CS578: Internet of Things

Dept. of CSE, IIT Guwahati

Mid-Sem Exam **Date:** 19-09-2022 **Marks:** 50 **Total Time:** 2 hrs

==========

1. Write your answer in one-two lines for each of the following questions. 5x2 = 10
   a) What does it mean by sectorization in 802.11ah?
   b) Which address type is globally unique for an IoT device?
   c) What are the use of message authentication code and message encryption in IoT?
   d) What is the difference between ABP and OTAA registration methods w.r.t. security keys?
   e) What is API?

   Ans:

   a) Sectorization partition the coverage area of a Basic Service Set (BSS) into sectors, each containing a subset of stations. It is implemented by beamforming technology. Basically it replaces the omni-directional antenna at each AP by multiple sector antennas of x degrees opening where x < 360 degree.
   b) DevEUI or EUI-64
   c) Message authentication code is used to achieve message integrity and message encryption is used to achieve confidentiality.
   d) In ABP, NwkSKey and AppSKey keys are preconfigured and stored in the end device as well as in the server.
   But, in OTAA, these keys are generated by the AppKey which is known to the device only, and during dynamic registration/join process this key is informed to the server.

   e) An API is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange. For example, web API. These APIs mainly deliver requests from web applications and responses from servers using HTTP.

2. Let us consider, in UART, synchronisation between transmitter and receiver is done in the form of a high to low transition on the data line when transmission starts. Consider all other requirements (frame format, baud rate, etc.) are pre-configured.
   a) Do you think that the above approach has some issues? If yes, what are they? 2+2
   b) Can you propose some solution for those issues? 2
   c) Give your justification or explain your solution using diagram. 2+2

   Ans:

   (a) there are two problems with this scheme:
   - The single high to low transition is not enough to synchronise timing over a long period of time.
     - So, they need periodic re-synchronisation.
   - If the first bit of data it wants to send is also represented by a high level, then there is no transition, and vice-versa.
     - So, they need some way to distinguish the first data bit.

   (b) Solution: Both of these problems are solved in UART by using **start** and **stop bits**.

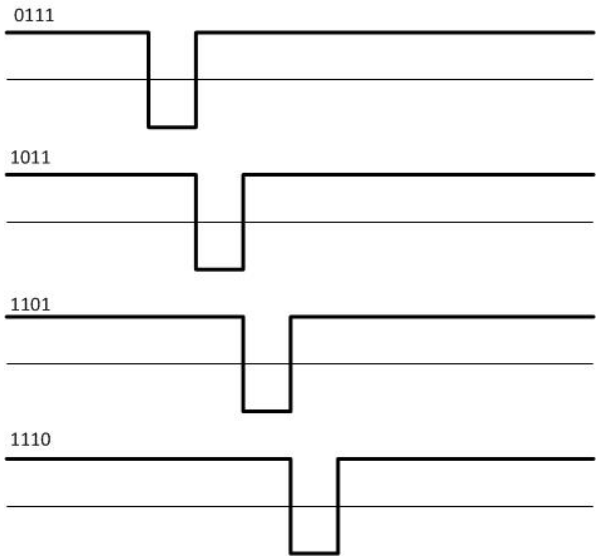
   Start Bit:
   - The UART transmission line is normally held at a high voltage level when it's not transmitting data i.e. idle
   - It start bit is set to logic low as it is used to signal the receiver that a new framing is coming.

Stop bit:

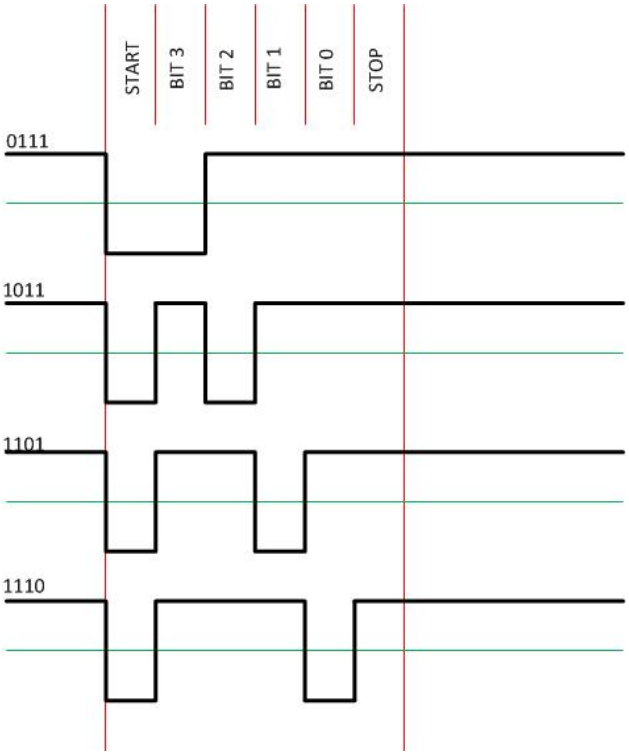
- To signal the end of the data packet, the sending UART drives the data transmission line from a low voltage to a high voltage

(c)

Use of Start bit:

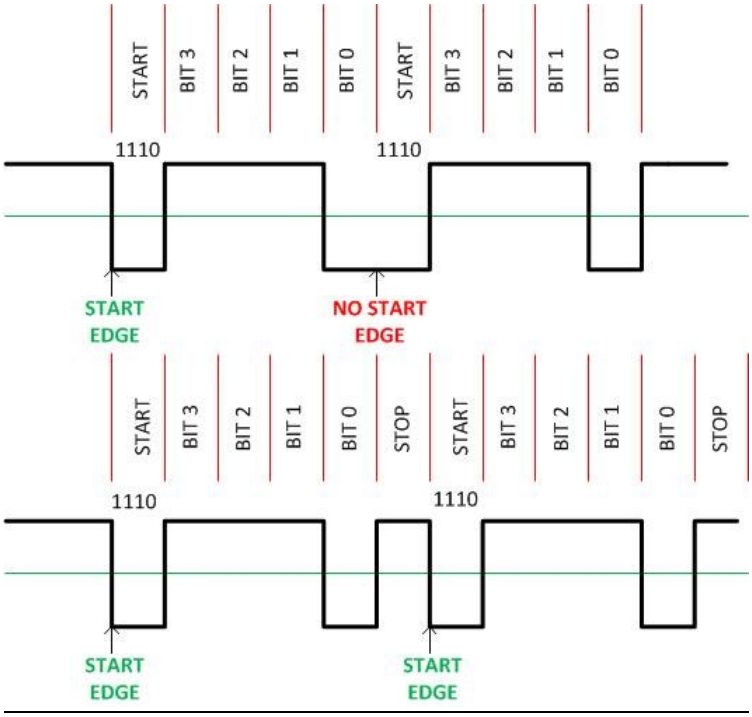


Without any time reference, they all look alike as a voltage signal.



As time matters with asynchronous signals, it is needed to send a "LISTEN UP" signal.

Use of Stop bits



- In the first figure, no transition for start bit !
- So, the stop bit ensures the line goes back to the idle state at the end of the transmission.
- It also gives the receiver a time window to handle the bytes just received, reset, and get ready to listen for the next start bit.

3. Let us consider that it is required to establish a serial communication between two devices -- Arduino UNO to NodeMCU. The devices should be connected as follows: Arduino Pin 10 and 11 are connected to NodeMCU Pin 5 and 6, respectively. Arduino Pin 11 is the digital Tx pin. Write a program to transfer the digits of your roll number from Arduino UNO to NodeMCU. Input in the Arduino program will be your roll number. The roll number is an integer and is provided by assigning to a variable.
   a) The Arduino program should extract all digits from the roll number and transmit those digits sequentially. 2+3
   b) Then, NodeMCU will receive them serially, and do the sum of all digits and finally display the computed sum. 3+2

Ans:

//This is for Arduino Code

```
#include<SoftwareSerial.h>
//Connect Arduino Digital pin 10 (Rx) --> Node MCU pin D5 (Tx)
//Connect Arduino Digital pin 11 (Tx) --> Node MCU pin D6 (Rx)
SoftwareSerial s(10,11); //(Rx,Tx)

int animationSpeed = 1000;
long RollNo = 190602045;
long varRollNo;
int RollNoDigits[10];
int noOfDigitInRollNo;
int digit;
int syncValue = 255;

void setup() {
  // put your setup code here, to run once:
```

```
  Serial.begin(9600); //initialize serial communication
  Serial.println("On the serial monitor in Arduino.");
  Serial.print("Roll Number: ");
  Serial.println(RollNo,DEC);

  s.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  varRollNo = RollNo;
  noOfDigitInRollNo = 0;

  //Extracts the digits from roll no. & count the no. of digits present in the roll no.
  while(varRollNo > 0)
  {
    digit = varRollNo % 10;
    varRollNo = varRollNo / 10;
    RollNoDigits [noOfDigitInRollNo] = digit;
    noOfDigitInRollNo ++;
  }
  //reverse the digit array
  int j = noOfDigitInRollNo-1;
  int i=0;
  while (i < j)
  {
    int temp = RollNoDigits [j];
    RollNoDigits [j] = RollNoDigits [i];
    RollNoDigits [i] = temp;
    i++;j--;
  }
  //if (s.available()>0)
  //{
  // To sync with node MCU, first transmit one big integer which is not the part of program
    s.write(syncValue);
    delay(animationSpeed);

    Serial.print("No. of Digits in Roll No: ");
    Serial.println(noOfDigitInRollNo,DEC);
    s.write(noOfDigitInRollNo);
    delay(animationSpeed);

    Serial.println("Send All Digits: ");
    for(i=0; i<noOfDigitInRollNo; i++)
    {
      Serial.print('\t');
      Serial.print(RollNoDigits[i],DEC);
      s.write(RollNoDigits[i]);
      delay(animationSpeed);
    }
  //}   // end of if s available
  Serial.println("");
}
```

//This is for NodeMCU Code

```
#include <SoftwareSerial.h>
//Connect Arduino Digital pin 10 (Rx) --> Node MCU pin D5 (Tx)
//Connect Arduino Digital pin 11 (Tx) --> Node MCU pin D6 (Rx)
SoftwareSerial s(D6,D5); //(Rx,Tx)

int digit;
int animationSpeed = 1000;
int noOfDigitInRollNo;
int sumDigits;
int syncValue;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  Serial.println("On the serial monitor in NodeMCU.");

  s.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  if (s.available()>0)
  {
    syncValue = s.read();
    Serial.print("Received syncValue:");
    Serial.println(syncValue);
    delay(animationSpeed);

    if (syncValue == 255)
    {
      noOfDigitInRollNo = s.read();
      Serial.print("Received noOfDigitInRollNo:");
      Serial.println(noOfDigitInRollNo, DEC);
      delay(animationSpeed);

      sumDigits = 0;
      while(noOfDigitInRollNo>0){
         digit = s.read();
         Serial.print("Received digit in NodeMCU:");
         Serial.println(digit, DEC);
         sumDigits = sumDigits + digit;
         noOfDigitInRollNo --;
         delay(animationSpeed);
      }
      Serial.print("Total Sum of the digits:");
      Serial.println(sumDigits, DEC);

    } //end of if syncValue
  } // end of if s available
}
```

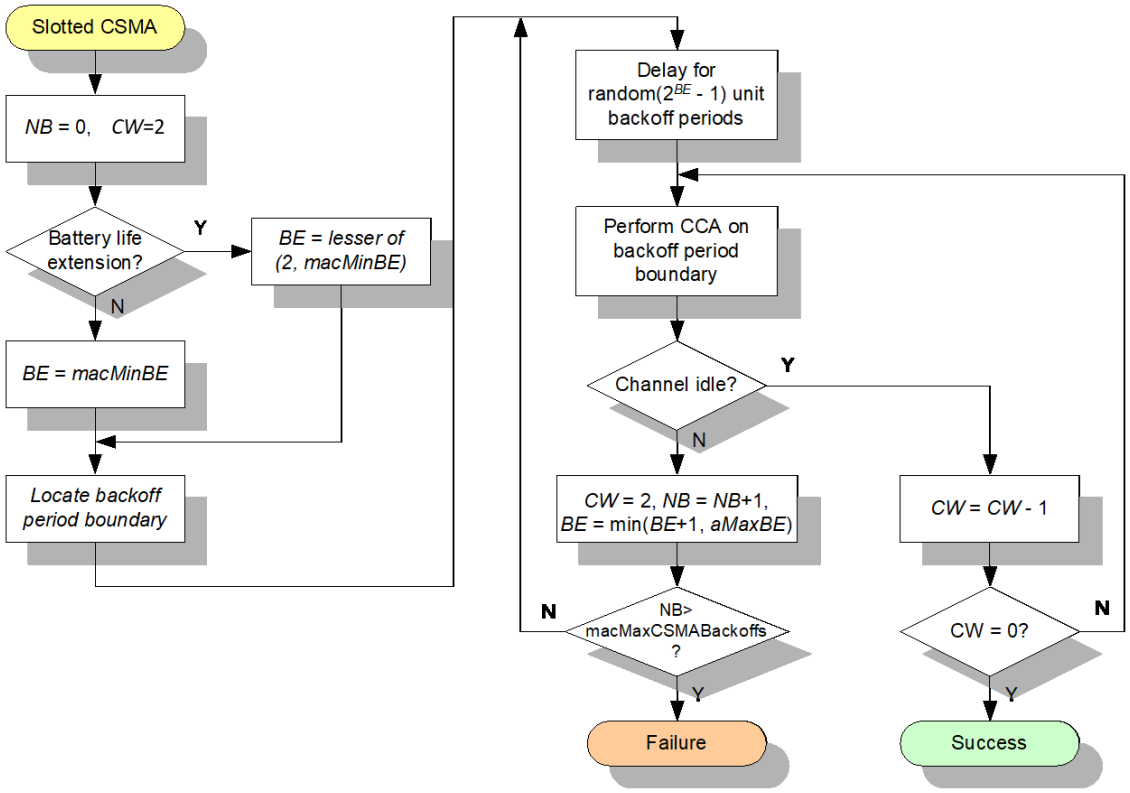
4. The CSMA-CA algorithm shall be used before the transmission of data or MAC command frames transmitted within the CAP of the superframe, unless the frame can be quickly transmitted following the ACK of a data request command.
   a) Draw the Flowchart of slotted CSMA-CA algorithm. 4
   b) Each device shall maintain three variables for each transmission attempt using CSMA-CA: NB, CW, and BE. Explain the use of these three variables in CSMA-CA. 2+2+2

Ans:

(a) (Give partial marks 2 if any student has mentioned the main steps or tasks, but did not write them in correct sequence and correct format. Those main tasks are: i) Locate Backoff Period Boundary, ii) Delay for randomly selected backoff period, iii) Perform CCA, iv) Check channel is idle or not, iv) check the limit of CW, v) check the maximum number of backoff cycles.

Give full marks 4, if the below diagram is drawn fully.

Otherwise give 0 marks.)



(b)

NB (Number of Backoff): *NB* is the number of times the CSMA-CA algorithm was required to back off while attempting the current transmission; this value shall be initialized to zero before each new transmission attempt. If NB >macMaxCSMABackoffs, the packet is dropped due to access failure.

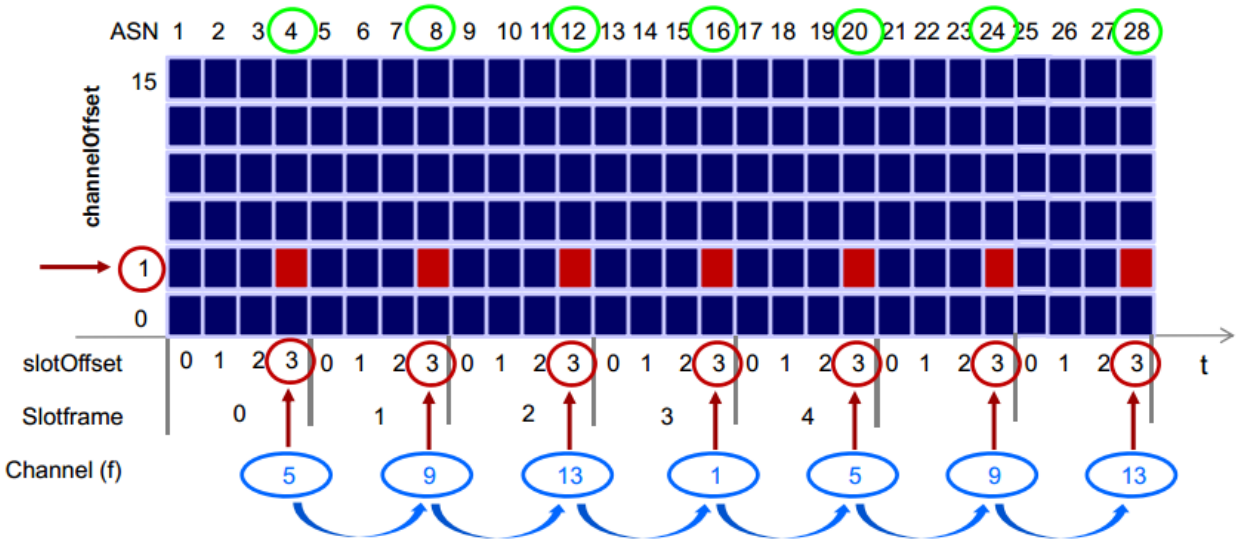
CW (Contention Window): *CW* is the contention window length, defining the number of backoff periods that need to be clear of channel activity before the transmission can commence. This value shall be initialized to CW_0 before each transmission attempt and reset to CW_0 each time the channel is assessed to be busy. By default, CW_0 shall be set to two. But, in unslotted CSMA-CA, CW is not required as it performs CCA only one before transmission.

BE (Backoff Exponent): *BE* is the backoff exponent, which is related to how many backoff periods a device shall wait before attempting to assess a channel. *BE* shall be initialized to the value of *macMinBE.* But, in slotted systems with the received BLE field set to one, this value shall be initialized to the lesser of two and the value of *macMinBE*. In slotted CSMA-CA, the MAC sublayer shall ensure that, after the random backoff, the remaining CSMA-CA operations can be undertaken and the entire transaction can be transmitted before the end of the CAP. If the number of backoff periods is greater than the remaining number of backoff periods in the CAP, the MAC sublayer shall pause the backoff countdown at the end of the CAP and resume it at the start of the CAP in the next superframe. If the number of backoff periods is less than or equal to the remaining number of backoff periods in the CAP, the MAC sublayer shall apply its backoff delay and then evaluate whether it can proceed.

5. Write your answer briefly.
   a) In a TSCH network, one cell say (i,j) is assigned to a link of the network for data transmission. It is claimed that the particular cell (i,j) will use all available channels if multiple packets are needed to be transmitted over time following the TSCH scheduling mechanism. Do you accept the claim? Give justification against your answer yes or no. Draw the diagram to explain one example corresponding to your answer. 2+3
   b) LoRaWAN endpoints are classified into three classes. What are those classes and how they are different from each other? Explain using diagram. 2+3

Ans:

(a)



$$f = F\{(ASN + chOf) \bmod n_{ch}\}; \qquad ASN = k \,.\, S + t$$

ASN (absolute slot number) : total # of slots elapsed since the network was deployed

n_{ch}: number of physical channels presently available to consider

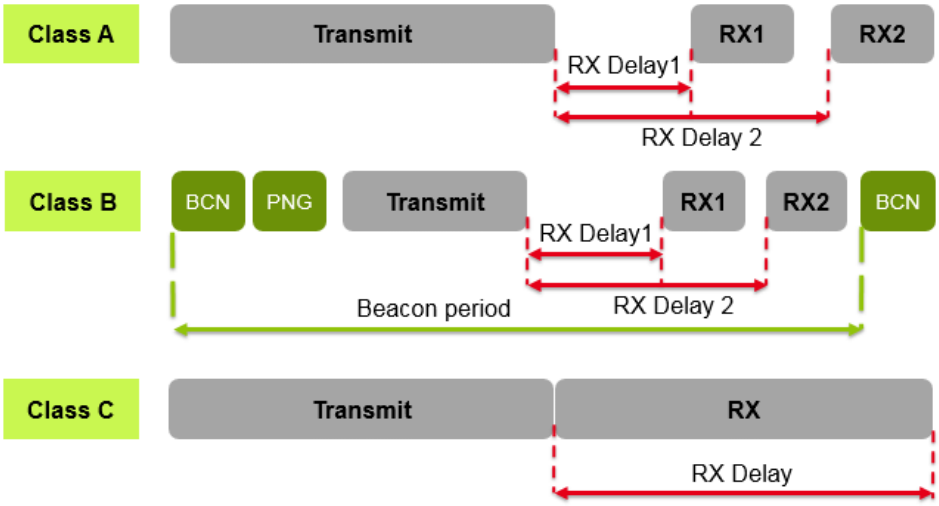
F is implemented as a look-up-table containing the set of available channels

k : count of slotframe cycle since the start of the network

S : slotframe size

t : timeslot in a slotframe

(b)



Class A:
this is default implementation
optimized for battery-powered nodes
allows bidirectional communications
two receive windows are available after each transmission

Class B:
Class B node should get additional receive windows compared to Class A
gateways must be synchronized through a beaconing process
“ping slots”, can be used by the network infrastructure to initiate a downlink communication

Class C:
This class is particularly adapted for powered nodes
enables a node to be continuously listening by keeping its receive window open when not transmitting