

Wireless Sensor Networks

Chiara Buratti

c.buratti@unibo.it

+39 051 20 93147

Office Hours:

Tuesday 3 – 5 pm @ Main Building, third floor

Credits: 6



Syllabus: Laboratory Activities

1. PAN Formation
2. Data Transfer (point-to-point)
3. MAC Protocol
4. **MAC Protocol: Small Network**
5. NET Protocol: Small Network



Outline

- 1. Modify the Association Procedure at Coordinator**
- 2. Send a query from Coordinator to End Devices (unicast packets)**
- 3. Reply from End Devices with a burst of Data frames**
- 4. Count the number of packets received from each End Device**



Outline

- 1. Modify the Association Procedure at Coordinator**
2. Send a query from Coordinator to End Devices (unicast packets)
3. Reply from End Devices with a burst of Data frames
4. Count the number of packets received from each End Device

Association

```

static uint8_t App_SendAssociateResponse(nwkMessage_t *pMsgIn)
{
    .....
    /* Create the Associate response message data. */
    pAssocRes = &pMsg->msgData.associateRes;
    if(pMsgIn->msgData.associateInd.capabilityInfo & gCapInfoAllocAddr_c)
    {
        if (countED == 0)
        { pAssocRes->assocShortAddress[0] = 0x01;
          pAssocRes->assocShortAddress[1] = 0x00;
          FLib_MemCpy(maDeviceShortAddress1, pAssocRes->assocShortAddress, 2);
          FLib_MemCpy(maDeviceLongAddress1, pAssocRes->deviceAddress, 8);
          countED++;
        }
        else
        { pAssocRes->assocShortAddress[0] = 0x02;
          pAssocRes->assocShortAddress[1] = 0x00;
          FLib_MemCpy(maDeviceShortAddress2, pAssocRes->assocShortAddress, 2);
          FLib_MemCpy(maDeviceLongAddress2, pAssocRes->deviceAddress, 8);
        }
    }
}

```



Outline

1. Modify the Association Procedure at Coordinator
2. **Send a query from Coordinator to End Devices (unicast packets)**
3. Reply from End Devices with a burst of Data frames
4. Count the number of packets received from each End Device

Send the Query

```

static void App_TransmitQuery(void)
{
    if( (mcPendingPackets < mDefaultValueOfMaxPendingDataPackets_c) && (mpPacket == NULL) )
    {
        mpPacket = MSG_AllocType(nwkToMcpsMessage_t);
    }

    if(mpPacket != NULL)
    {
        mpPacket->msgData.dataReq.pMsdu = "query";
        mpPacket->msgType = gMcpsDataReq_c;
        FLib_MemCpy(mpPacket->msgData.dataReq.dstAddr, maDeviceShortAddress1, 2);
        FLib_MemCpy(mpPacket->msgData.dataReq.srcAddr, (void *)maShortAddress, 2);
        FLib_MemCpy(mpPacket->msgData.dataReq.dstPanId, (void *)maPanId, 2);
        FLib_MemCpy(mpPacket->msgData.dataReq.srcPanId, (void *)maPanId, 2);
        mpPacket->msgData.dataReq.dstAddrMode = gAddrModeShort_c;
        mpPacket->msgData.dataReq.srcAddrMode = gAddrModeShort_c;
        mpPacket->msgData.dataReq.msduLength = 5;
        mpPacket->msgData.dataReq.txOptions = gTxOptsAck_c | gTxOptsIndirect_c;
        .....
    }
}

```



Outline

1. Modify the Association Procedure at Coordinator
2. Send a query from Coordinator to End Devices (unicast packets)
- 3. Reply from End Devices with a burst of Data frames**
4. Count the number of packets received from each End Device



Reply to the Query

Send the burst of data when Data Indication is received.

Transmission with ACK and Retx (3):

```
mpPacket->msgData.dataReq.txOptions = gTxOptsAck_c;
```

Transmission of periodic data:

```
TMR_StartSingleShotTimer(mTimer_c, period, App_TransmitData);
```

Where:

period → to be set, corresponds to the period between two packets in ms

```
static void App_TransmitData(uint8_t tmr)
```

No need to wait for the ACK from Coordinator



Outline

1. Modify the Association Procedure at Coordinator
2. Send a query from Coordinator to End Devices (unicast packets)
3. Reply from End Devices with a burst of Data frames
4. **Count the number of packets received from each End Device**



Check the Address at the Coordinator

```
uint8_t rxPacketAddress[2];
```

```
rxPacketAddress[0] = pMsgIn->msgData.dataInd.srcAddr[0];
```

```
if ( rxPacketAddress[0] == maDeviceShortAddress1[0] )
```

```
{count1++;}
```

```
else
```

```
{count2++;}
```



Wireless Sensor Networks

Chiara Buratti

www.chiaraburatti.org

c.buratti@unibo.it