Review—1

createBitstream

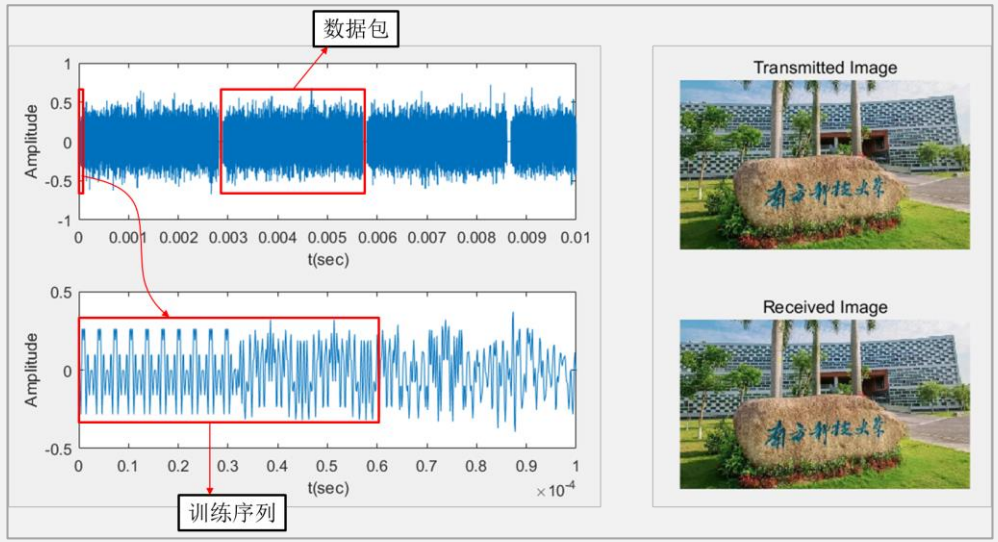
createPSDU

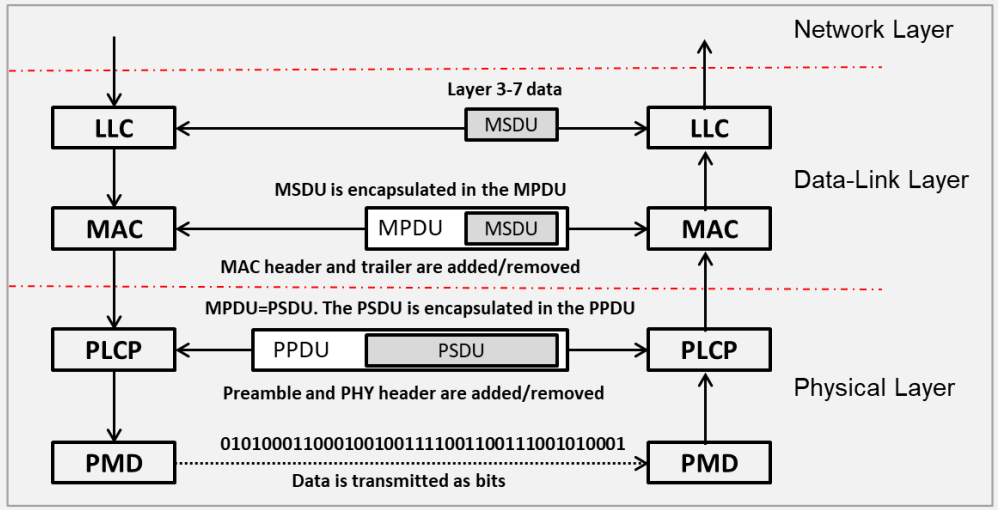
createTxWaveform

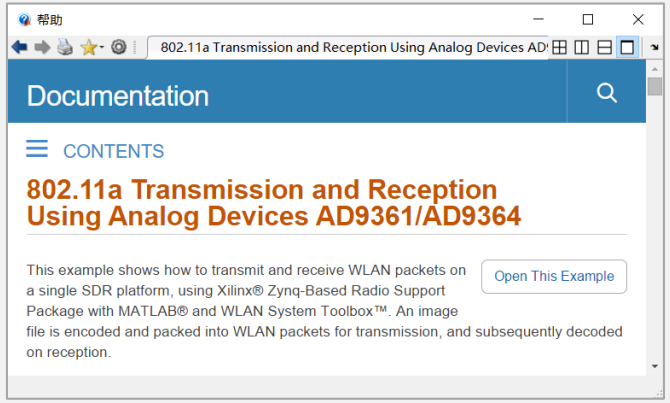
createAWGNChannel

receiverProc

reBuildImage

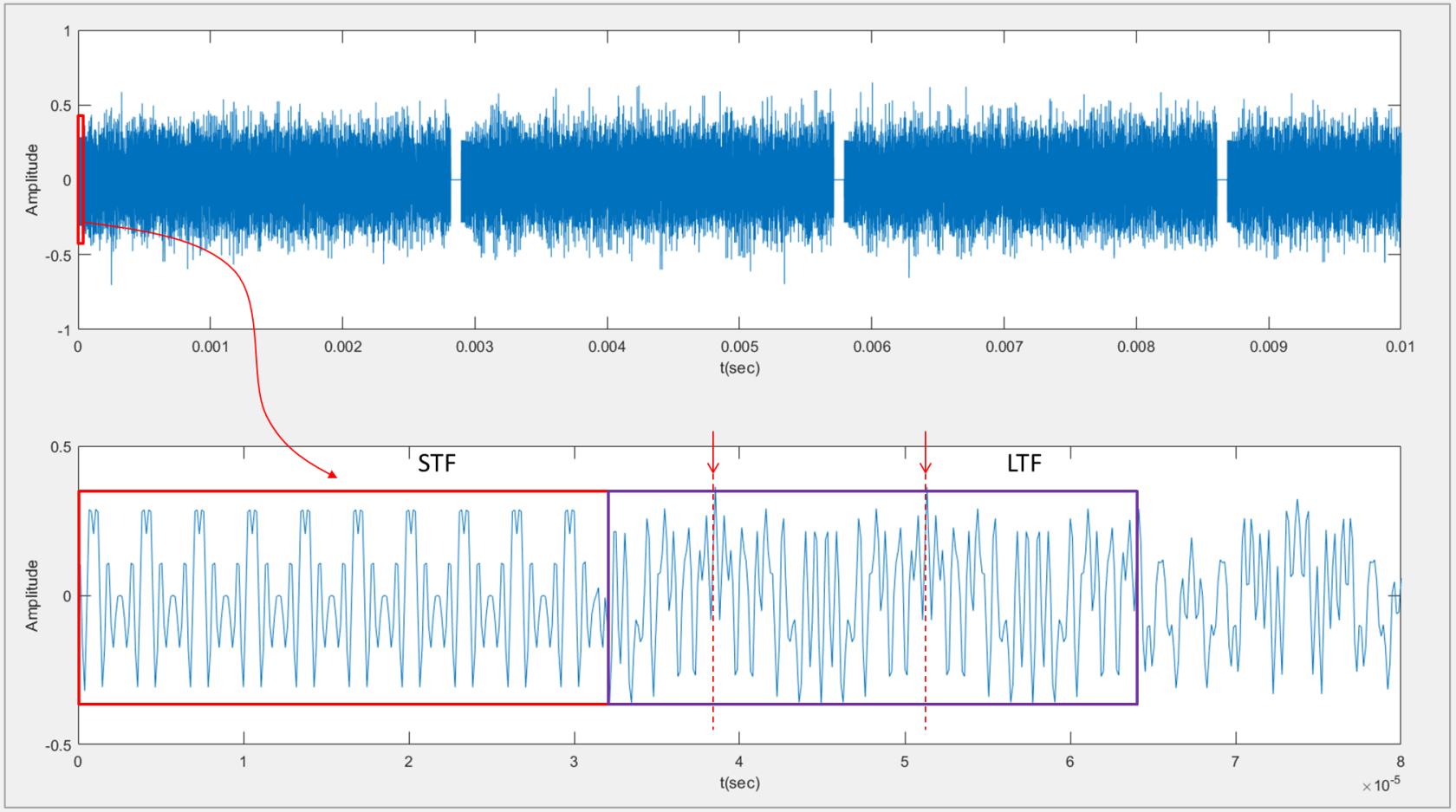






南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Review—2

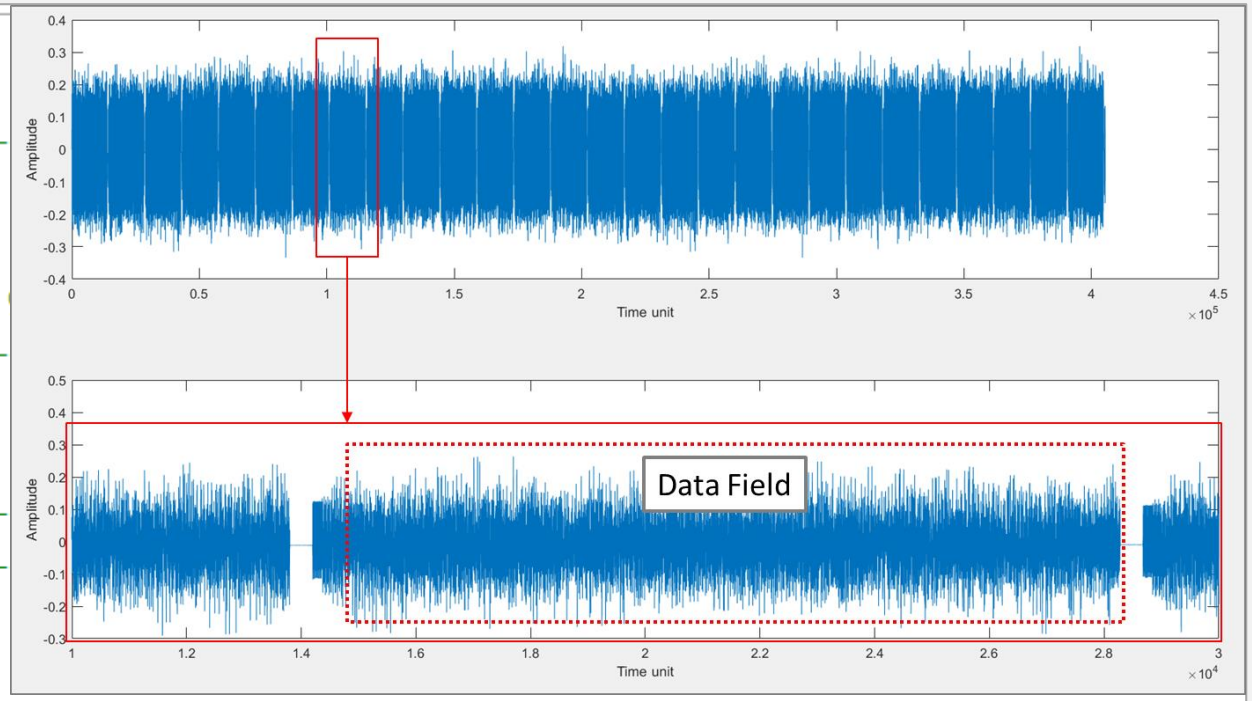


Transmitted Image

Received Image

Review—3

```
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];
81
82      % burstCaptures = [txWaveformAWGN;txWaveformAWGN];
83 —      step(hsa,burstCaptures);     %--------------------
84
85      % %5.2. 接收参数设置
86      %  samplesPerFrame = length(txWaveform);  %---------
87      %  rxSamplesPerFrame = samplesPerFrame*2; %---------
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)
```




前沿通信系统设计（32学时）
1 WiFi通信系统（9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（4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（2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、MICAZ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（1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

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Communication Systems Design

Lab 5: 802.11a Image Transmission and Reception

(Part 4)

Dr. Wu Guang

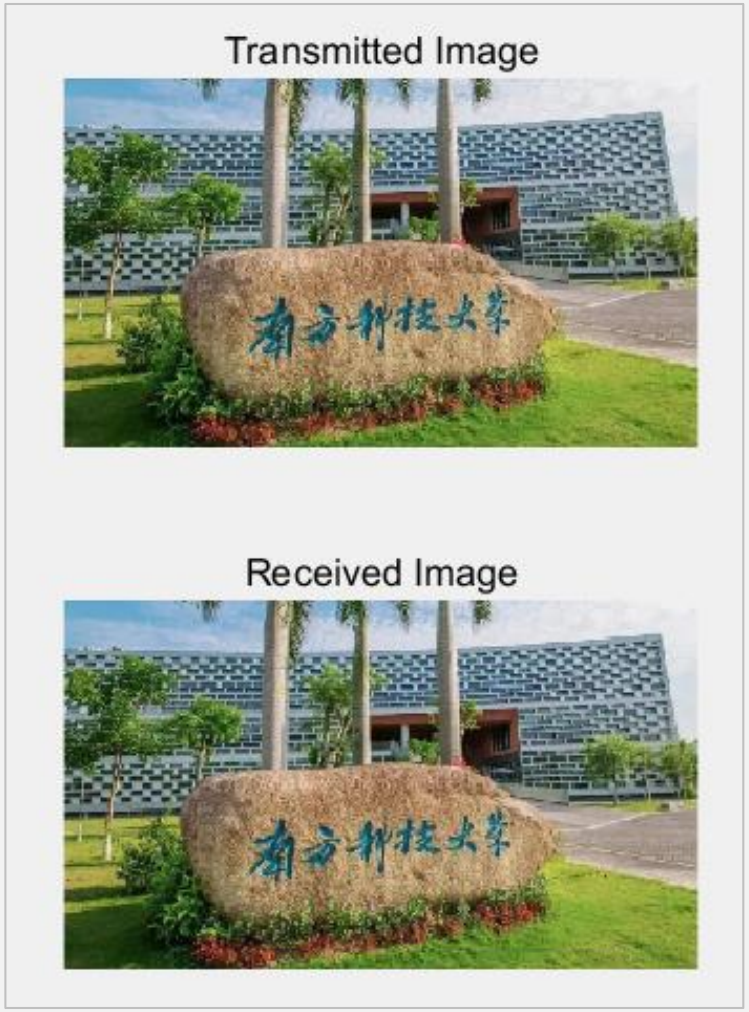
wug@sustech.edu.cn

Electrical & Electronic Engineering

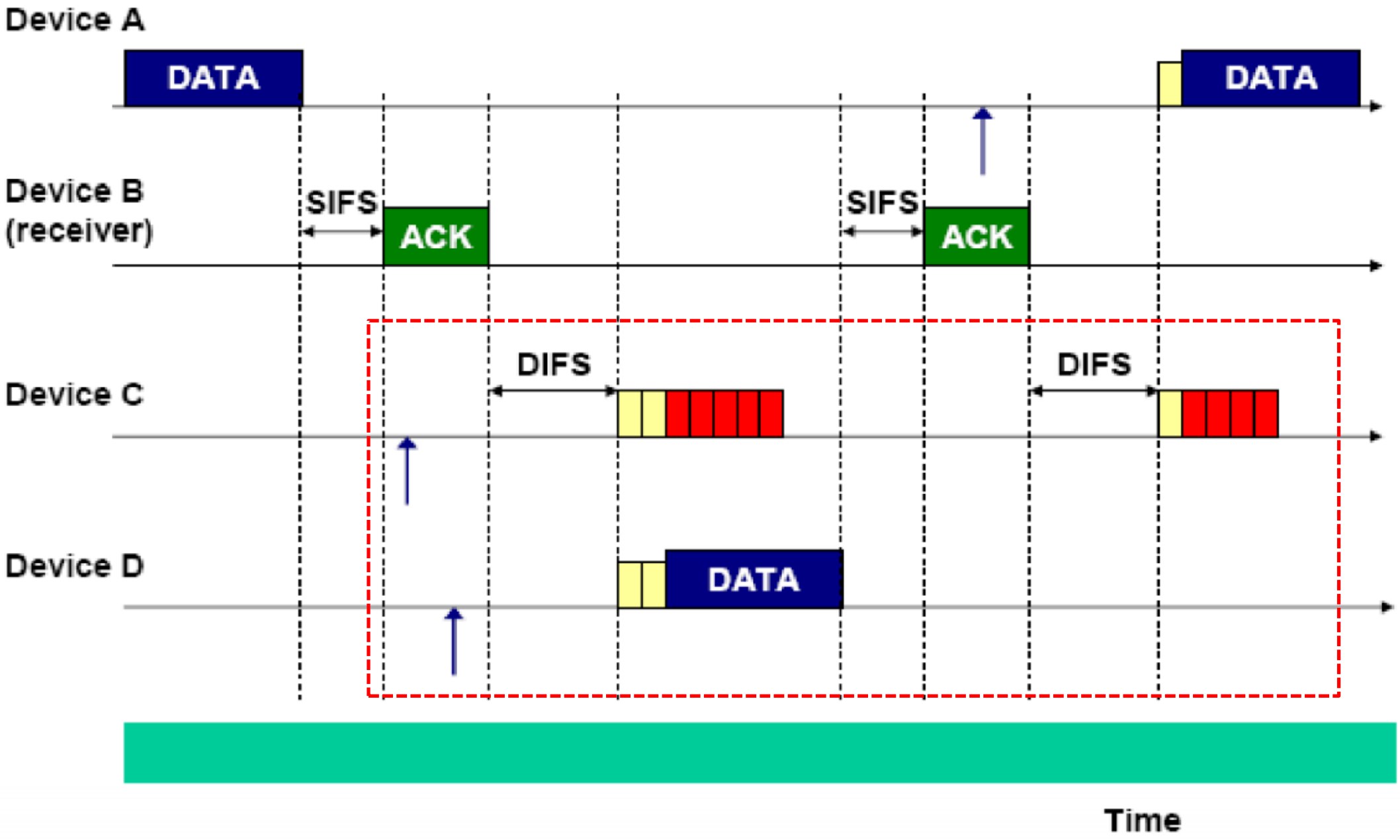
Southern University of Science and Technology

How to build a WiFi packet ?

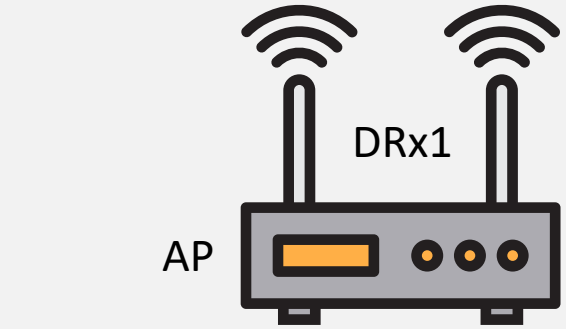
- How to pack the information bits?
- How to design the training sequences ?
- How to decode the data field ?
- How to coordinate the multiple access ?



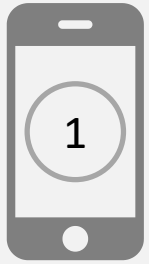
How to coordinate the multiple access ?


Device A
DATA
DATA
Device B
(receiver)
SIFS
ACK
SIFS
ACK
Device C
DIFS
DIFS
Device D
DATA
Time

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance

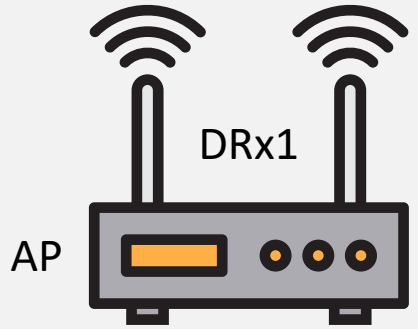


802.11b

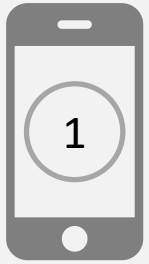


DTx1

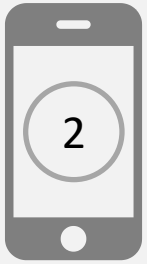
Transmitted Image



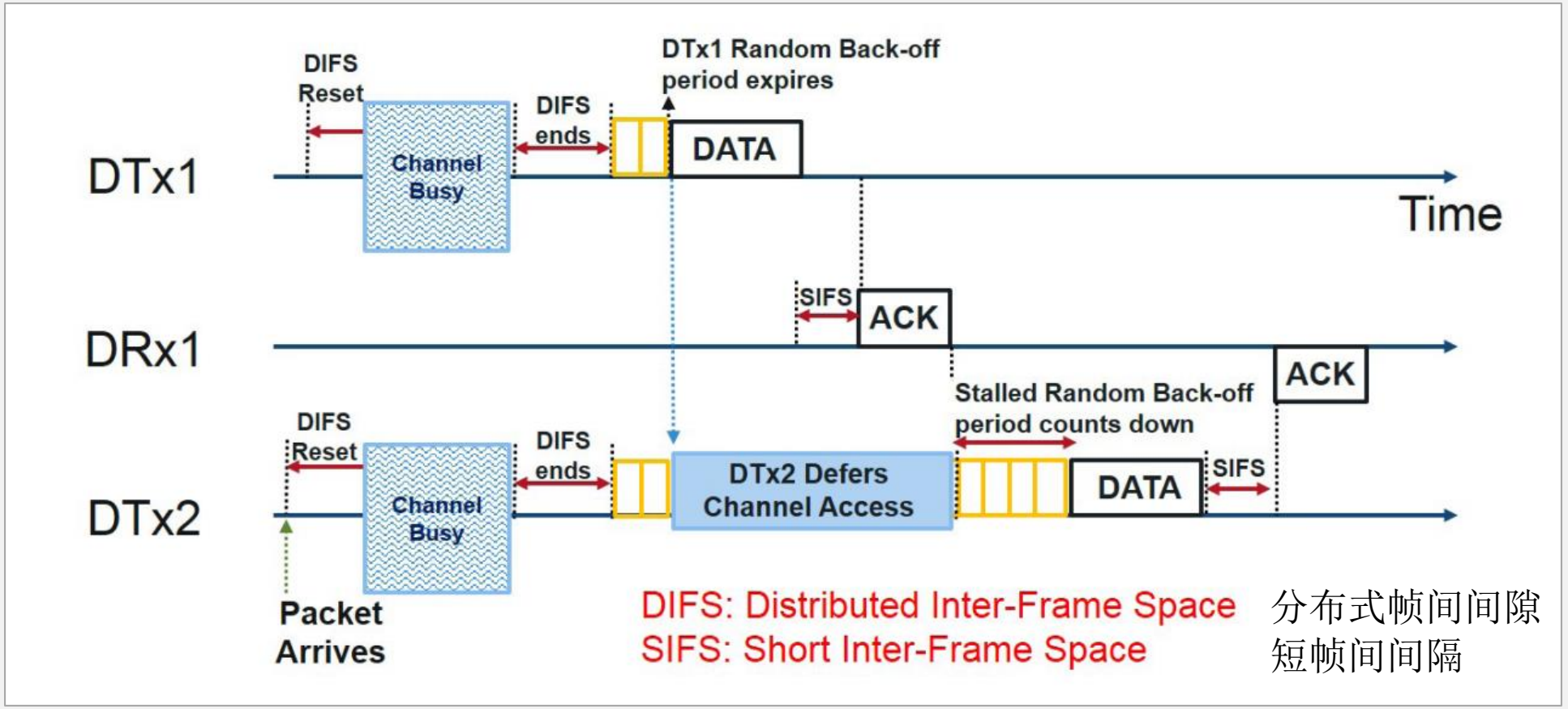
802.11b



DTx1



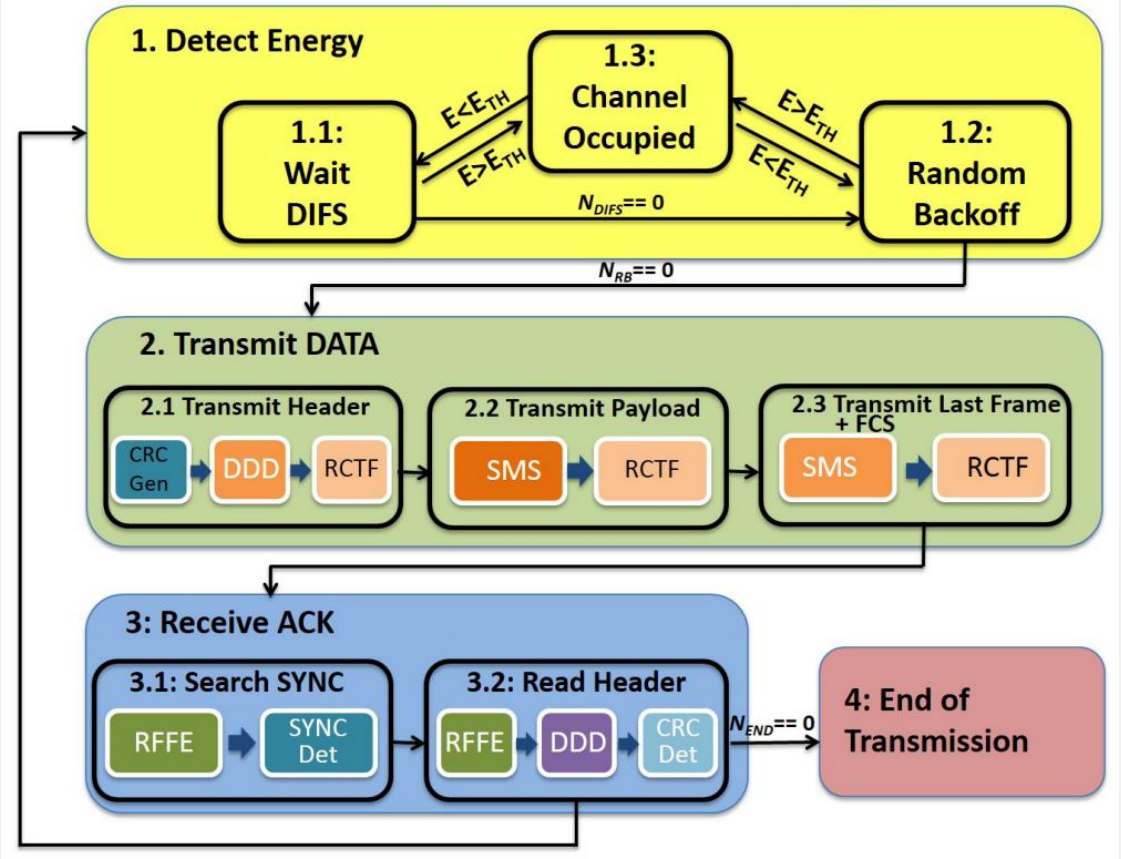
DTx2


DTx1
DRx1
DTx2
DIFS
Reset
Channel
Busy
DIFS
ends
DTx1 Random Back-off
period expires
DATA
Time
SIFS
ACK
ACK
Stalled Random Back-off
period counts down
DIFS
Reset
Channel
Busy
DIFS
ends
DTx2 Defers
Channel Access
DATA
SIFS
Packet
Arrives
DIFS: Distributed Inter-Frame Space
SIFS: Short Inter-Frame Space
分布式帧间间隙
短帧间间隔

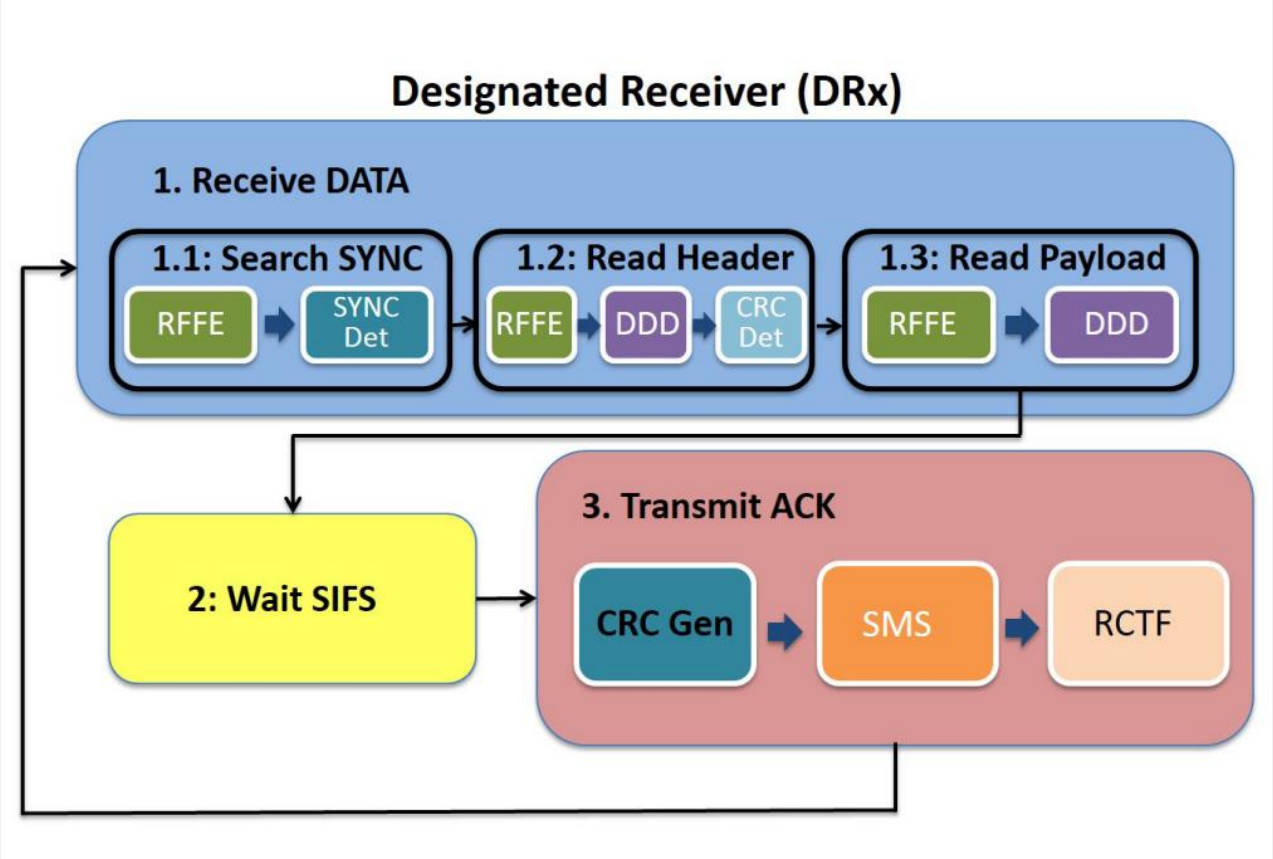
High-Level System Design of IEEE 802.11b Standard-Compliant Link Layer for MATLAB-based SDR

ABSTRACT Software-defined radio (SDR) allows the unprecedented levels of flexibility by transitioning the radio communication system from a rigid hardware platform to a more user-controlled software paradigm. However, it can still be time-consuming to design and implement such SDRs as they typically require thorough knowledge of the operating environment and a careful tuning of the program. In this paper, our contribution is the design of a bidirectional transceiver that runs on the commonly used USRP platform and implemented in MATLAB using standard tools like MATLAB Coder and MEX to speed up the processing steps. We outline strategies on how to create a state-action-based design, wherein the same node switches between transmitter and receiver functions. Our design allows the optimal selection of the parameters toward meeting the timing requirements set forth by various processing blocks associated with a differential binary phase shift keying physical layer and CSMA/CA/ACK MAC layer, so that all the operations remain functionally compliant with the IEEE 802.11b standard for the 1 Mb/s specification. The code base of the system is enabled through the Communications System Toolbox and incorporates channel sensing and exponential random back-off for contention resolution. The current work provides an experimental testbed that enables the creation of new MAC protocols starting from the fundamental IEEE 802.11b standard. Our design approach guarantees consistent performance of the bi-directional link, and the three-node experimental results demonstrate the robustness of the system in mitigating packet collisions and enforcing fairness among nodes, making it a feasible framework in higher layer protocol design.

DATA-ACK Mode


1. Detect Energy
1.1: Wait DIFS
1.3: Channel Occupied
1.2: Random Backoff
E<E_TH
E>E_TH
E>E_TH
E<E_TH
N_DIFS== 0
N_RB== 0
2. Transmit DATA
2.1 Transmit Header
CRC Gen
DDD
RCTF
2.2 Transmit Payload
SMS
RCTF
2.3 Transmit Last Frame + FCS
SMS
RCTF
3: Receive ACK
3.1: Search SYNC
RFFE
SYNC Det
3.2: Read Header
RFFE
DDD
CRC Det
N_END== 0
4: End of Transmission

DTx


Designated Receiver (DRx)
1. Receive DATA
1.1: Search SYNC
RFFE
SYNC Det
1.2: Read Header
RFFE
DDD
CRC Det
1.3: Read Payload
RFFE
DDD
2: Wait SIFS
3. Transmit ACK
CRC Gen
SMS
RCTF

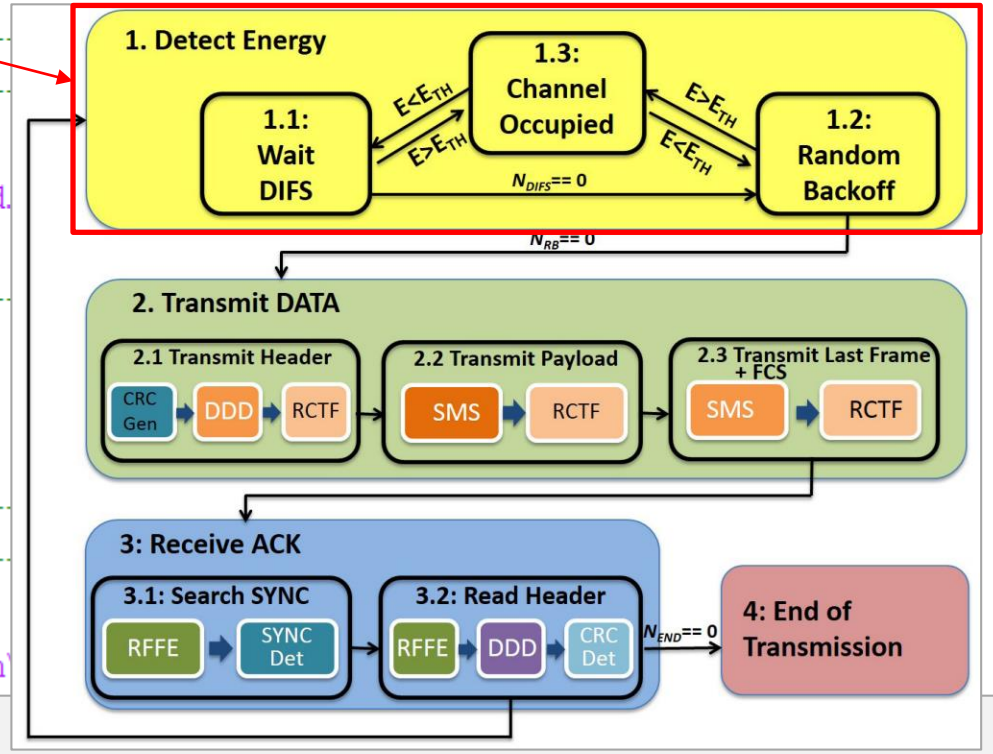
DRx

11:Detect Energy

12:Transmit DATA

13:Receive ACK

```
45 —    tic;
46 —    while ~fe          %-------------------------------------------------------------> 循环结束标志
47 —        smt = st/uint8(10);          %-------------------------------------------------> 设置发射机起始状态：能量检测状态
48 —        if (smt==uint8(11))    %-------------------------------------------------> 如果发射机处于能量检测状态
49 —            st = dtxMACLayerSlot(st,frt);    %-------------------------------------> MAC层检测DIFS时间，下一个状态：发数据
50 —
51 —
52 —        elseif (smt==uint8(15)) %DTxStateTransmitRTS  ----------------------
53 —            f8t =  logical(true(1));  %-----------------------------------
54 —            if (f8t)
55 —                if (vm), fprintf(1,'@%5.2f: 802.11b RTS Packet Transmitted.
56 —                end
57 —                st = uint8(161);   %-----------------------------------
58 —            end
59 —
60 —
61 —        elseif (smt==uint8(16)) %prm.DTxStateRxCTS ----------------------
62 —            faf = logical(true(1));  %-----------------------------------
63 —            if (faf)
64 —                if (vm), fprintf(1,'@%5.2f: 802.11b CTS Packet Received.\n'
```

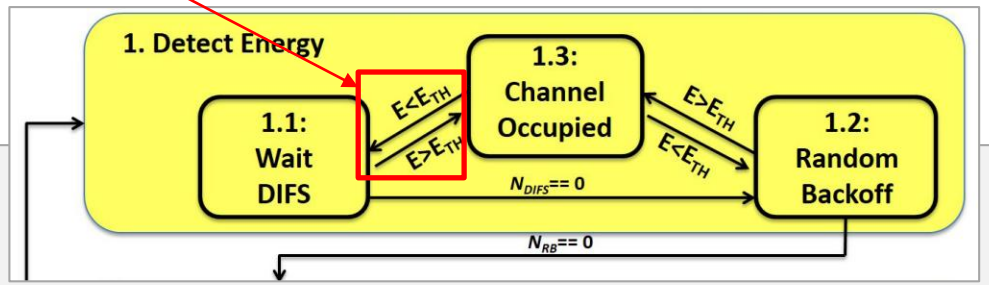


DIFS State

Random Backoff

Switch state

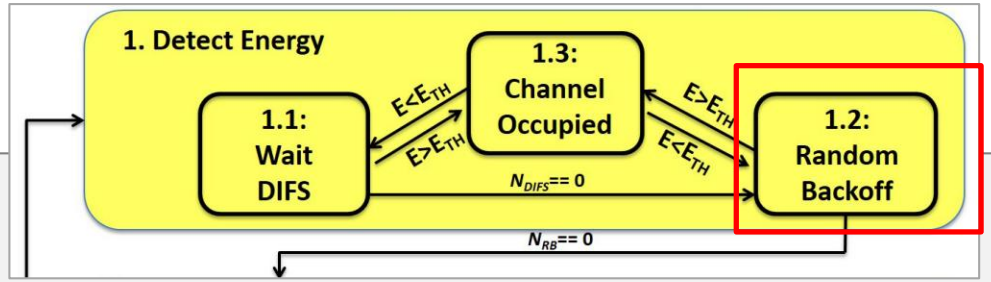
```
57 —  if (vcsFlag == 0)  %-------------------------------------------------------------------------> 不启用RTS/CTS机制
58 —      if (st == uint8(111))
59 —          if (vm), fprintf('Entering DIFS state..\n'); end %-------------------------------------> 进入DIFS状态
60 —                SlotCount = 1;
61
62 —          while SlotCount < DIFS_Slots    %-------------------------------------------------------> 循环检测DIFS_Slots时间
63 —                  df = rand(1,25);    %-------------------------------------------------------------> 13%的概率信道忙
64 —              if (sum(abs(df).^2) > energyThreshold)  %-----------------------------------------> 检测到信道忙
65 —                  if (vm), fprintf('Energy detected in DIFS state, Backing off!!\n'); end
66 —                  SlotCount=1;        %-------------------------------------------------------------> 开始退避
67 —              end
68 —                  SlotCount=SlotCount+1;
69 —          end
70
71 —              if (vm), fprintf('...DIFS ends.\n'); end
72 —      end
```



1: Binary Exponential Backoff

0: Binary Linear Backoff

```
74 —      if (vm), fprintf('Entering Random Backoff state..\n'); end
75 —
76 —          SlotCount = 1;
77 —      while SlotCount < BEB_Slots
78 —          df = rand(1,25);   %-------------------------------------------------------------> 13%的概率信道忙
79 —         if (sum(abs(df).^2) > energyThreshold)    %-----------------------------------------> 检测到信道忙
80 —             if (vm), fprintf('Energy detected in Random Backoff state, Backing off!!\n'); end
81 —             BEB_FreezeSlot = SlotCount;
82 —             SlotCount = 1;
83 —             BEB_Slots = BEB_Slots - BEB_FreezeSlot;   %-------------------------------------> 下次退避总时间减少
84 —         end
85 —             SlotCount=SlotCount+1;
86 —      end
87 —      if (vm), fprintf('...Random Backoff ends.\n'); end
```



```
Entering DIFS state..
Energy detected in DIFS state, Backing off!!
...DIFS ends.
Entering Random Backoff state..
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
...Random Backoff ends.
@ 0.01: 802.11b DATA Packet #1 Transmitted.
@ 0.01: 802.11b ACK Packet #1 Received.
```

```
Entering DIFS state..
...DIFS ends.
Entering Random Backoff state..
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
...Random Backoff ends.
@ 0.02: 802.11b DATA Packet #2 Transmitted.
@ 0.02: 802.11b ACK Packet #2 Received.
```

```
Entering DIFS state..
...DIFS ends.
Entering Random Backoff state..
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
...Random Backoff ends.
@ 0.02: 802.11b DATA Packet #4 Transmitted.
@ 0.02: 802.11b ACK Packet #4 Received.
```

```
Entering DIFS state..
...DIFS ends.
Entering Random Backoff state..
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
...Random Backoff ends.
@ 0.02: 802.11b DATA Packet #3 Transmitted.
@ 0.02: 802.11b ACK Packet #3 Received.
```

```
Entering DIFS state..
...DIFS ends.
Entering Random Backoff state..
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
Energy detected in Random Backoff state, Backing off!!
...Random Backoff ends.
@ 0.03: 802.11b DATA Packet #5 Transmitted.
@ 0.03: 802.11b ACK Packet #5 Received.
```

RTS-CTS-DATA-ACK Mode

11:Detect Energy

15:Transmit RTS

16:Receive CTS

```
47 —    smt = st/uint8(10);          %-------------------------------------------------> 设置发射机起始状态：能量检测状态
48 —    if (smt==uint8(11))    %-------------------------------------------------> 如果发射机处于能量检测状态
49 —        st = dtxMACLayerSlot(st,frt);     %-------------------------------------> MAC层检测DIFS时间，下一个状态：发数据
50
51
52 —    elseif (smt==uint8(15)) %DTxStateTransmitRTS  ---------------------------------> 发射机发送RTS
53 —        f8t =  logical(true(1));  %-------------------------------------------------> RTS发送完成
54 —        if (f8t)
55 —            if (vm), fprintf(1,'@%5.2f: 802.11b RTS Packet Transmitted.\n',toc); %---> 输出RTS发送完成消息
56 —            end
57 —            st = uint8(161);    %-------------------------------------------------> 转移到16状态
58 —        end
59
60
61 —    elseif (smt==uint8(16)) %prm.DTxStateRxCTS ------------------------------------> 发射机接收CTS
62 —        faf = logical(true(1));  %-------------------------------------------------> 成功接收CTS
63 —        if (faf)
64 —            if (vm), fprintf(1,'@%5.2f: 802.11b CTS Packet Received.\n\n',toc); end
65 —            st = uint8(121); %prm.DTxStateTransmitHeader  ---------------------------> 转移到数据发送状态
```

vcsChoice = logical(false(1))　　vcsFlag = 0

dtxTestsuite.m | dtxPHYLayer.m | dtxMACLayerSlot.m*

```
88 —    else
89          %Virtual Carrier Sensing
90 —        SlotCount = 1;
91 —        if (vcsFlag == 1)
92 —            while SlotCount < vcs_Slots
93 —                SlotCount = SlotCount+1;
94 —            end
95 —            if (vm), fprintf('Defered Medium Access for NAV Duration - Exiting VCS!!\n'); end
96 —        end
97 —    end
98
99 —    if (vcsChoice == 1)
100 —       st = uint8(151); % Virtual Carrier Sensing;  %----------------------------------------> 启用虚拟载波侦听
101 —   else
102 —       st=uint8(121); % No Virtual Carrier Sensing; %----------------------------------------> 不启用虚拟载波侦听
103 —   end
```

(vm = true) Verbose Mode

```
@ 0.04: 802.11b DATA Packet #1 Transmitted.
@ 0.04: 802.11b ACK Packet #1 Received.

Defered Medium Access for NAV Duration - Exiting VCS!!
@ 0.04: 802.11b RTS Packet Transmitted.
@ 0.04: 802.11b CTS Packet Received.

@ 0.04: 802.11b DATA Packet #2 Transmitted.
@ 0.04: 802.11b ACK Packet #2 Received.

Defered Medium Access for NAV Duration - Exiting VCS!!
@ 0.04: 802.11b RTS Packet Transmitted.
@ 0.04: 802.11b CTS Packet Received.
```

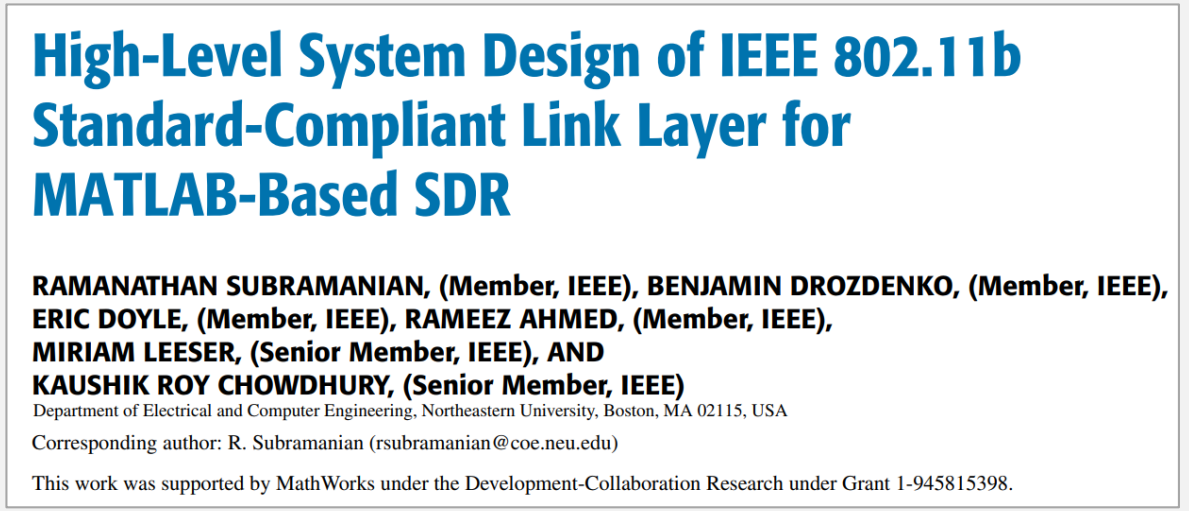
```
@ 0.04: 802.11b DATA Packet #3 Transmitted.
@ 0.04: 802.11b ACK Packet #3 Received.

Defered Medium Access for NAV Duration - Exiting VCS!!
@ 0.04: 802.11b RTS Packet Transmitted.
@ 0.04: 802.11b CTS Packet Received.

@ 0.04: 802.11b DATA Packet #4 Transmitted.
@ 0.04: 802.11b ACK Packet #4 Received.

Defered Medium Access for NAV Duration - Exiting VCS!!
@ 0.04: 802.11b RTS Packet Transmitted.
@ 0.04: 802.11b CTS Packet Received.

@ 0.04: 802.11b DATA Packet #5 Transmitted.
@ 0.04: 802.11b ACK Packet #5 Received.
```

Sketch out:
Flow-chart of the RTS-CTS mode

Assignments

- Read the paper '**High-Level System Design of IEEE 802.11b Standard-Compliant Link Layer for MATLAB-based SDR**'.

- Explain the functions of the following six subcomponents respectively,
  (1) Slot-Time Synchronized Operations
  (2) Designated Transmitter State Machine
  (3) Designated Receiver State Machine
  (4) PHY Layer algorithm
  (5) MAC Layer Design
  (6) Experiment Results



High-Level System Design of IEEE 802.11b Standard-Compliant Link Layer for MATLAB-Based SDR

RAMANATHAN SUBRAMANIAN, (Member, IEEE), BENJAMIN DROZDENKO, (Member, IEEE), ERIC DOYLE, (Member, IEEE), RAMEEZ AHMED, (Member, IEEE), MIRIAM LEESER, (Senior Member, IEEE), AND KAUSHIK ROY CHOWDHURY, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA
Corresponding author: R. Subramanian (rsubramanian@coe.neu.edu)

This work was supported by MathWorks under the Development-Collaboration Research under Grant 1-945815398.

- Presentation, 5-6 Students work as a Group, submit your report.

- Question ?