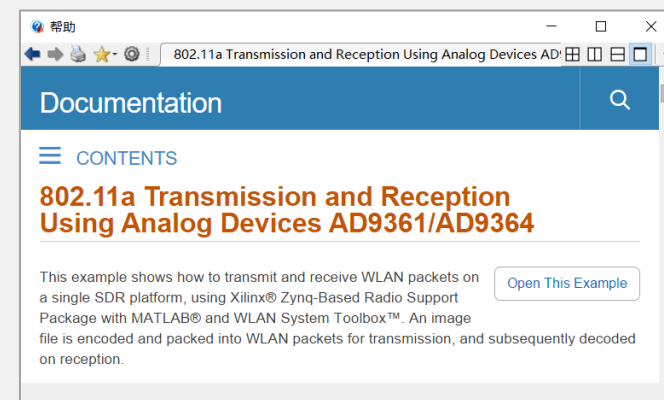
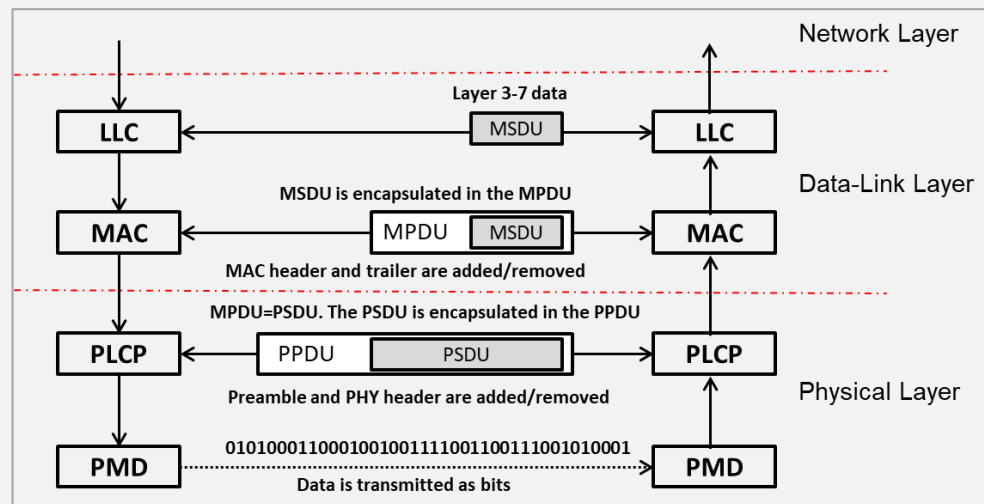
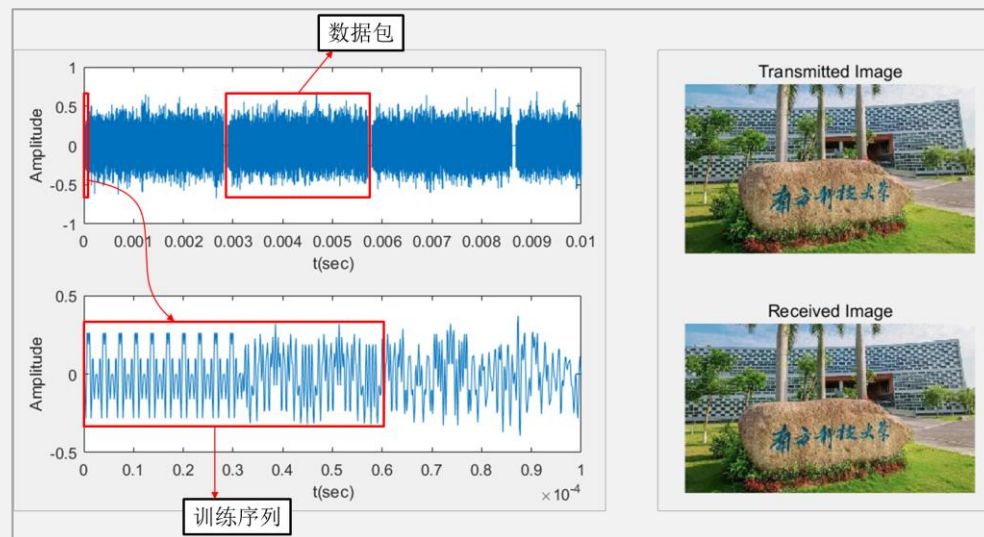
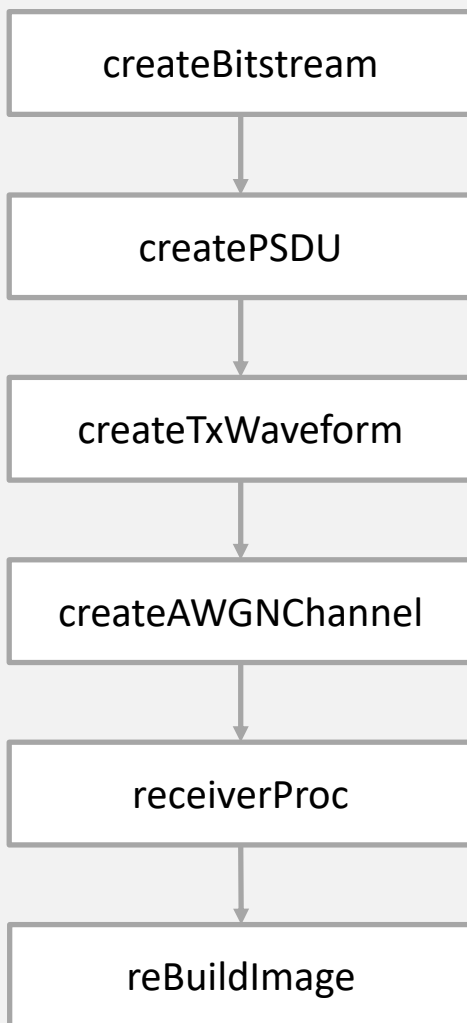
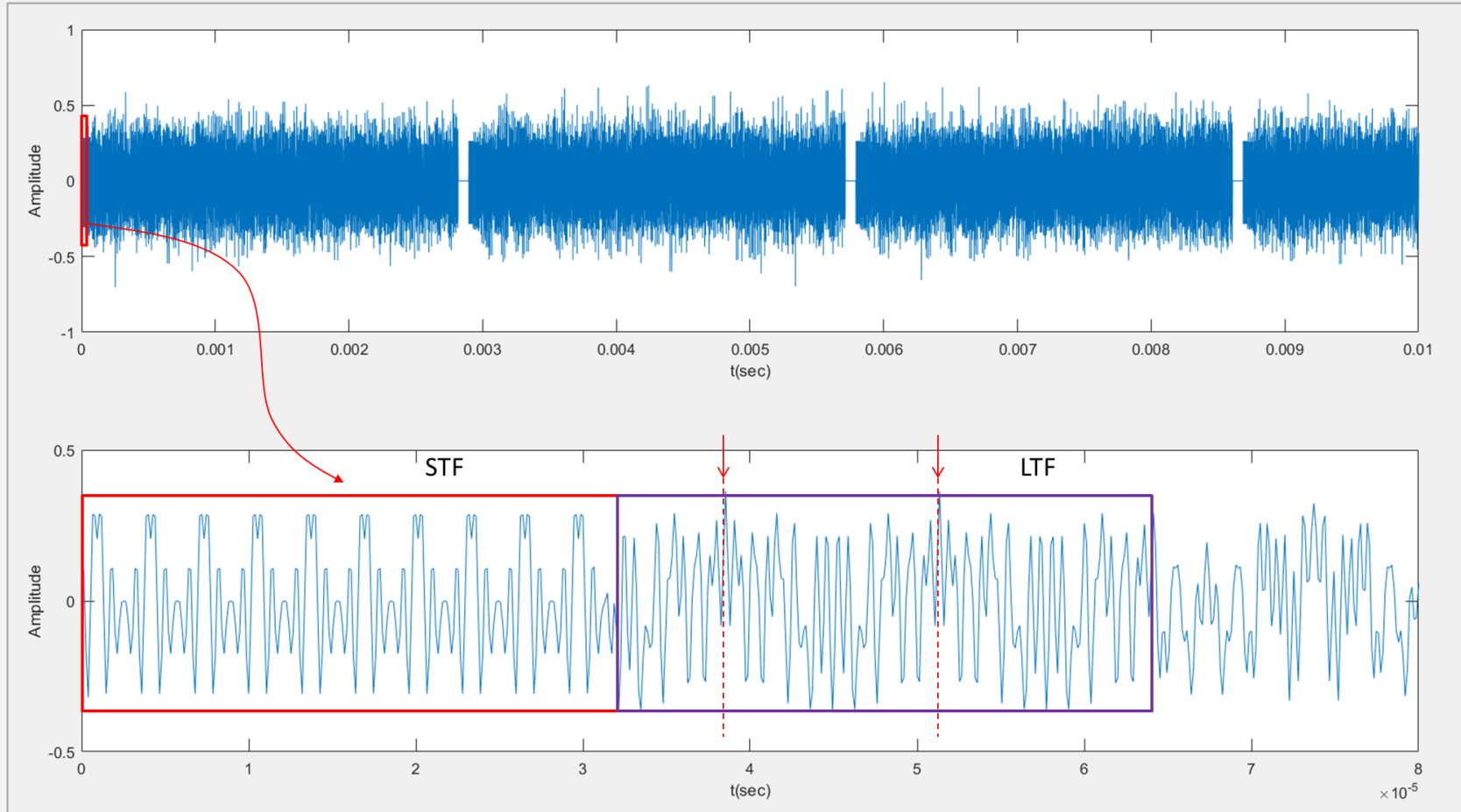


Review—1



Review—2



Transmitted Image



Received Image



前沿通信系统设计

1 WiFi通信系统 (6周)

实验目标: 利用USRP实现802.11a/n图像传输

软件: MATLAB, 硬件: USRP

授课内容: MATLAB通信编程、USRP文本传输、MIMO系统、802.11a/n仿真、802.11a/n图像传输

2 5G/4G-LTE系统 (5周)

实验目标: 利用USRP实现LTE图像传输

软件: MATLAB, 硬件: USRP

授课内容: 小区搜索过程、MIB/SIB解码过程、LTE图像传输、LDPC编解码过程、srsLTE系统

3 无线网络传输系统 (3周)

实验目标: 利用Telos实现无线多跳网络传输数据

软件: TinyOS、NesC

授课内容: TinOS编程、MICA2平台介绍、无线多跳网络数据收集、无线信道建模、无线定位、路由和数据收集

4 雷达感知系统 (2周)

实验目标: 利用KerberosSDR实现测向

软件: MATLAB, 硬件: KerberosSDR、树莓派

授课内容: MUSIC算法、空间谱估计、KerberosSDR原理, 无线开源项目, 课程Presentation

Communication Systems Design

Lab 5: 802.11a Image Transmission and Reception

(Part 3)

Dr. **Wu Guang**

wug@sustech.edu.cn

Electrical & Electronic Engineering

Southern University of Science and Technology

How to build a WiFi packet ?

- How to pack the information bits?
- How to design the training sequences ?
- How to decode the data field?
- How to compete for the wireless channels ?

Transmitted Image



Received Image



How to decode the data field ?


```

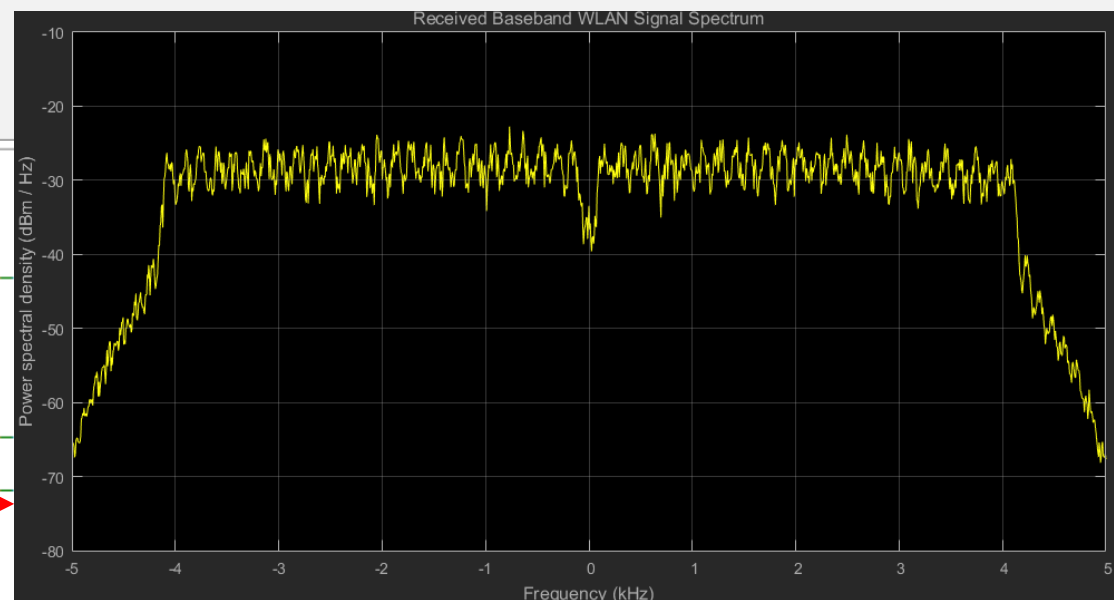
77 %% 5. 接收机信号处理过程
78 %5.1. 捕获WLAN数据包
79 — load('rxWaveform.mat') %----->这一行用于离线程序测试
80 — burstCaptures=[rxWaveform;rxWaveform];
81
82 % burstCaptures = [txWaveformAWGN;txWaveformAWGN]; %-----> 波形复制，为信号解码准备
83 — step(hsa,burstCaptures); %-----> 显示信号的频谱
84
85 % %5.2. 接收参数设置
86 % samplesPerFrame = length(txWaveform); %-----> 单帧采样点数
87 % rxSamplesPerFrame = samplesPerFrame*2; %-----> 波形复制后，单帧采样点数
88
89 % %5.3. 数据包处理
90 — overSampleFactor = 1; %-----> USRP预录数据图像恢复
91 — [rxBit, offsetLLTF, pktOffset, packetSeq]=ReceiverProc(MPDU_Param, nonHTcfg, hcd, chanBW, overSampleFactor, burstCaptures);
92
93 %5.4. 图像重构和误码率计算
94 — lengthTxImage = length(txImage); %-----> 计算图像的尺寸
95 — rebuildImage(rxBit, offsetLLTF, pktOffset, packetSeq, numMSDUs, MPDU_Param, txData, lengthTxImage, imsize)

```

```

77 %% 5. 接收机信号处理过程
78 %5.1. 捕获WLAN数据包
79 - load('rxWaveform.mat') %-----> 单帧采样点数
80 - burstCaptures=[rxWaveform;rxWaveform];
81
82 % burstCaptures = [txWaveformAWGN;txWaveformAWGN]; %-----> 波形复制后，单帧采样点数
83 - step(hsa,burstCaptures); %----->
84
85 % %5.2. 接收参数设置
86 % samplesPerFrame = length(txWaveform); %-----> 单帧采样点数
87 % rxSamplesPerFrame = samplesPerFrame*2; %-----> 波形复制后，单帧采样点数
88
89 % %5.3. 数据包处理
90 - overSampleFactor = 1; %-----> USRP预录数据图像恢复
91 - [rxBit,offsetLLTF,pktOffset,packetSeq]=ReceiverProc(MPDU_Param,nonHTcfg,hcd,chanBW,overSampleFactor,burstCaptures);
92
93 %5.4. 图像重构和误码率计算
94 - lengthTxImage = length(txImage); %-----> 计算图像的尺寸
95 - rebuildImage(rxBit,offsetLLTF,pktOffset,packetSeq,numMSDUs,MPDU_Param,txData,lengthTxImage,imshow)

```





```
77 %% 5. 接收机信号处理过程
```

```
78 %5.1. 捕获WLAN数据包
```

```
79 load('rxWaveform.mat') %-
```

```
80 burstCaptures=[rxWaveform
```

```
81
```

```
82 % burstCaptures = [txWavefo
```

```
83 step(hsa, burstCaptures);
```

```
84
```

```
85 % %5.2. 接收参数设置
```

```
86 % samplesPerFrame = length
```

```
87 % rxSamplesPerFrame = samp
```

```
88
```

```
89 % %5.3. 数据包处理
```

```
90 overSampleFactor = 1;
```

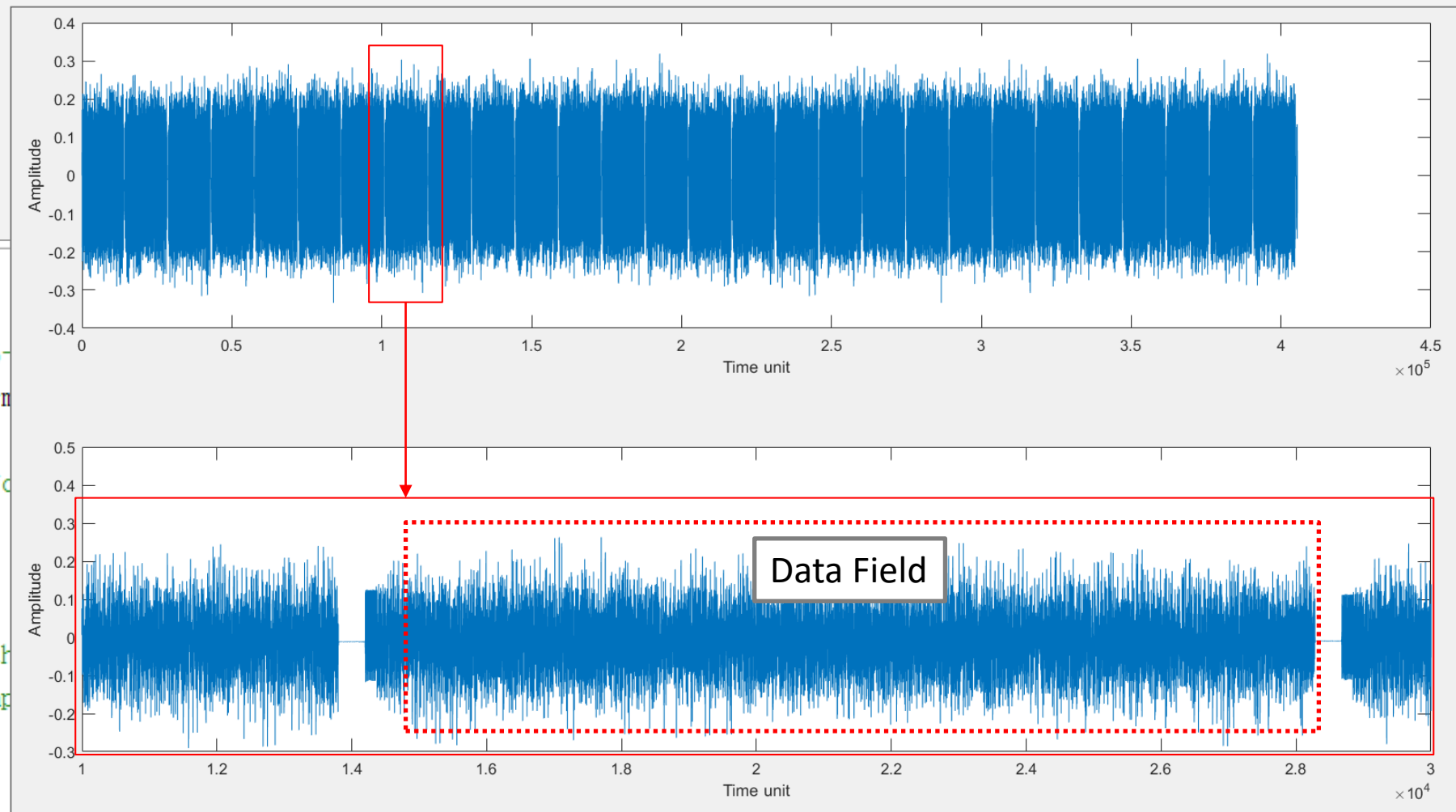
```
91 [rxBit, offsetLLTF, pktOffset, packetSeq]=ReceiverProc(MPDU_Param, nonHTcfg, hcd, chanBW, overSampleFactor, burstCaptures);
```

```
92
```

```
93 %5.4. 图像重构和误码率计算
```

```
94 lengthTxImage = length(txImage); %-----> 计算图像的尺寸
```

```
95 rebuildImage(rxBit, offsetLLTF, pktOffset, packetSeq, numMSDUs, MPDU_Param, txData, lengthTxImage, imsize)
```



%-----> USRP预录数据图像恢复

ReceiverProc

计算图像的尺寸



77 %% 5. 接收机信号处理过程

78 %5.1. 捕获WLAN数据包

```
79 — load('rxWaveform.mat') %-----  
80 — burstCaptures=[rxWaveform;rxWaveform];
```

```
82 % burstCaptures = [txWaveformAWGN;txWaveformAWGN]; %-----  
83 — step(hsa,burstCaptures); %-----
```

85 %%5.2. 接收参数设置

```
86 % samplesPerFrame = length(txWaveform); %-----  
87 % rxSamplesPerFrame = samplesPerFrame*2; %-----
```

89 %%5.3. 数据包处理

```
90 — overSampleFactor = 1; %-----  
91 — [rxBit,offsetLLTF,pktOffset,packetSeq]=ReceiverProc(MPDU_Param,nonHTcfg,hcd,chanBW,overSampleFactor,burstCaptures);
```

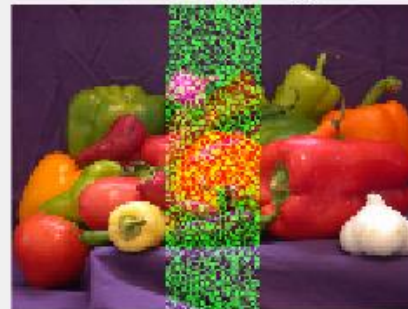
93 %5.4. 图像重构和误码率计算

```
94 — lengthTxImage = length(txImage); %-----  
95 — rebuildImage(rxBit,offsetLLTF,pktOffset,packetSeq,numMSDUs,MPDU_Param,txData,lengthTxImage,imsize)
```

Transmitted Image



Received Image



Packet-12 detected at index 173480

L-SIG check pass
MAC CRC check pass
Estimated CFO: -139.0 Hz

Decoded L-SIG contents:

MCS: 6
Length: 4084
Number of samples in packet: 3520

EVM:

EVM peak: 8.628% EVM RMS: 2.220%

Decoded MAC Sequence Control field contents:
Sequence number:4

Packet-13 detected at index 187960

L-SIG check pass
MAC CRC check pass
Estimated CFO: -157.7 Hz

Decoded L-SIG contents:

MCS: 6
Length: 4084
Number of samples in packet: 3520

EVM:

EVM peak: 10.610% EVM RMS: 2.247%

——> 波形复制后，单帧采样点数

——> USRP预录数据图像恢复

——> 计算图像的尺寸



77 %% 5. 接收机信号处理过程

78 %5.1. 捕获WLAN数据包

```
79 load('rxWaveform.mat') %-----> USRP预录数据图像恢复
80 burstCaptures=[rxWaveform;rxWaveform];
81
82 % burstCaptures = [txWaveformAWGN;txWaveformAWGN];
83 step(hsa,burstCaptures); %-----> 计算图像的尺寸
```

84 %%5.2. 接收参数设置

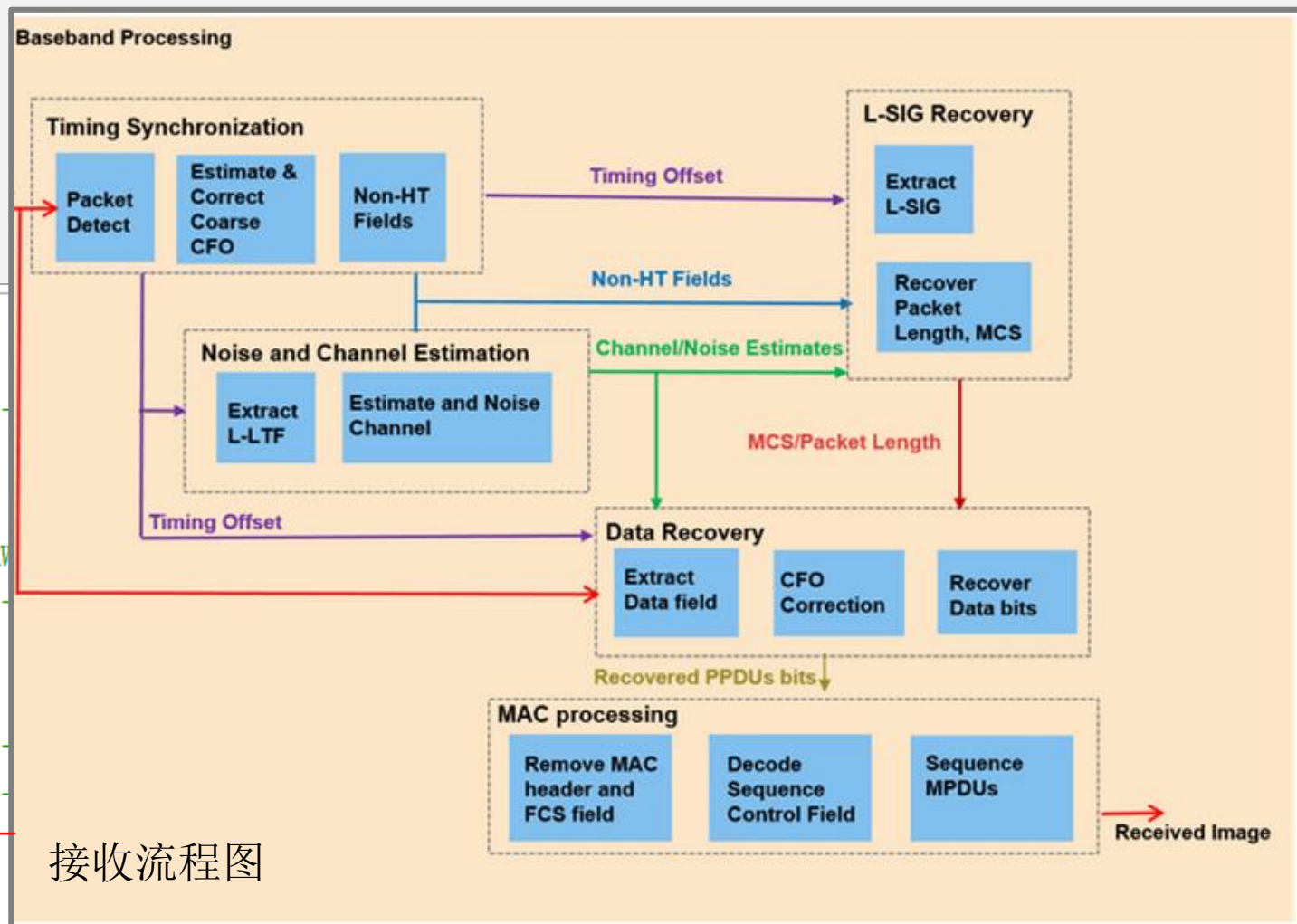
```
85
86 % samplesPerFrame = length(txWaveform); %-----> 计算图像的尺寸
87 % rxSamplesPerFrame = samplesPerFrame*2; %-----> 计算图像的尺寸
```

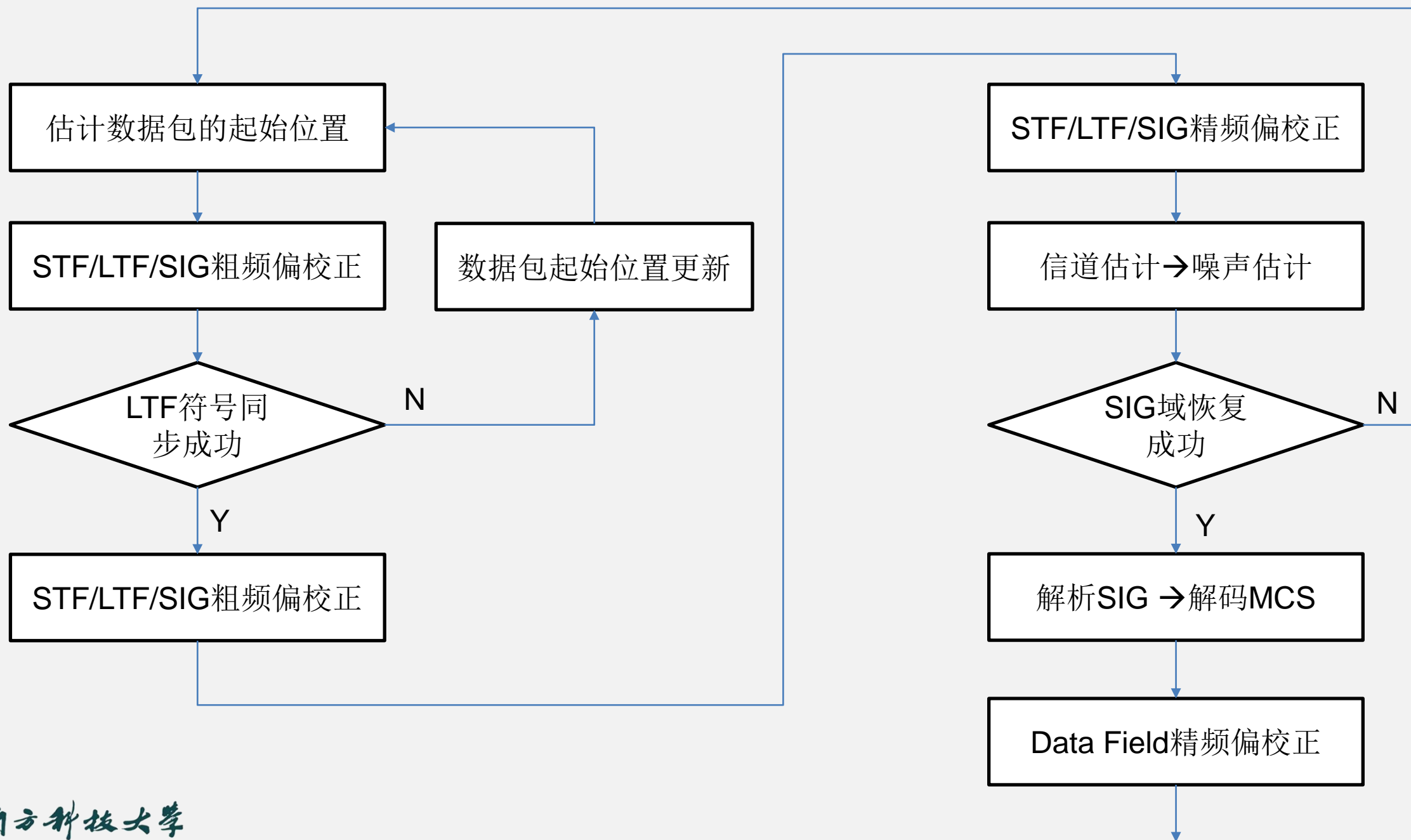
88 %%5.3. 数据包处理

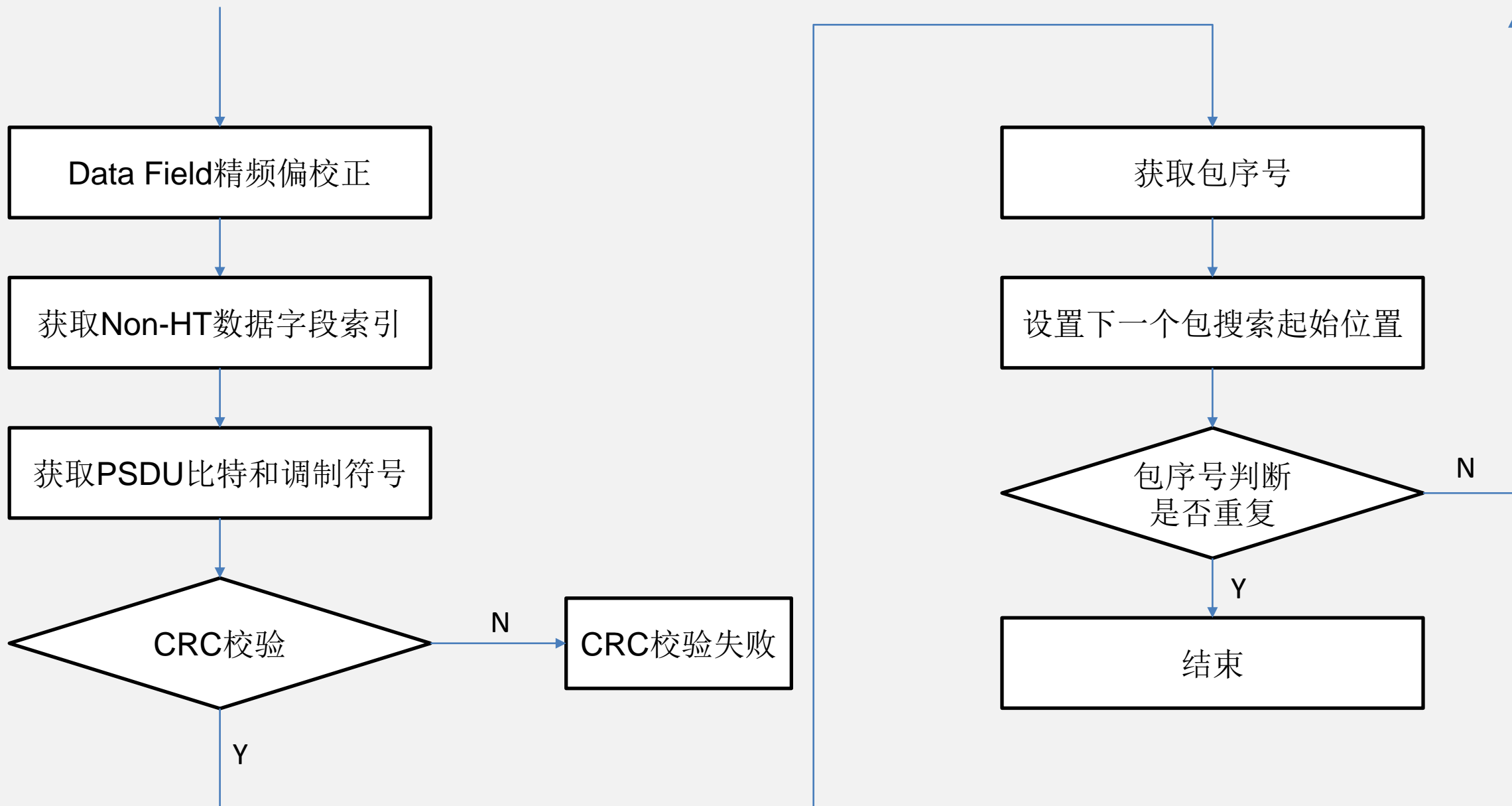
```
89 overSampleFactor = 1;
90
91 [rxBit, offsetLLTF, pktOffset, packetSeq]=ReceiverProc(MPDU_Param, nonHTcfg, hcd, chanBW, overSampleFactor, burstCaptures);
```

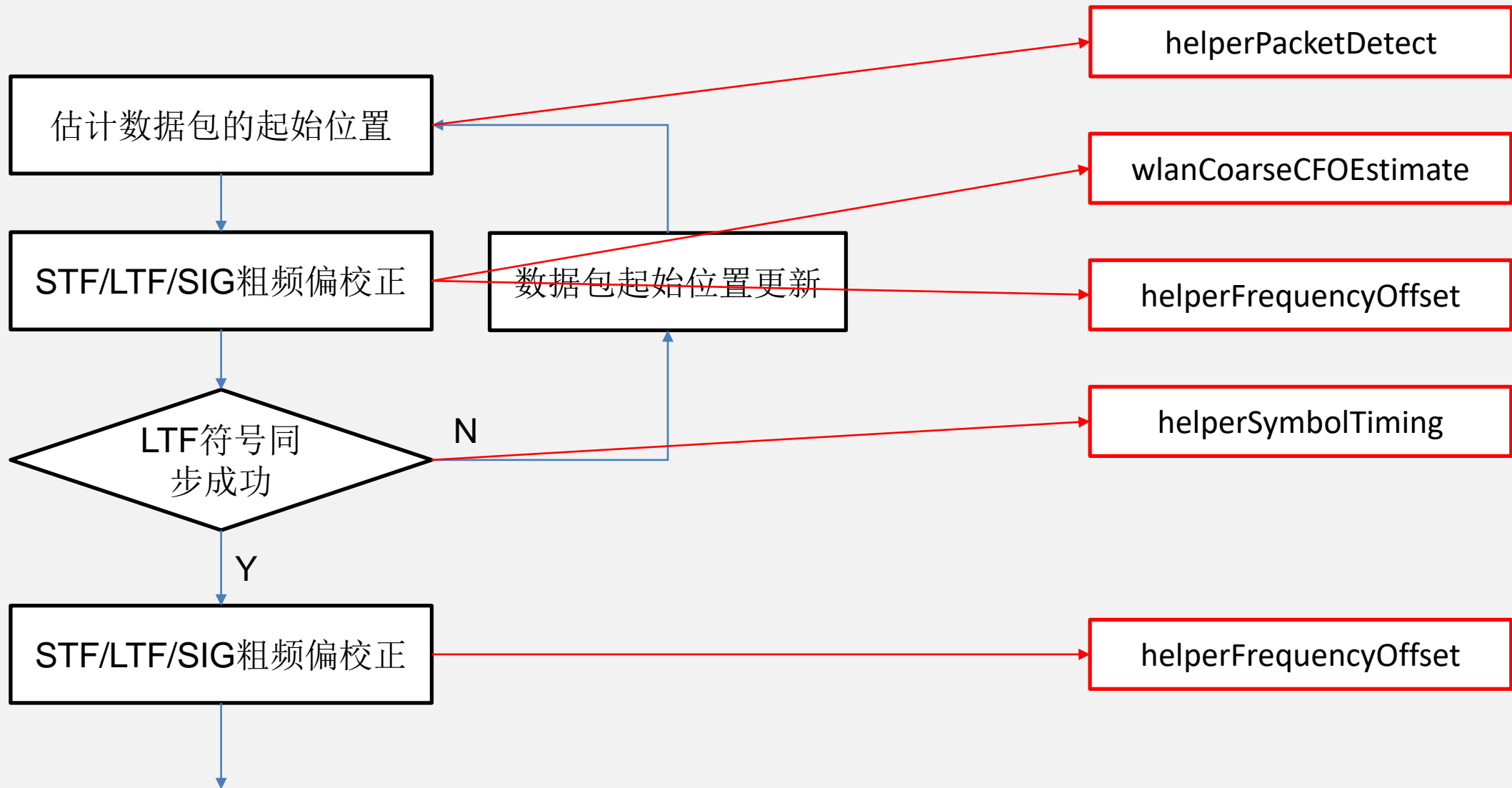
92 %5.4. 图像重构和误码率计算

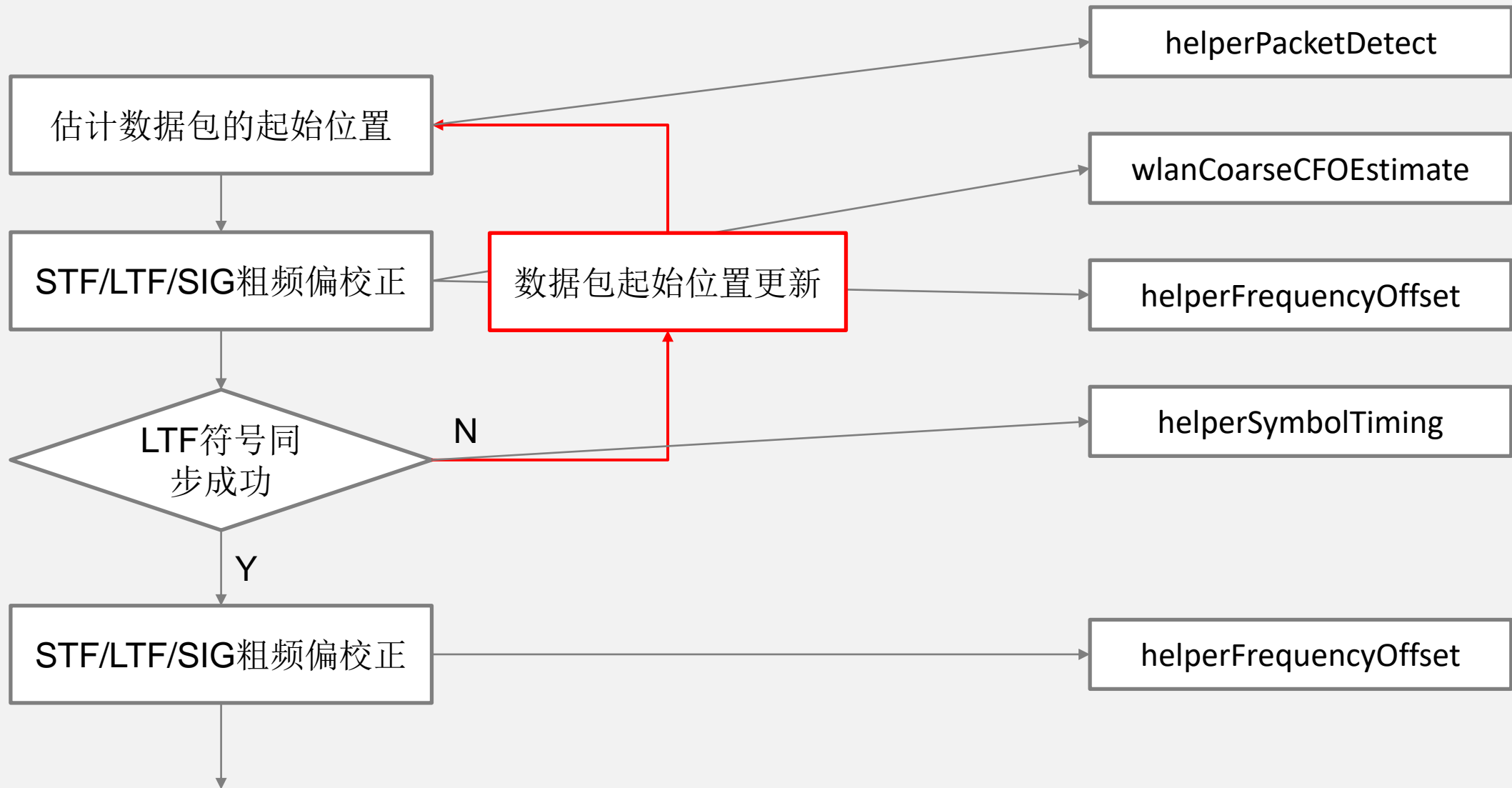
```
93
94 lengthTxImage = length(txImage); %-----> 计算图像的尺寸
95 rebuildImage(rxBit, offsetLLTF, pktOffset, packetSeq, numMSDUs, MPDU_Param, txData, lengthTxImage, imsize)
```

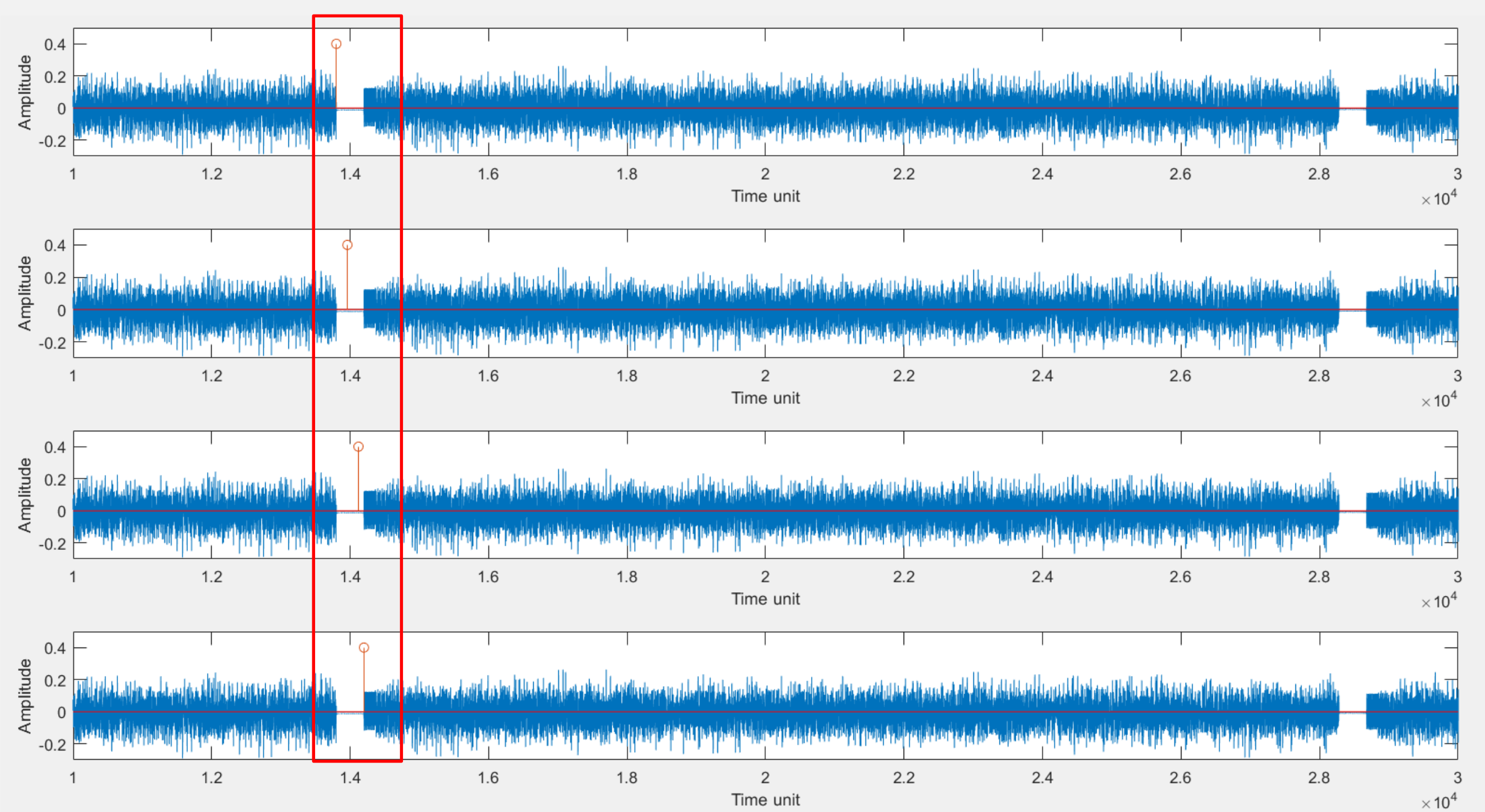


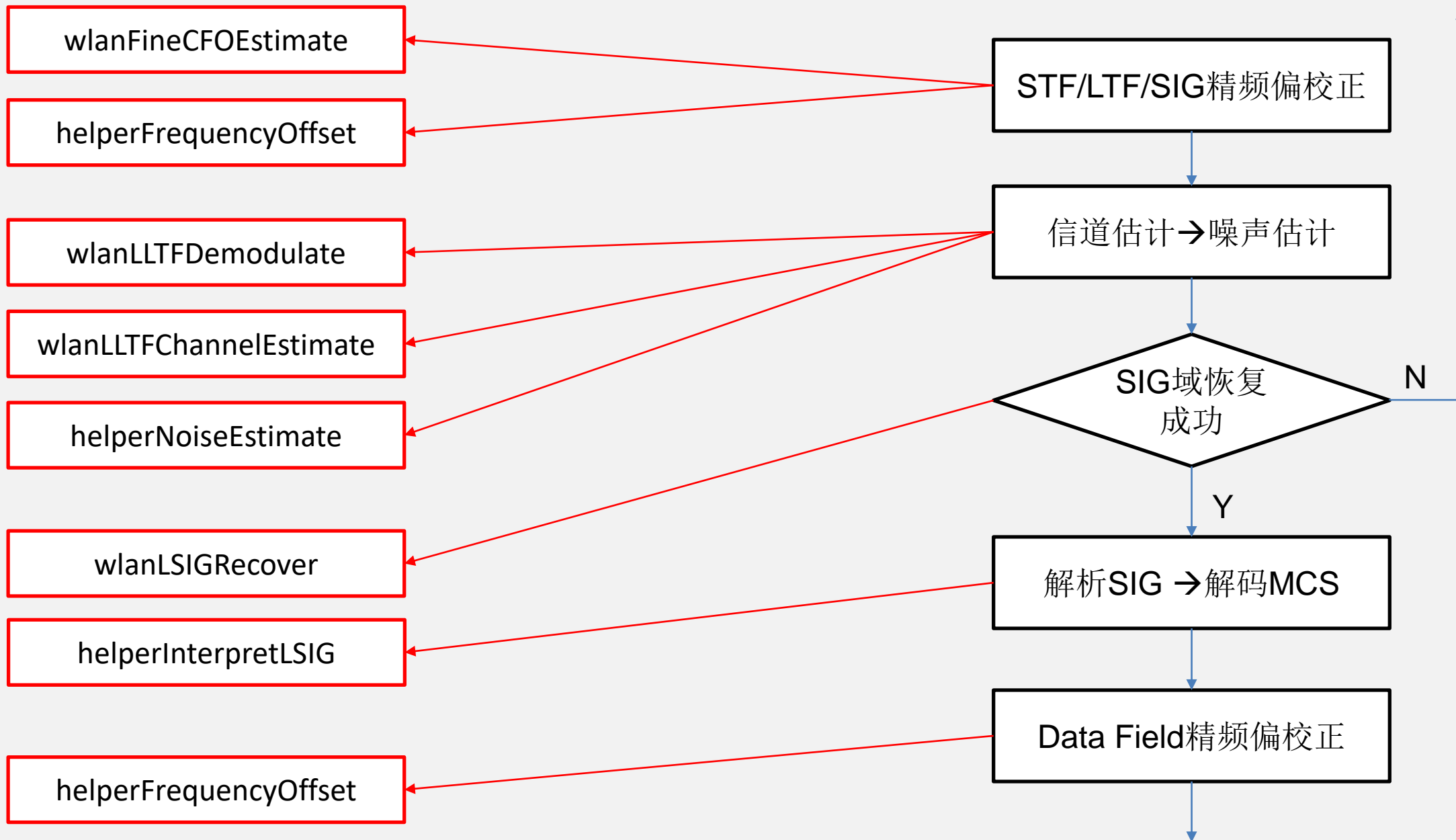


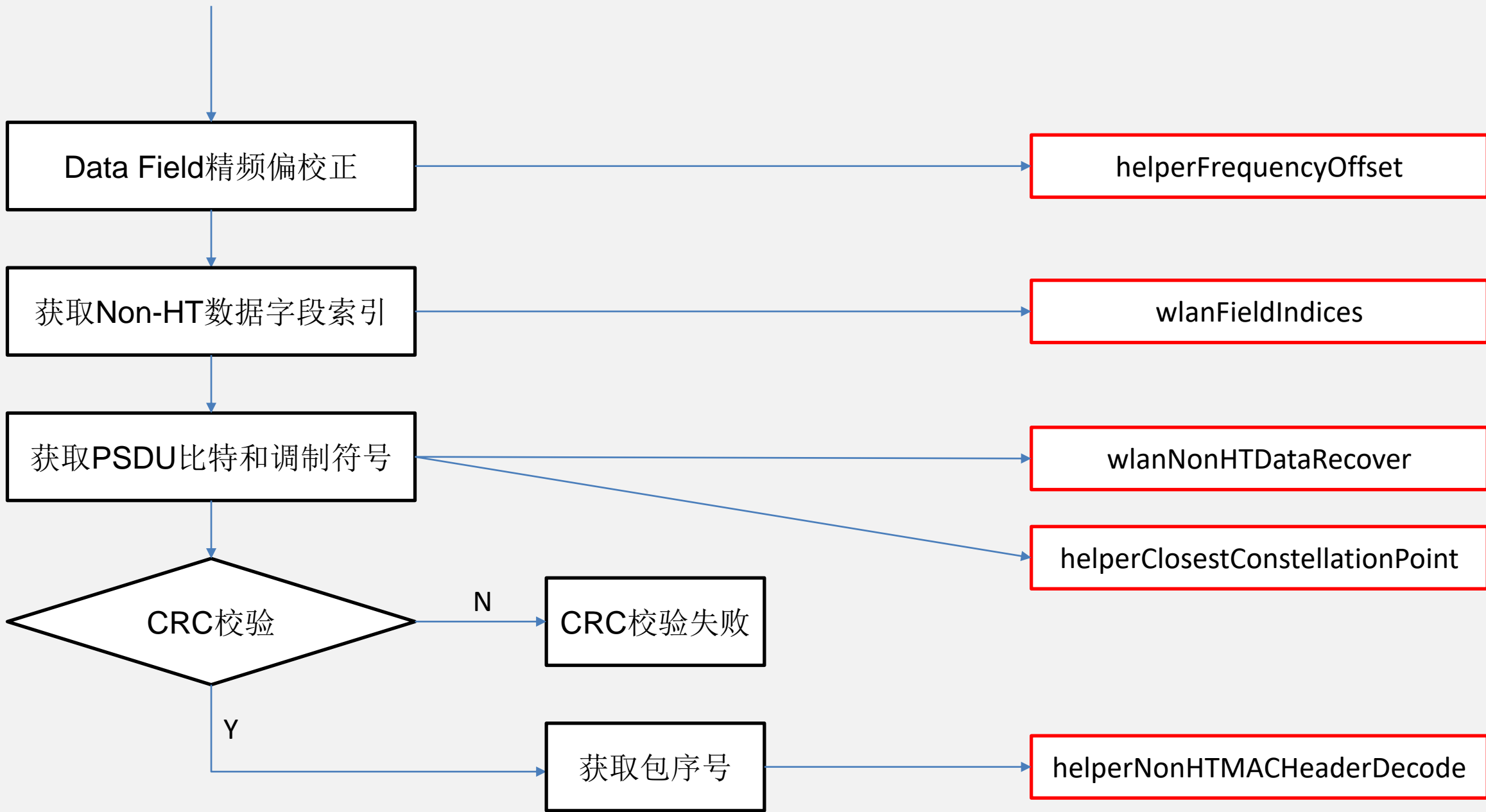


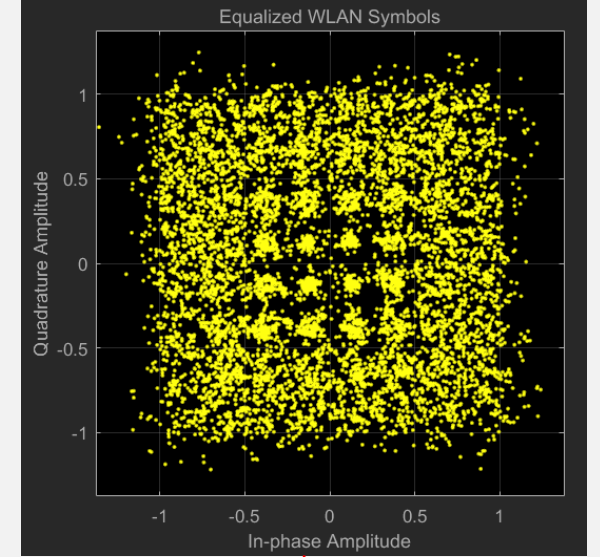
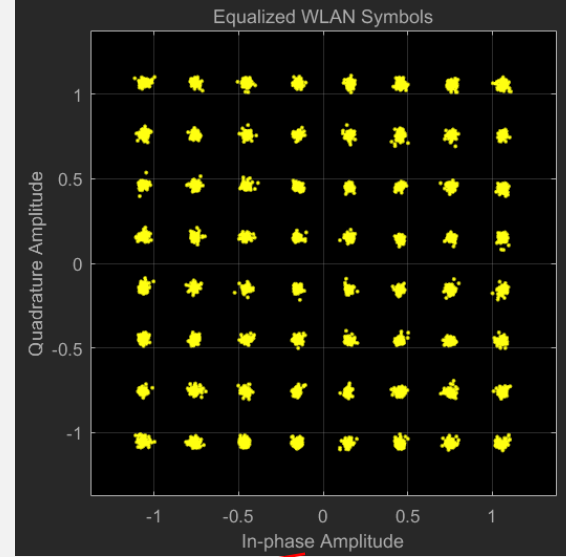
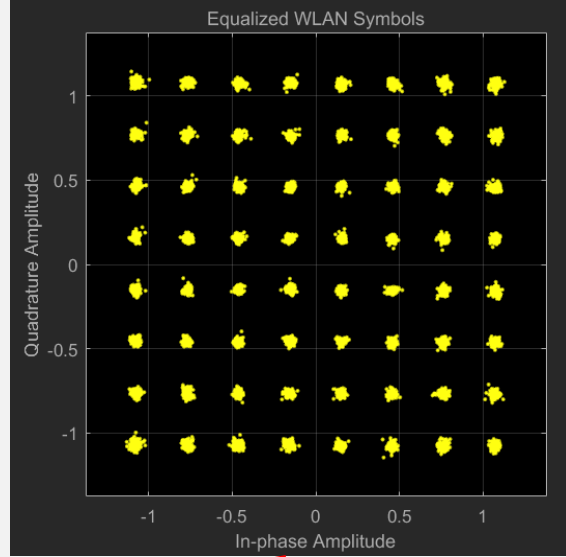
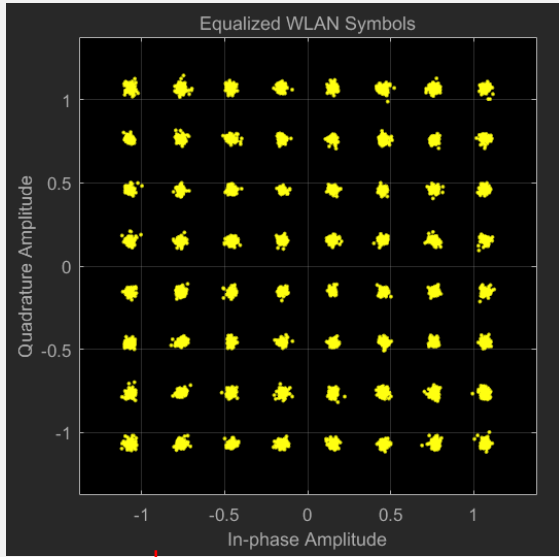












packetSeq =

7 8 9 10 11 12 13 0 1 2 3 4 5 255 7

Packet-14 detected at index 202440

L-SIG check pass

MAC CRC check fail

Estimated CFO: -6091.1 Hz

Decoded L-SIG contents:

MCS: 6

Length: 4084

Number of samples in packet: 3520

EVM:

EVM peak: 39.304% EVM RMS: 15.038%

Decoded MAC Sequence Control field contents:

Sequence number: 255

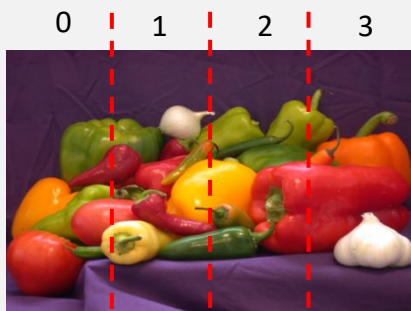


```

- for ind=0:numMSDUs-1
-
-     % Extract bits for each MPDU
-     frameBody = txData(ind*msduBits+1:msduBits*(ind+1),:);
-
-     % Generate MPDU header bits
-     mpduHeader = helperNonHTMACHeader(mod(numFragment,16),mod(ind,4096));
-
-     % Create MPDU with header, body and FCS
-     psdu = step(fcsGen,[mpduHeader;frameBody]);
-
-     % Concatenate PSDUs for waveform generation
-     data(lengthMPDU*ind+1:lengthMPDU*(ind+1)) = psdu;
-
- end

```

Packet sequence

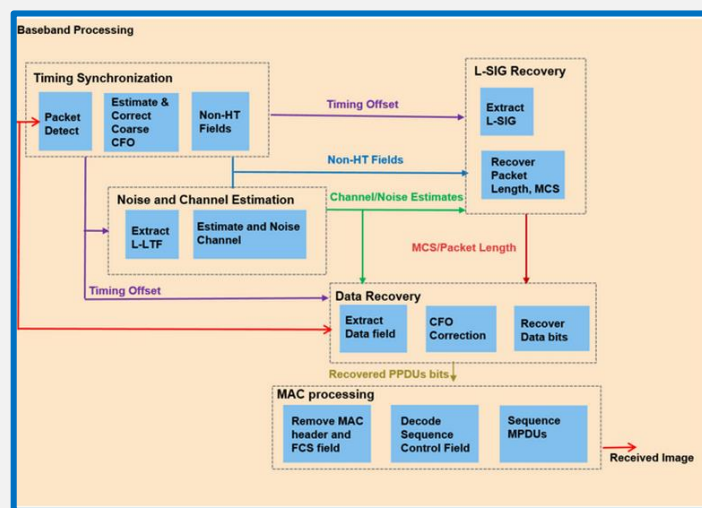
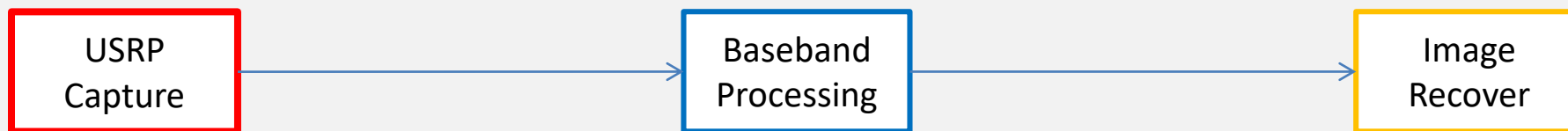


```

- function macHeader = helperNonHTMACHeader(fragNum, seqNum)
-
-     type = 2;    % Data frame type 2 (10)
-     subtype = 0; % Data subtype 0 (00)
-     % Create MPDU header
-     mac = struct;
-     frameControl = getFrameControl(type, subtype);
-     fields = fieldnames(frameControl);
-     frameControlBits = [];
-     for i=1:numel(fields)
-         frameControlBits = [frameControlBits frameControl.(fields{i})];
-     end
-     mac.FrameControl = bi2de(reshape(frameControlBits,8,[ ]).').'; % 2 octets
-     mac.Duration = [0 0]; % Duration of frame for NAV (2 octets)
-     mac.Address1 = [0 0 0 0 0 0]; % Destination address (6 octets)
-     mac.Address2 = [0 0 0 0 0 0]; % Station address (6 octets)
-     mac.Address3 = [0 0 0 0 0 0]; % BSSID (6 octets)
-     seq = getSequenceControl(fragNum, seqNum);
-     mac.Sequence = seq; % Sequence control (2 octets)
-     mac.Address4 = [0 0 0 0 0 0]; % 6 octets
-     mac.QoS = [0 0]; % 2 octets
-     % Convert mac header structure to bit vector
-     macHeader = octetStruct2bits(mac);
- end

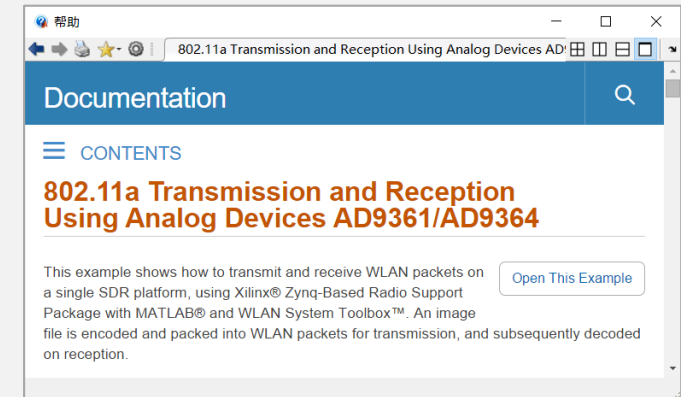
```


Reception Process



Assignments

- Read the example '**802.11a Transmission and Reception Using Analog Devices AD9361/AD9364**' in WLAN System Toolbox.
- Explain the functions of the following six subcomponents respectively,
 - (1) helperPacketDetect.m
 - (2) wlanCoarseCFOEstimate.m
 - (3) helperFrequencyOffset.m
 - (4) helperSymbolTiming.m
 - (5) helperInterpretLSIG.m
 - (6) wlanNonHTDataRecover.m
-
- Image recover with Pre-Recorded data and further analyze the results.



- Question ?

