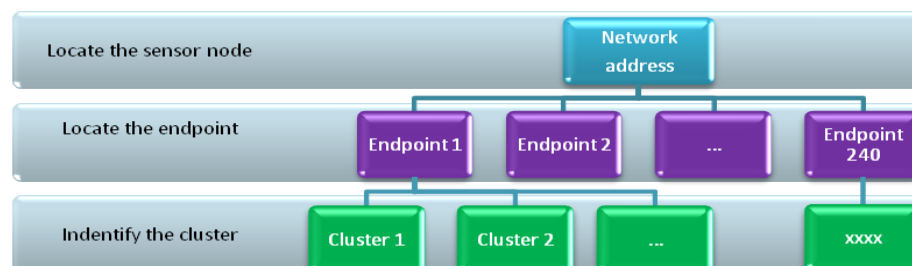
-----Section 1 -----

ZigBee endpoints and clusters

1.



Recall this ZigBee stack structure introduced in the first lab. Your ZigBee network applications will be sitting in these Application Object blocks, which are defined as endpoints in ZigBee specification. The valid range of endpoint ID is 2~240. Another important thing which is not shown in this structure is the clusters. The valid range of cluster ID is 0x0000~0xFFFF.



The above block shows you that, an ordinary communication in a ZigBee

network needs this three-level addressing. Thus if you want to add a ZigBee application, the first thing you need to figure out is which endpoint and cluster you would like to use.

-----Section 1 stops here-----
////////////////////////////////////
-----Section 2 -----

Read and write payload

1. Below is the code to initiate a packet transmission:

```
/* Load the payload with the data to send */
for(i = 0; i < length; i++)
{
    TxBuffer[TxData++] = sending_data;    // write the data into transmit buffer
}

params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT;    // use network address
params.APSDE_DATA_request.DstAddress.ShortAddr.v[1] = NWK_address; //MSB of Dest address
params.APSDE_DATA_request.DstAddress.ShortAddr.v[0] = NWK_address; //LSB of Dest address

params.APSDE_DATA_request.SrcEndpoint    = SourceEndpoint;    // set the endpoints
params.APSDE_DATA_request.DstEndpoint    = DestinationEndpoint;
params.APSDE_DATA_request.ProfileId.Val  = MY_PROFILE_ID;      // ignore this

//params.APSDE_DATA_request.asduLength; TxData
params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS;      // ignore this
params.APSDE_DATA_request.TxOptions.bits.acknowledged = 1;      // use ACK or not
params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS; // select the
routing mode

// ignore the security
#ifdef I_SUPPORT_SECURITY
    params.APSDE_DATA_request.TxOptions.Val = 1;
#else
    params.APSDE_DATA_request.TxOptions.Val = 0;
#endif

params.APSDE_DATA_request.ClusterId.Val = Cluster_ID;    // set the cluster ID
ZigBeeBlockTx();    // block other transmission until finish
currentPrimitive = APSDE_DATA_request;    // give the command of transmission
```

If you want to transmit any data through the ZigBee network, that is the code you need. You can place the code in anywhere, for example, in your own PuTTY Menu.

2. The following code shows you what to do when a node receives a packet:

```
/*firstly you may want to read the payload first*/
received_data 1 = APLGet();    //APLGet() function will return the
```

```

received_data 2 =APLGet();    // data in payload one by one.
received_data 3 =APLGet();

    // then you can do some processing or other work

/* you may also want to send some data back*/

TxBuffer[TxData++] = sending_data1;    //writ the data you want to send back into
TxBuffer[TxData++] = sending_data2;    // the buffer, if there is any
TxBuffer[TxData++] = sending_data3;
params.APSDE_DATA_request.DstAddrMode = APS_ADDRESS_16_BIT; //use network address
params.APSDE_DATA_request.DstAddress.ShortAddr=params.APSDE_DATA_indication.SrcAddress.ShortAddr;
// use the source address of the received packet as Dest address

params.APSDE_DATA_request.SrcEndpoint = endpoints; // set the endpoints
params.APSDE_DATA_request.DstEndpoint = endpoints;
params.APSDE_DATA_request.ProfileId.Val = MY_PROFILE_ID; // ignore this one

//params.APSDE_DATA_request.asduLength;
params.APSDE_DATA_request.RadiusCounter = DEFAULT_RADIUS; // ignore it

params.APSDE_DATA_request.TxOptions.bits.acknowledged = 1; // use ACK or not
params.APSDE_DATA_request.DiscoverRoute = ROUTE_DISCOVERY_SUPPRESS; //set the
routing mode

// ignore the security
#ifdef I_SUPPORT_SECURITY
    params.APSDE_DATA_request.TxOptions.Val = 1;
#else
    params.APSDE_DATA_request.TxOptions.Val = 0;
#endif

params.APSDE_DATA_request.ClusterId.Val= Cluster_ID; //set the cluster ID

ZigBeeBlockTx(); // block other transmission until finish
currentPrimitive = APSDE_DATA_request; // give the command of transmission

```

Now, it is your task to find out where to add the code for receiving a packet in "Coordinator.c" file. You can also set up the network and try to send and receive some data by yourself. You are also able to check the endpoints and clusters, you have set, through ZENA.

3. After learned how to transmit and receive data, next you will be asked to achieve a communication task.
 - a. Programing two nodes: one coordinator and one router (or RFD)
 - b. Achieving the lab 4 tasks in this lab. Please remind that the coordinator just plays the role of transmitting data and displaying the feedback. The other node does the processing task like doing summing and converting. You can use the ZENA to monitor the communicating between the two nodes. You need to attach the ZENA printscreen in your lab report 1.

-----finished ☺-----

By now, all the basic knowledge is covered in five labs, the project will begin soon.