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. NET Protocol (small network)

Chiara Buratti
DEIS, University of Bologna


MAC Protocol

Chiara Buratti
DEIS, University of Bologna

Outline

1. IEEE 802.15.4 MAC Protocol
2. Aim of the Experiment
3. How to SET MAC parameters
4. Delay evaluation

Chiara Buratti
DEIS, University of Bologna

Outline

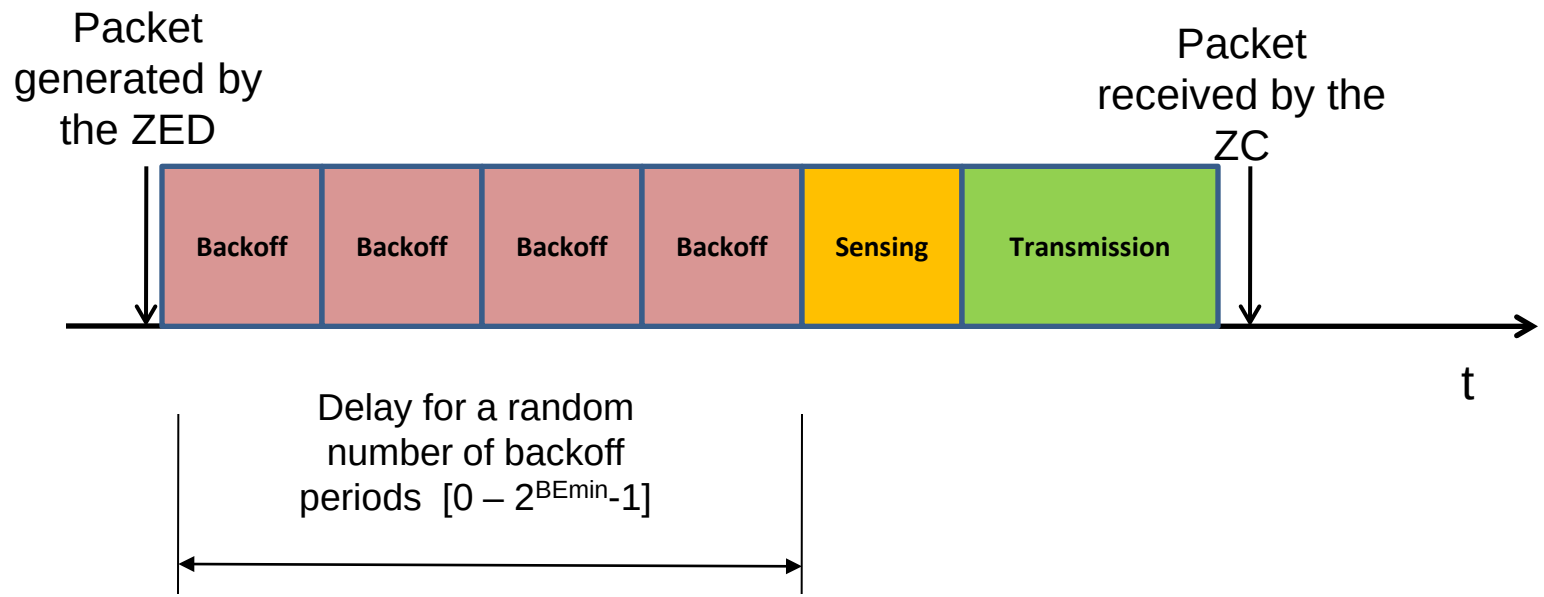
1. **IEEE 802.15.4 MAC Protocol**
2. Aim of the Experiment
3. How to SET MAC parameters
4. Delay evaluation

Chiara Buratti
DEIS, University of Bologna

Non Beacon-Enabled Mode: CSMA/CA

ZED transmits a data to the ZC.

Point-to-point → channel is found free at the first time.



Chiara Buratti
DEIS, University of Bologna

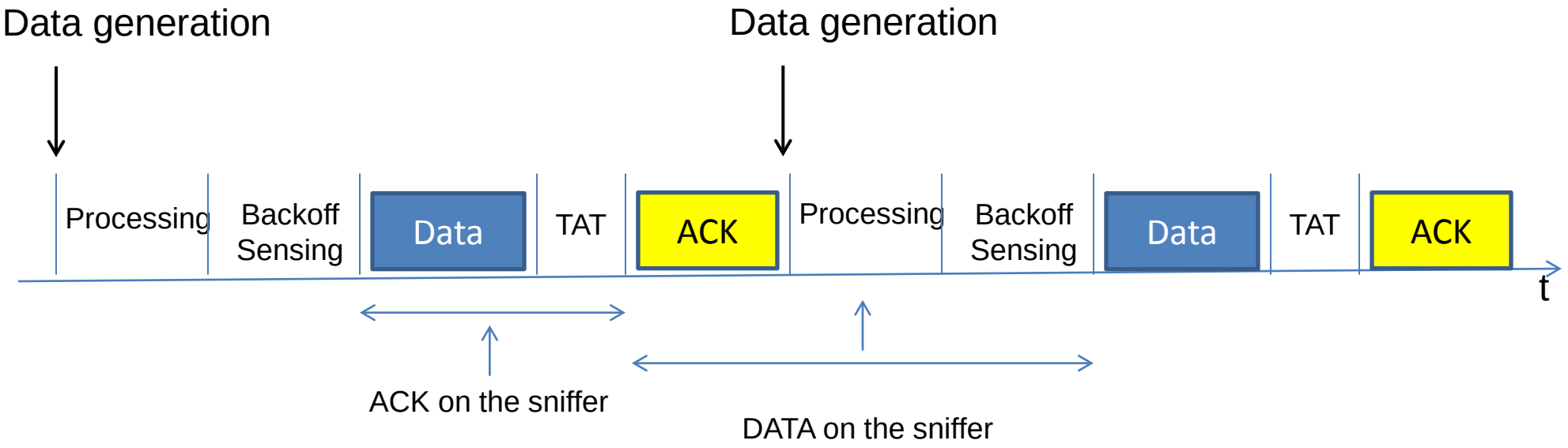
Outline

1. IEEE 802.15.4 MAC Protocol
2. **Aim of the Experiment**
3. How to SET MAC parameters
4. Delay evaluation

Chiara Buratti
DEIS, University of Bologna

Experiment

- Send many consecutive packets from the ZED to the ZC, by setting different values of BE_{min}
- Observe how the average delay varies by changing BE_{min}



Chiara Buratti
DEIS, University of Bologna

Outline

1. IEEE 802.15.4 MAC Protocol
2. Aim of the Experiment
3. **How to SET MAC parameters**
4. Delay evaluation

Chiara Buratti
DEIS, University of Bologna

Set MAC PIB (PAN Information Based) Attributes

The MAC PIB holds all the MAC attributes/variables that are accessible for the application. According to the 802.15.4 Standard, the primitive MLME-SET.request could be used to write a given PIB attribute.

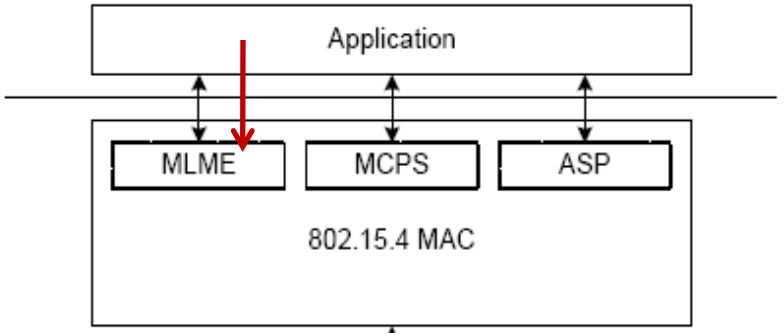
```
MLME-SET.request (
                    PIBAttribute,
                    PIBAttributeValue
                   )

MLME-SET.confirm (
                    status,
                    PIBAttribute
                   )
```

Chiara Buratti
DEIS, University of Bologna

Message Types: NWK-to-MLME direction

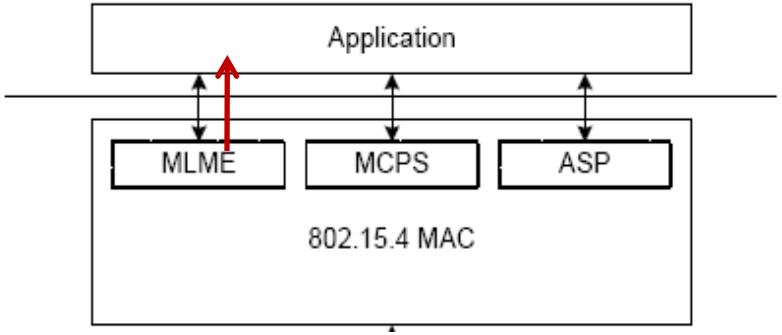
Message Identifier (primNwkToMlme_t)	802.15.4 NWK to MLME Primitives
gMlmeAssociateReq_c	MLME-ASSOCIATE.Request
gMlmeAssociateRes_c	MLME-ASSOCIATE.Response
gMlmeDisassociateReq_c	MLME-DISASSOCIATE.Request
gMlmeGetReq_c	MLME-GET.Request
gMlmeGtsReq_c	MLME-GTS.Request
gMlmeOrphanRes_c	MLME-ORPHAN.Response
gMlmeResetReq_c	MLME-RESET.Request
gMlmeRxEnableReq_c	MLME-RX-ENABLE.Request
gMlmeScanReq_c	MLME-SCAN.Request
gMlmeSetReq_c	MLME-SET.Request
gMlmeStartReq_c	MLME-START.Request
gMlmeSyncReq_c	MLME-SYNC.Request
gMlmePollReq_c	MLME-POLL.Request



Chiara Buratti
DEIS, University of Bologna

Message Types: MLME-to-NWK direction

Message Identifier (primMlmeToNwk_t)	802.15.4 MLME to NWK Primitives
gNwkAssociateInd_c	MLME-ASSOCIATE.Indication
gNwkAssociateCnf_c	MLME-ASSOCIATE.Confirm
gNwkDisassociateInd_c	MLME-DISASSOCIATE.Indication
gNwkDisassociateCnf_c	MLME-DISASSOCIATE.Confirm
gNwkBeaconNotifyInd_c	MLME-BEACON-NOTIFY.Indication
gNwkGetCnf_c	N/A
gNwkGtsInd_c	MLME-GTS.Indication
gNwkGtsCnf_c	MLME-GTS.Confirm
gNwkOrphanInd_c	MLME-ORPHAN.Indication
gNwkResetCnf_c	MLME-RESET.Confirm
gNwkRxEnableCnf_c	MLME-RX-ENABLE.Confirm
gNwkScanCnf_c	MLME-SCAN.Confirm
gNwkCommStatusInd_c	MLME-COMM-STATUS.Indication
gNwkSetCnf_c	N/A
gNwkStartCnf_c	MLME-START.Confirm
gNwkSyncLossInd_c	MLME-SYNC-LOSS.Indication
gNwkPollCnf_c	MLME-POLL.Confirm
gNwkPollNotifyIndication_c	Freescale proprietary Poll Notify Indication



Freescale 802.15.4 MAC SW
PAN Information Base (PIB) Attributes - SET

```
uint8_t App_SetMacPib_Example(uint8_t attribute, uint8_t *pValue)
{
  mlmeMessage_t mlmeSet;

  /* Create and execute the Set request */
  mlmeSet.msgType = gMlmeSetReq_c;
  mlmeSet.msgData.setReq.pibAttribute = attribute;
  mlmeSet.msgData.setReq.pibAttributeValue = pValue;

  return MSG_Send(NWK_MLME, &mlmeSet);
}
```

Synchronous operation, MSG_Send() call is processed immediately .

The return value of MSG_Send() is gSuccess_c if the set request was processed correctly or gInvalidParameter_c if parameter verification failed.

Chiara Buratti
DEIS, University of Bologna

Set MAC PIB (PAN Information Based) Attributes

The MAC PIB holds all the MAC attributes/variables that are accessible for the application. According to the 802.15.4 Standard, the primitive MLME-GET.request could be used to read a given PIB attribute.

```
MLME-GET.request (
                    PIBAttribute,
                  )

MLME-GET.confirm (
                  status,
                  PIBAttribute,
                  PIBAttributeValue
                  )
```

Chiara Buratti
DEIS, University of Bologna

Freescale 802.15.4 MAC SW
PAN Information Base (PIB) Attributes - GET

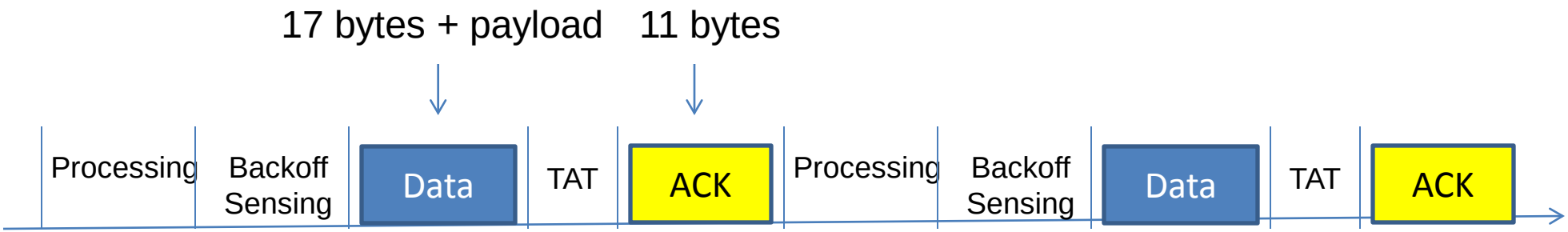
```
static uint8_t App_GetParam(uint8_t attribute, uint8_t *pValue)
{    mlmeMessage_t mlmeGet;
     mlmeGet.msgType=gMlmeGetReq_c;
     mlmeGet.msgData.getReq.pibAttribute=attribute;
     mlmeGet.msgData.getReq.pibAttributeValue=pValue;
     return MSG_Send (NWK_MLME,&mlmeGet);
}
```

Chiara Buratti
DEIS, University of Bologna

Freescale 802.15.4 MAC SW PAN - Information Base (PIB) Attributes

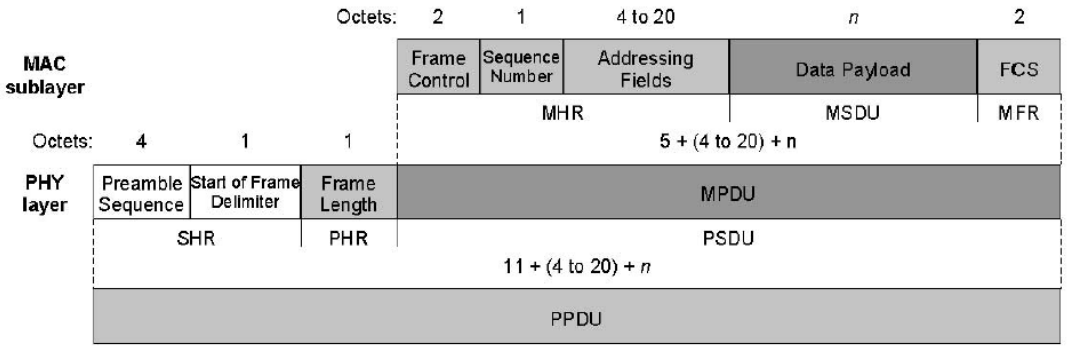
802.15.4 Specific Attributes (see [1] for descriptions)	
0x40	gMPibAckWaitDuration_c
0x41	gMPibAssociationPermit_c
0x42	gMPibAutoRequest_c
0x43	gMPibBattLifeExt_c
0x44	gMPibBattLifeExtPeriods_c
0x45	gMPibBeaconPayload_c
0x46	gMPibBeaconPayloadLength_c
0x47	gMPibBeaconOrder_c
0x48	gMPibBeaconTxTime_c
0x49	gMPibBsn_c
0x4A	gMPibCoordExtendedAddress_c
0x4B	gMPibCoordShortAddress_c
0x4C	gMPibDsn_c
0x4D	gMPibGtsPermit_c
0x4E	gMPibMaxCsmaBackoffs_c
0x4F	gMPibMinBe_c
0x50	gMPibPanId_c
0x51	gMPibPromiscuousMode_c
0x52	gMPibRxOnWhenIdle_c
0x53	gMPibShortAddress_c
0x54	gMPibSuperFrameOrder_c
0x55	gMPibTransactionPersistenceTime_c

Compute the Processing Time - Sniffer



TAT = 12 T_s = 192 μs
Packet Size =
11 + addressing fields + payload

Addressing fields = 6 Bytes



Chiara Buratti
DEIS, University of Bologna

Outline

1. IEEE 802.15.4 MAC Protocol
2. Aim of the Experiment
3. How to SET MAC parameters
4. **Delay evaluation**

Timer MC1322

The time pointer points to a 4-byte array where a 30-bit value represents the MACA symbol clock time.
The function does not return any value because the call is always successful.

```
zbClock32_t currentTime=0;

Asp_GetTimeReq(&currentTime);

UartUtil_PrintHex((uint8_t *)&currentTime, sizeof(currentTime),0);
UartUtil_Print("\n\r",gAllowToBlock_d);
```

The timer received on UART is in Simbol Time.

Standard Input / Output

- In the world of microcontrollers, there is no environment that provides standard IO. Unlike ANSI C, there is no *printf()* function that prints formatted strings to the default output and no *scanf()* function that reads formatted strings from standard input.

- In Freescale 802.15.4 MAC Stack:
  - Sending strings to the serial port:

    void UartUtil_Print(uint8_t* pString, uartUtilBlock_t allowToBlock)

    allowToBlock: gNoBlock_d, gAllowToBlock_d

  - Sending hex data to serial port:

    void UartUtil_PrintHex(uint8_t* hex, uint8_t len, uint8_t flags)

    flags: gPrtHexBigEndian_c, gPrtHexNewLine_c , gPrtHexCommas_c, PrtHexSpaces_c

Printing data on UART

```
UartUtil_Tx(pMsgIn->msgData.dataInd.pMsdu, pMsgIn->msgData.dataInd.msduLength);
UartUtil_Print(" ",gAllowToBlock_d);

UartUtil_PrintHex(&pMsgIn->msgData.dataInd.mpduLinkQuality,1, gPrtHexBigEndian_c |
gPrtHexNewLine_c);

UartUtil_Print("\n\r",gAllowToBlock_d);
```

Wireless Sensor Networks

Chiara Buratti

www.chiaraburatti.org

c.buratti@unibo.it