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

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Name | Student ID | Contribution |
| 邓煜 | 11911528 | System architecture overview and related work |
| 吉辰卿 | 11911303 | Design of IEEE 802.11a system based on state machine |
| 刘远卓 | 11910921 | PHY layer design |
| 仇琨元 | 11913019 | Synthesis, MAC layer design and experiment setup, report and beamer |
| 吴沭豪 | 11911118 | Two nodes experiment |
| 赵青宇 | 11510473 | Three nodes experiment, goodput analysis and conclusions |
| I.Synthesis Software-defined radio (SDR) can still be time-consuming to design and implement, as they typically require thorough knowledge of the operating environment and a careful tuning of the program.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. II.SYSTEM ARCHITECTURE OVERVIEW The architecture of the whole project can be sectioned as the three stages below:  • 1.**Parameter Initialization**: Recommended parameters are preset in the system.  • 2.**Simulation**：Do simulation to explore the parameter with less than 5% packet loss at receiver.  • 3.**Experiment**：Apply the parameter to USRPs for over-the-air experiments.  In a given SDR pair, we identify clearly the transmitting and receiving node by using the terms designated transmitter (DTx) and designated receiver (DRx).Here the DTx and DRx are relative, they are all able to transmit and receive message.  And the USRP SDR platform is adopted as the hardware platform of this experiment, as the Figure 1 shows.    Figure 1: End-to-end TX/RX chain using USRP  **IEEE 802.11b PHY and MAC layer** packet structure specifications is adopted. All the bits in the packet are in multiples of 8 octets, which forms one USRP frame. USRP frames will compose the 802.11b packet.  **DBPSK**(differential binary phase shift keying) is adopted in the research, implying MCS=0. DBPSK is able to recover a binary sequence from the phase angles of the received signal at any phase offset, without compensating for phase. The bit error rate (BER) approaches theoretical values when residual frequency offset is much less than DBPSK symbol rate. III. RELATED WORKSDR Software Platforms Specialized software needed to work with SDR system for modulation, preamble detection, encoding, and filtering.  Example:  • GNU Radio: open source, hardware-independent, and modifiable  • Software Communications Architecture(SCA):open-source, HW-independent framework, using data flow diagram. The UI of GNU Radio, which is shown in Figure 2, adopts the design scheme of signal flow diagram similar to Simulink.  • OSSIE: using the SCA framework for interaction with the USRP board  The user interface of a SDR backend software (GNU Radio) is shown as Figure 2.    Figure 2: GNU Radio UI  • MATLAB/Simulink: Commercial high performance computing software with good user friendliness and cutting edge signal processing functionalities. With various drivers, i.e. hardware support packages/board support packages(BSPs), MATLAB is able to communicate with corresponding SDR hardware and process digital baseband from SDR. SDR On Heterogeneous Systems Some SDR projects are needed to combine with specialized hardware components for more complex designs. These specialized hardware components differing from general CPU(implemented the *Von Neumann*/*Harvard* architecture) are called as heterogenous hardware or heterogenous platform.  e.g. digital signal processors (DSP),application-specific integrated circuits (ASIC), and field-programmable gate arrays (FPGA).  Example  In fact, recent SDR platforms, either commercial products like USRP and mobile phones or open source projects like AntSDR, has already utilized heterogeneous platforms widely.  • Some SDR projects like USRP are implemented in both hardware and software on a platform that comprises both processor and FPGA.  • There are other SDR projects that are implemented using Xilinx Zynq SoC, utilizing both the PS/ARM processor and PL/FPGA fabric, like AntSDR, LimeSDR.  The USRP is a typical SDR device utilizing heterogeneous systems.  The architecture of the USRP is shown as Figure 3.    Figure 3: Architecture of USRP  From this figure we can clearly find the whole process of how SDR project works,and how the user design and control the system. IV.State-Action Based System Design For the Design of State-Action Based System in 802.11b, our approach are mainly divided into the following two ways: Firstly designing a number of state diagrams to reflect the logical and time-dependent operational steps and slot-time synchronized methods of our system.Then, we design a series of block diagrams to reflect the sequential order of operations and different state machine of transmitter and receiver in our system. Background In 802.11b, transmitter and receiver will have different working states at different time of transmitting information. Therefore, it is necessary to use some state transformation diagrams or block diagrams to describe the working state of transmitter and receiver at different time nodes, which is the main problem we solve in this section.  We structure the MATLAB code in a way that enables slot-time synchronized operations. For the implementation, we use MATLAB Coder to generate the MEX functions for the USRP objects on an Ubuntu 64-bit platform that serves as the host computer for the USRPs.  Our system design builds upon an already-defined platform, the USRP, produced by a well-known platform supplier, Ettus Research. The communication between the USRP and host computer is established in MATLAB using the Communications System Toolbox (CST) USRP Radio support package, which acts as a wrapper for the Ettus USRP Hardware Driver (UHD) drivers. Identifying the manner in which the RF samples are transported between the USRP and a calling function defines the manner in which we must build the physical (PHY) layer. Time Slot Designation The UHD transfer of a frame of samples to a transmit buffer is performed as soon as it is requested while the UHD retrieval of a frame from a receive buffer has to wait until the next rising edge of a clock cycle before trying to retrieve again.  When a frame is fetched from the receive buffer, it must wait until the next rising edge of the clock cycle before it can be fetched again. The most common undesirable behaviors that can occur are underflow and overflow. Underflow occurs when the radio requests for a frame of data from the transmit buffer, but the host is not yet ready to provide it. Overflow occurs when the receive buffer becomes full and buffered data must be overwritten.  An underflow occurs when a radio request sends a data frame to the buffer, but the host is not ready to provide it. An overflow occurs when the receive buffer is full and the buffered data must be overwritten. In this regard, we define real-time operation over the course of an entire DATA-ACK packet exchange using equation (1) below:  where is the frame time stipulated by the USRP analog-to-digital converter (ADC) and is the average time to recover any given frame, which includes the time to retrieve a frame from the receive buffer, process the retrieved frame to decode it into the corresponding bits, and other memory and conditional operations.  Essentially, we operate in real-time if we meet the timing deadline set forth by equation (1). Such an operation will guarantee a stable, basic bi-directional link that shows no sign of any undesirable system behavior, such as buffer underflow or buffer overflow. A MAC protocol that effectively schedules packet transmissions reduces the potential for packet collisions and buffer overflow, thereby decreasing packet errors.  As can be seen from the above introduction, synchronization based on some time gaps of the transmitter and receiver and the different states and strategies adopted by the transmitter and receiver at different times are very important. So I'm going to focus on these issues in a couple of sections. Slot-Time Synchronized Operations We firstly define a slot time as the smallest unit of time in which our SDR can make a decision. By design, transceive is called at a constant time interval that we define as a slot time. At each slot time, transceive sends and receives a fixed number of samples, which we refer to as a USRP frame.Then, we must know: Any IEEE 802.11-based wireless transceiver implementation must have the ability to perform operations based on some slot-based timing. For example, let the node wait to back off before sending a packet.  Next, I simply divide the time slot synchronization mode between transmitter and receiver into two types:   1. **Focus on the slot-time synchronized operations of the state of one transmission node (transmitter or receiver)：**   For the transmit state: When a node (DTx or DRx) enters the transmit state, it sends samples in the transmit buffer and ignores all samples in the receive buffer.  For the receiver state: When a node (DTx or DRx) enters the receive state, it retrieves the sample from the receive buffer for processing and puts zero into the send buffer. In this way, we ensure that the samples in the send and receive buffers are current and relevant.  In a conclusion, the step method of the transmitter object operates in a blocking way as it returns only after the radio accepts the frame to be transmitted. On the other hand, the step method of the receiver object returns right away, hence it is non-blocking.    Figure 1. Slot-Time synchronized operations, focusing on the state of one TX node   1. **Focus on the slot-time synchronized operations of the state of two transmission node (transmitter and receiver)：**   If the receiver does not receive enough data, the step call to the receiver object returns 0 as the length of the received frame. Once the radio has collected enough data, the next call returns a non-zero length value and valid data. Since we know the sampling rate of the data and the number of samples in a frame, we can calculate how long it takes to get a frame of data from the radio. The while loop blocks the transceiver function until a data frame is received. Therefore, we can use the call duration of this function as our clock source.  20220410001538  Figure 2. TX/RX function code of the Slot-Time synchronized operations, focusing on the state of two transmission nodes Transmitter State Machine For the design of state-action system, in the dual-node system of transmitter and receiver, there are altogether 4 different state machines in the transmitter at different times, as shown in the figure below:    Figure 3. The Transmitter State Machine  **State 1: Energy Detection and CSMA/CA Algorithm**  At the start, a new USRP frame arrives, and gets stored in a receive buffer. The DTx begins to continually sense energy in the channel and decides to transition either into a backoff state or to a transmit state depending on whether or not the channel is busy.  This detection sets a threshold energy . When the detected channel energy is greater than the threshold energy, the channel will immediately enter the occupied state. Whether the channel is in the state of waiting for a DIFS or in the state of random retreat, if the detected channel energy is greater than the threshold energy, the channel will enter the occupied state.  When the transmitter is in the occupied state, the transmitter will exit the state only when the detected channel energy is less than the threshold energy Eth. Then, if the transmitter is in the waiting DIFS state, it firstly waits for a DCF interframe spacing (DIFS) duration and then waits for a random amount of time that is chosen uniformly from a progressively increasing time interval. Only when the channel is free does the DTx decrement the chosen random backoff time. And only when the backoff time counts down to zero does the DTx attempt to transmit.  **State 2: Data Transmission**  When in the transmission state, The DTx prepares the packet and then puts it into the USRP's send buffer by constantly calling the transceiver function, and then transmits it over the wireless channel. After the packet is sent, two possibilities exist:  **(a)** An ACK is received and the transmission is successful.  **(b)** If the transmission fails due to packet conflict with another DTx, the transmission fails and the transmitter retransmits the packet in the next transmission.  **State 3: ACK Acquisition**  Once the packet is sent, DTx goes into the receive ACK state, searching and decoding the PLCP header in the received ACK. If the search and decoding is successful, the frame control and address fields are read from the subsequent MAC header and checked for accuracy. DTx then continues to send the next new frame and repeats the sequence of steps above until the last frame is successfully transmitted. On the other hand, if no ACK is received, the packet is considered lost, and the DTx returns the increased random return time and tries again at the next transmission.  **State 4: End of Transmission**  When there are no more packets to transmit, DTx stops the transmission process when the transmission ends. Receiver State Machine For the design of state-action system, in the dual-node system of transmitter and receiver, there are altogether 3 different state machines in the receiver at different times, as shown in the figure below:    Figure 4. The Receiver State Machine  **State 1: Data Reception**  When DRx successfully detects Preamble and start Frame separator (SFD), it firstly begins decoding the PHY and MAC headers. Then, it proceeds to extract the main data portion of the packet. When extracting the last set of data bits, Frame Check Sequence (FCS) is obtained and checked.  **State 2: SIFS Waiting**  The DRx waits for a fixed interval of time, referred to as Short Inter-frame Space (SIFS), before sending an ACK packet post reception of the data packet. I think the point of waiting for this SIFS interval is to ensure that the packet is fully received and to monitor the channel condition at this point in order to make a better decision on whether to send ACK packets in the next step.  **State 3: ACK Acknowledgment**  When the DRx successfully retrieves all valid data information and waits for a SIFS time, it sends an ACK message to the DTx.   1. **System Blocks**   In the diagrams at Part 3 and 4, we see that there are sequential operations that need to be performed in each of the substates in the diagram. To simplify the logic of what operations must be performed in each state, we have defined blocks to make up the most common operations, as shown in the diagram below. Identifying groupings of blocks with related sub-states helps to better organize and reorganize the code implemented.  20220410004622  Figure 4. The System Blocks V. PHY Layer Algorithm This chapter contain four parts, which introduce the physical layer algorithms containing signal processing and algorithms optimization. RF Layer End algorithms This part contain the automatic gain control, frequency offset estimation and compensation and cosine filtering.  First, the AGC counter attenuation by raising the envelop of the received signal to a desired level. In the article, they used function comm.AGC to implement this.  Second, in order to estimate the frequency offset between transmitter and receiver the function comm.PSKCoarseFrequencyEstimator. This is based on the equation :  where x is the signal ,which is a FFT of the signal. In order to speed up the RFFE block, they decimate the signal by a factor of 22 , because the original frequency resolution (1~10Hz)is too low to execute with a sample rate of 200KHz per sec. This decimate is actually an FFT. Besides, they designed a step as follow to get frequency estimation with reduced sample rate and 100 Hz frequency resolution.   Preamble Detection algorithms The IEEE 802.11b stander provides Start Frame Delimiter to fine tune the synchronization time. First we do a cross-correlation of of the signal filtered by a raised cosine filter to find the start point of preamble. Then, we are going to search for the SFD. If the SFD is not in the place expected, an other cross-correlation is needed on a window of descrambled frame samples to the left and right to further fifine-tune the synchronization delay. In order to speed up this step, they tested and compared 5 different kind of ways to compute the cross-correlation. As a result they chose the DSP.   Parameter Selection Some parameter are as follow.    The frame size should be carefully chosen. As a large frame size can reduce the proportion of header, but a small frame size can help making decision faster. In the end they chose 1408 as a balance. Same-Frequency Channel Operation In a multi-node setting, it is advantageous to operate the transmit and receive links, at the DTx and DRx, in the same band of frequencies. Thus, we set both DTx and DRx to operate at the same center frequency.  Unlike different-frequency channel operation, this eliminates the need for repeated switching of transmit and receive center frequencies when transitioning among the energy detection, transmit, and receive states. In addition, it makes for an easier implementation of mediumaccess and contention resolution    While using USRP, the port RF2 leaks 7 dBm into RF1. A logic is implemented to make sure it will reject its own data. VI.MAC Layer DesignMAC Layer Overview The MAC layer employs the Distributed Coordination Function (DCF) strategy incorporating the CSMA/CA mechanism as it is described in the IEEE 802.11 specification. The state machine structure of the DCF and CSMA/CA algorithm consists of 3 steps:  • Energy detection  • DIFS period  • Binary exponential backoff  These steps has already been discussed thoroughly in the previous lectures of the course. For a clear and illustrative explanation, the time sequence scheme of the CSMA/CA process is shown below in figure 13 and 14.    Figure 13. CSMA/CA in normal scenarios  When the ACK frames are interrupted by foreign transmission or spurious interference, the CSMA/CA process are also executed.    Figure 14. CSMA/CA when ACK frame is compromised Energy Detection The energy of the signal is evaluated using the easiest 2nd norm  This algorithm is convenient for fast implementations both on the general CPU with optimized math instructions and the specialized processing unit like FPGA. Time Slot Definition The standard specifies that when a packet is prepared by the DTx and ready to be sent to the intended DRx, the DTx must actively listen to the channel for a fixed specified amount of time known as the DIFS period.   1. **Short Interframe Space(SIFS)**   In 802.11 series wireless LANs SIFS is a fixed value and SIFS is the smallest inter-frame interval, so the node with SIFS has the highest priority to access the wireless link. It is equal to the time it takes for a node to switch from transmitting state to receiving state and be able to decode correctly, or the time it takes to switch from receiving state to transmitting state. The packets that may be sent after SIFS expires include ACK, CTS frames, and the SIFS values specified in different standards are different   |  |  | | --- | --- | | **Standard** | **SIFS(μs)** | | IEEE 802.11b | 10 | | IEEE 802.11a | 16 | | IEEE 802.11g | 10 |  1. **DCF Interframe Space(DIFS)**   In the DCF protocol, a node needs to monitor whether the channel is idle before starting to send data. If the channel is already idle, the node still needs to wait for the DIFS period before starting to send data; and if the channel is monitored as busy at any point during the DIFS period, the node has to postpone its data transmission.The relationship between DIFS and SIFS is calculated as formula (4).   |  |  |  | | --- | --- | --- | | **Standard** | **Slot Time(µs)** | **DIFS(µs)** | | IEEE 802.11b | 20 | 50 | | IEEE 802.11a | 9 | 34 | | IEEE 802.11g | 9 or 20 | 28 or 50 |  1. **PCF Interframe Space(PIFS)**   PCF makes the AP wait for PIFS instead of DIFS time to access the channel, and since DIFS > PIFS > SIFS, the AP always has a higher priority to access the channel than a normal node.   |  |  |  | | --- | --- | --- | | **Standard** | **Slot time(µs)** | **PIFS(µs)** | | IEEE 802.11b | 20 | 30 | | IEEE 802.11a | 9 | 25 | | IEEE 802.11g | 9 or 20 | 19 or 30 |  1. **Extended Interframe Space(EIFS)**   In case of an error in the previous frame, the sending node has to delay the EIFS instead of the DIFS time period before sending the next frame.  EIFS = Transmission time of Ack frame at lowest basic rate + SIFS + DIFS  If during this period, the DTx senses RF signal energy from other transmitting devices (i.e. when the channel is found busy), it defers the transmission and enters a Channel Occupied state.  In this state, the DTx stays idle as long as the ambient RF energy is above a specified threshold. When the energy drops below the threshold (i.e. the medium is sensed to be free), the DTx resets the DIFS duration and starts counting down again. Random Backoff and Binary Exponential Algorithm To avoid the collision of contention window, the range of the possible length of the contention windows should be as long as possible. Meanwhile, the minimization of latency requires the contention window take its minimal length as long as no collision happens. Therefore, an adaptive approach is adopted to resolve this issue. The length of the contention window are extended only when collision happens.  Since the linear progression of the contention window length is unable to reduce the possibility of collision quickly, the length are doubled once encounters collision. And when the size of contention window is large enough, the possibility of collision are reduced to a neglectable level, then the contention window can be recovered to the shortest length.  The following MATLAB code implements the binary exponential backoff algorithm.  1 function [backoff] = random\_backoff(k,t\_radio)  2 backoff = randi ([0,2^k])\* t\_radio;  3 end VII. Experimental Setups MATLAB operating on the Ubuntu 20.04 is used for the SDR backend software, and the HW support package for the USRP is the SDRu support package to cooperate with USRP N210.  The HW setup is the simplest connection between the USRP and the laptop running MATLAB using the gigabit Ethernet, showing in figure 5 below.    Figure 5. Hardware Setup MATLAB Communication Toolbox The Communication Toolbox and the SDRu support package are used for the SDR backend software. The Communication Toolbox provides the capability of processing the digital baseband at a high speed, and the SDRu support package enables the MATLAB to communicate with specific USRP devices, for example, USRP N210. MATLAB Coder The MATLAB code involved in the experiment is compatible to MATLAB coder compilation, therefore the C++ version of all the algorithms can be easily obtained. For further optimization, MATLAB coder also provides Verilog output.    Figure 6. Design Progress utilizing MATLAB Coder and Vivado HLS VIII.Experiments and ResultsThe DRx receives periodic data packets In DTx, we transmit a DATA packet of 258 USRP frame. Since the Preamble is 128 bits long, it corresponds to 2 USRP frames. Hence, we account for the reception of (258-2) = 256 USRP frames in the DATA packet. The processing time for any given frame is usually less than the required frame time tradio and is fairly constant at 2.87 ms. The first set of frames contains MAC header information that must be parsed and has a high processing time. RFFE block timing   The figure above shows RFFE under MATLAB compilation and under MEX compilation respectively. The addition of a FIR decimation step in the RFFE block reduces the sampling rate of the input for the subsequent coarse frequency offset estimation (CFOE). This reduction helps in increasing the frequency resolution. CFOE can corrects the signal so well that the later preamble detection block produces the correct synchronization delay to detect the start of DATA/ACK packet.  The results clearly show that the average execution time of RFFE blocks decreases with increasing frequency resolution. The reason for this is that CFOE uses progressively smaller FFT lengths.In addition, the average execution time using MEX is generally smaller than using interpreted MATLAB, so MEX is a better choice for RFFE block times. Two Node performance(1 DTx and 1 DRx) In order to make the experimental results statistically significant, we transmitted 100 packets in five different transmission gain environments. Using two hosts as DTx and DRx, both running MATLABR2015b in an Ubuntu operating system environment, each connected to USRPN210 via Ethernet cable. Main engines should be kept about 1 meter apart.  Packet error rate (PER) and bi-directional link latency are key performance indicators of the two node system:  (1)Packet error rate  A packet is in error if the ACK for the same is not received in time by the DTx. This could mean that either the packet could not be decoded properly by the DRx or that the ACK was corrupted or lost while in transit to the DTx. And the system should recover quickly from such errors.  (2)Bi-directional link latency  Bi-directional link delay refers to the average time it takes for a DTx to send a DATA packet and receive an ACK packet. (In the case of two nodes, there is no channel competition, only the time of transmission failure needs to be considered) Two node experiment result At the DTx, we measured the time elapsed in each state for a DATA-ACK packet exchange. The stacked plots shown below show the breakdown of the time spent in each substate.      The plot at the top shows the small contributors to the overall processing time, and the one at the bottom shows the large contributors. Both the plots are part of the same DATA-ACK packet exchange and are separated for clarity.    In the gain of 15 to 30dB, the system can guarantee a consistent ≤ 5% packet error rate.However, the presence of many metallic surfaces, such as in our lab setting, give rise to multipath reflections that can be strong and result in packet errors. The fact that the performance was significantly better when the nodes were connected by RF cables confirms the case.    In the gain of 15 to 30dB, the system have a bi-directional link latency of approximately 7 seconds. Importantly, changing the distance between two nodes in the experiment does not significantly affect performance. Even when the two nodes are separated in line of sight (for example, 15 meters), the PER and Bi-directional link delay remain consistent.  In a two-node system, increasing DIFS and fallback time has little effect on the message error rate due to lack of contention. However, increasing DIFS and fallback times also increases link latency by the same amount.  The ideal delay of the link in the experiment is (ignoring channel contention, blocking time and retransmission):  This value is in the same order of magnitude with the data obtained before, and its error is within an acceptable range. Three Nodes (2 TX and 1 RX) Experiment Previously, we have introduced the experimental situation of two nodes. In this part, we will introduce the situation of three nodes.  First of all, why three nodes instead of four, five or more. Because the author emphasizes at the beginning: more nodes would take a large amount of effort. Such an effort would not have helped us in attaining our goal of fairness assessment.  In the experiment of two nodes, we mainly verified packet error rate and bi-directional link latency. In this part, we will continue to study them. The biggest difference is the second part, that is, the competition between two different channels in the MAC layer. We should ensure the fairness of the designed protocol.    Figure22: Three nodes experiment  So how do we design the three nodes experiment? Firstly, we need to distinguish two different TX. We can know from figure 23, DRx determines the DTx address from the MAC header of the received DATA packet and sends out an ACK addressed to that DTx. Furthermore, the DRx can reject DATA packets not addressed to it. Similarly, DTX can do the same.  The second point is the setting of parameters. In the actual USRP operation, DATA/ACK packet processing in the host machine takes significantly more time compared to time taken in transmitting. And the ideal SIFS waiting time is also difficult to achieve.  So we performed our experiments with DIFS duration, minimum contention window, and ACK timeout duration set at 0.75, 0.5, and 5.0 seconds, respectively. Secondly, in the parameter setting, in order to ensure the fairness of channel competition and reduce packet retransmission, we need to reasonably select the energy detection threshold.    Figure23: MAC header of Data Frame  The following two figures are the results of the three nodes experiment. Figure 24 is similar to the two nodes experiment, They are all PER under different payload sizes. Larger payload sizes increase the likelihood of packet collisions. Therefore, the PER increases and the link latency increases, as shown in Figure 25. In addition, we can also see from Figure 25 that the three nodes experiment has basically achieved the goal of fairness. Under different payload sizes, the delays of the two links are basically the same and meet the fairness requirements.    Figure24: PER Comparison of three nodes System    Figure25: Latency Comparison of three nodes System  We also calculate the goodput in the three nodes experiment. Notice that the goodput increases with the payload size. The reason for this is that the combined PHY and MAC header occupies a decreased fraction of the entire DATA packet as the payload size increases. And the two data are basically consistent, which can also reflect the fairness of competition.    Figure 26: Average Goodput Ⅸ. Conclusion This experiment is based on USRP and MATLAB to realize the PHY and MAC layer that is IEEE 802.11b standard compliant. In general, the experiment has the following advantages.  1. IEEE 802.11b standard compliant. It helps us understand standards, and can communicate with common devices.  2. User could reconfigure the parameter values as needed.  3. The system is modular and extensible. We can download its code on the Internet.  4. It achieves a high fairness in multi node. The fair competition of channels is basically realized.  And it still has some shortcoming waiting for perfection.  1. It has trouble realizing slot-synchronized operations.  2. It is difficult to pick the right energy threshold.  3. The parameters should be readjusted in each experiment. | | | |
| Experience  1. (11911528 邓煜) In this midterm project we have deep knowledge into IEEE802.11b and SDR system implement,especially about the physical and MAC layer. Since DIFS > PIFS > SIFS , the AP has the highest priority. (PIFS=SIFS+SlotTime,DIFS = SIFS + 2\*SlotTime).In transmission , the AP has higher priority than STA.So the DIFS with 2\*slot time can make sure this.Besides, A SlotTime contains time for PHY-layer reception,time for channel sensing,time for MAC-layer processing,time for switch from Rx mode to Tx mode.DIFS with 2 SlotTime can make the DCF machanism work better and avoid channel conflict.This is the most appropriate result after balancing many factors. 2. (11911303 吉辰卿) The state machine of a transmitter consists of four parts: energy detection, data transmission, ACK packet receiving, and end of transmission. In fact, I think the most complicated part of the four parts is the energy detection part. As I said in the previous report, the energy detection part actually has three sub-states. These three sub-states are interrelated and can be converted to each other by the relationship between the current channel capability and the energy threshold Eth. There are two results of transmitting data and receiving ACK packets: transmission success and transmission failure. If the transmission succeeds, the next data packet can be transmitted. If the transmission fails, the next data packet can be retransmitted. 3. (11913019 仇琨元) There can be problem for the USRP to communicate with the commercial 802.11a AP, due to the delay of the transmission of the digital baseband between the USRP frontend and the MATLAB processing backend, and the general CPU running the MATLAB also encounters difficulties when processing the high speed symbol stream from the USRP. However, there has already been [implementation of USRP based 802.11a STA](https://www.bilibili.com/video/BV1UA411Y7b8/) using similar techniques. Such implementation are enabled by the extremely optimization of the MATLAB processing program. 4. (11910921 刘远卓) After the leading code is complete, PLCP headers contain the physical parameters related to the data transfer. These parameters include SIGNAL, SERVICE, LENGTH of data to be transmitted, and 16-bit CRC. According to these parameters, the receiver will adjust the receiving rate, choose the decoding method, and decide when to end the data reception. The SIGNAL field is 8 bits long and defines the data transmission rate. It has four values: 0Ah, 14h, 37h and 6Eh, specifying transmission rates of 1Mbps, 2Mbps, 5.5Mbps and 11Mbps respectively, and the receiver will adjust its reception rate accordingly. The SERVICE field is also 8 bits long and specifies which modulation code (CCK or PBCC) to use. The LENGTH field is 16 bits long and indicates how long (in microseconds) it will take to send the subsequent PSDU. The 16-bit CRC check code is used to check whether the received signaling, service, and length fields are correct. 5. (11911118 吴沭豪)In Slot time, the CCA time of channel detection and the antenna transmit/receive switch are composed. CCA monitors the channel. If the data packet is not transmitted to itself, CCA monitors busy and waits for a Slot to continue monitoring. If the listening packet is for itself, switch to receive state. DIFS uses two channel listening processes, that is, two Slot times, which do not trigger backoff. DIFS is followed by backoff only when two consecutive channels are monitored to be idle. In conclusion, SIFS is the smallest inter-frame spacing, so nodes using SIFS have the highest priority for accessing the wireless link. 6. (11510473赵青宇) In this experiment, although we did not study the implementation of 802.11b code in detail. By reading the paper, we have a deep understanding of the framework of the whole system. And understood the flow of programming and the transformation of state machine between transmitter and receiver. We are more familiar with the functions of AGC, PSKCoarseFrequencyEstimator, CarrierSynchronizer, SymbolSynchronizer in the signal synchronization part of MATLAB. For questions in class: 1 After the implementation of this program, can we use USRP to communicate with the router? I think we can do it. They support 802.11b protocol. However, it is necessary to interpolate and extract the signal sampled in USRP. Because in the 5M bandwidth LTE system, the sampling rate is 7.68MHz, while USRP only supports the sampling rate of integral multiple frequency division of 100M without changing the FPGA code. And the sampling rate closest to 7.68M is 6.25M. Therefore, interpolation and extraction are needed to realize rate conversion. 2. What is PLCP? In the composition of WiFi packets, there are packets in the format of PPDU in the PHY layer. The full name of PLCP is physical layer convergence protocol. It refers to the PPDU protocol packet. 3. What is the state machine of RTX in the paper? It includes four states. 1.Energy detection, 2. Data transmission, 3.ACK detection, 4.Data retransmission. | | | |
| Score |  | | |