USN | | | | | | | | | | |

Internal Assessment Test 3 – Nov,2018

| Sub: | IOT &WSN | | | | | Sub Code: | 15EC752 | Branch: | ECE/TCE |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 22-11-18 | Duration: | 90 min's | Max Marks: | 50 | Sem / Sec: | 7 A,B,C,D | | OBE |

Answer all FIVE Questions

| | | | | |
|---|---|---|---|---|
| 1 (a) | Explain LEACH protocol with necessary figures | [10] | CO2 | L2 |
| | **OR** | | | |
| (b) | Explain routing protocols with proper classification. | [10] | CO4 | L2 |
| 2 (a) | Explain about CSMA protocol with proper flow diagram. | [10] | CO4 | L2 |
| | **OR** | | | |
| (b) | Explain S-MAC protocol with necessary figures. | [10] | CO4 | L2 |
| 3 (a) | What is geographical routing? Explain Greedy Perimeter Stateless Routing for Wireless Networks with example. | [10] | CO4 | L2 |
| | **OR** | | | |
| (b) | Explain Mediation Device Protocol with advantages and disadvantages. | [10] | CO4 | L2 |

USN | | | | | | | | | | |

Internal Assessment Test 3 – Nov,2018

| Sub: | IOT &WSN | | | | | Sub Code: | 15EC752 | Branch: | ECE/TCE |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 22-11-18 | Duration: | 90 min's | Max Marks: | 50 | Sem / Sec: | 7 A,B,C,D | | OBE |

Answer all FIVE Questions

| | | | | |
|---|---|---|---|---|
| 1 (a) | Explain LEACH protocol with necessary figures. | [10] | CO5 | L2 |
| | **OR** | | | |
| (b) | Explain routing protocols with proper classification. | [10] | CO5 | L2 |
| 2 (a) | Explain about CSMA protocol with proper flow diagram. | [10] | CO5 | L2 |
| | **OR** | | | |
| (b) | Explain S-MAC protocol with necessary figures. | [10] | CO5 | L2 |
| 3 (a) | What is geographical routing? Explain Greedy Perimeter Stateless Routing for Wireless Networks with example. | [10] | CO5 | L2 |
| | **OR** | | | |
| (b) | Explain Mediation Device Protocol with advantages and disadvantages. | [10] | CO5 | L2 |

| | | | | |
|---|---|---|---|---|
| | | | | |
| 4 (a) | Explain about vulnerabilities in IOT/M2M. | [10] | CO3 | L2 |
| | **OR** | | | |
| (b) | Write a short note on 1) PAMAS protocol 2) TRAMA protocol. | [10] | CO5 | L2 |
| 5 (a) | Explain about Arduino platform and write a program for displaying traffic lights. | [10] | CO3 | L2 |
| | **OR** | | | |
| (c) | Explain about the security and threat analysis in IOT/M2M using neat figure. | [10] | CO3 | L2 |

| | | | | |
|---|---|---|---|---|
| 4 (a) | Explain about vulnerabilities in IOT/M2M. | [10] | CO4 | L2 |
| | **OR** | | | |
| (b) | Write a short note on 1) PAMAS protocol 2) TRAMA protocol. | [10] | CO4 | L2 |
| 5 (a) | Explain about Arduino platform and write a program for displaying traffic light. | [10] | CO4 | L2 |
| | **OR** | | | |
| (c) | Explain about the security and threat analysis in IOT/M2M using neat figure. | [10] | CO4 | L2 |

Q1(a)

3) LEACH:

Low Energy Adaptive Clustering Hierarchy:

→ Assumes dense sensor n/w of homogeneous, energy contrainted nodes, which shall report their data to sink node. TDMA based MAC protocol, integrated with clustering & simple "routing" protocol.

→ Partitions the nodes into clusters and in each cluster, a dedicated node, clusterhead, is responsible for creating & maintaining the TDMA schedule.

→ Other nodes are member nodes. To all member nodes, don't have their time slot, they can sleep.

→ The CH aggregates the data of its members & transmits it to sink node / other node for further relaying. As the sink is often faraway, the CH must spend significant energy for transmission.

→ For member, it is cheaper to reach CH than to transmit directly.

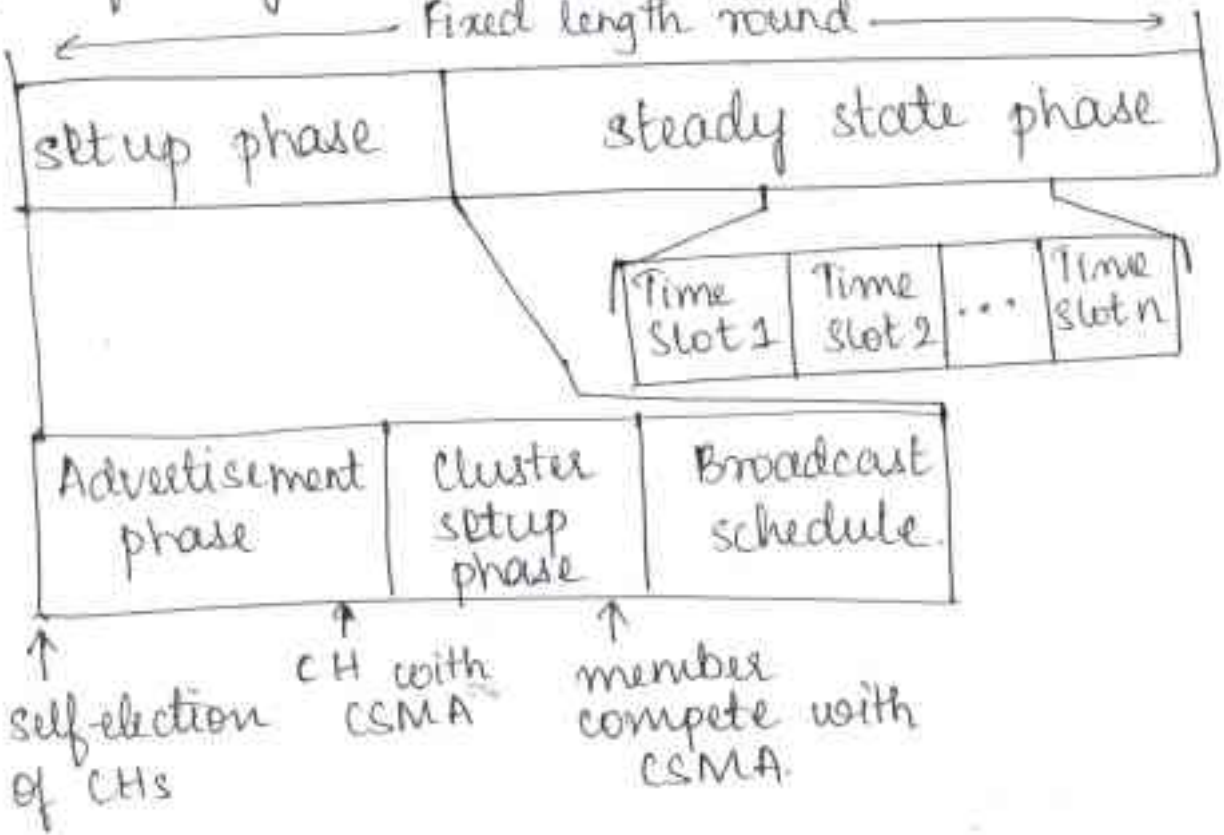
→ CH role - is energy consumed as it always switched on and is responsible for the long range transmissic
- is given on rotational basis to all the nodes.

→ Nodes decide independently whether to become a CH, hence signalling traffic required for CH election.

→ Signalling traffic is required to associate the nodes to CH. Decision depends on when it was CH previously, if its been long it is more likely to become CH.

→ Protocol is round based, all nodes has to become CH

→ Subsequently associate themselves to CH



Q2(a).

1) CSMA protocol with necessary flow diagrams.

→ Carrier sense multiple access:

In this technique, station senses the medium before trying to use it.

CSMA requires that each station first listens to the medium before sending the packet

CSMA is based on 2 principles

- sense before transmit
- listen before talk

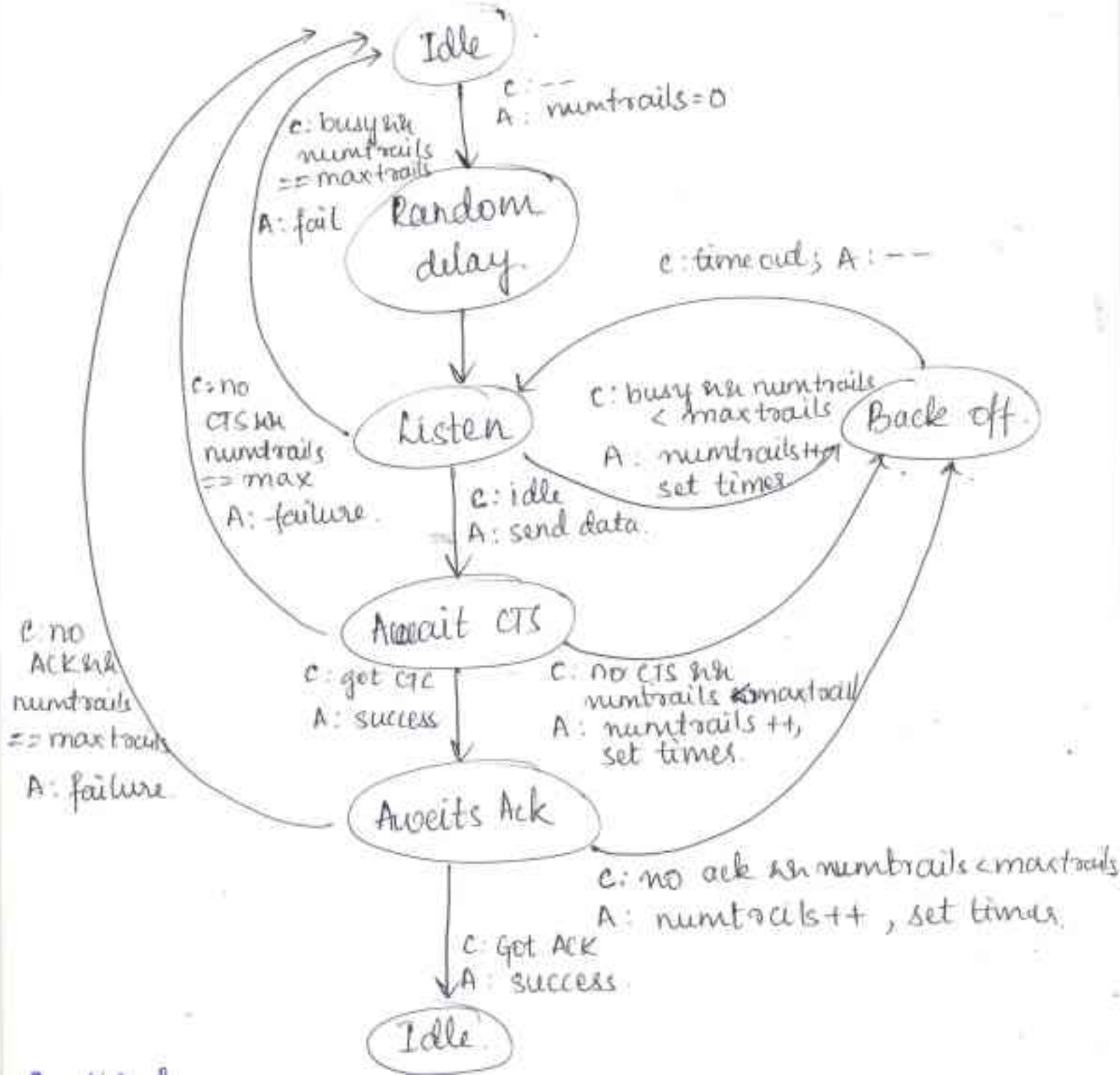
CSMA is used in detection of congestion and in order to avoid collision.

Below is the schematic for CSMA protocol,

where C: condition and

A: action.

P-T-O


Idle
C: --
A: numtrails=0
C: busy && numtrails == maxtrails
A: fail
Random delay
C: timeout; A: --
C: no CTS && numtrails == max
A: failure
Listen
C: busy && numtrails < maxtrails
A: numtrails++, set timer
Back off
C: idle
A: send data
Await CTS
C: no ACK && numtrails == maxtrails
A: failure
C: got CTS
A: success
C: no CTS && numtrails < maxtrails
A: numtrails ++, set timer
Awaits Ack
C: no ack && numtrails < maxtrails
A: numtrails++, set timer
C: Got ACK
A: success
Idle

Q2(b).

4) SMACS:

→ Self Organising Medium Access Control for sensor n/w. SMACS is scheduled based medium Access control protocol for the wireless sensor n/w.

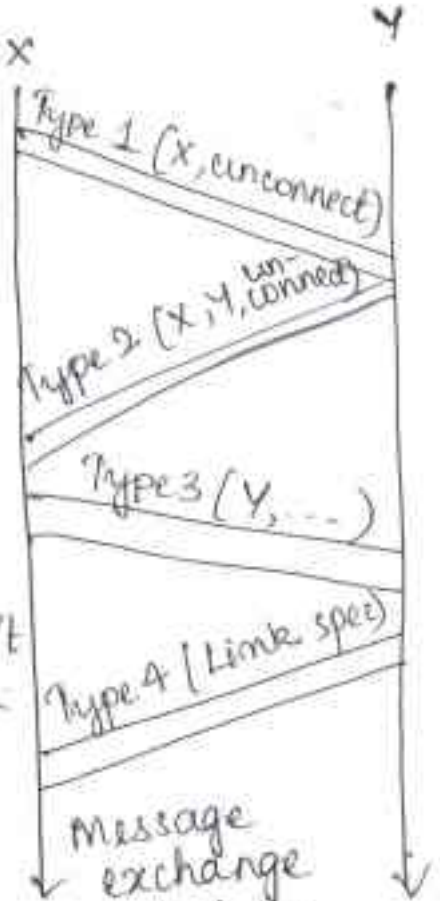
→ This MAC protocol uses a combination of TDMA & FDMA or CDMA for accessing the channel

→ In this protocol, the time slots are wasted if the sensor node doesn't have data to be sent to the intended receiver. Detects neighbouring nodes & to setup exclusive links or channels for these

→ Assignments of links will be such that no collision at receivers. To acheive SMACS to be care of that for a single node the time slots different links do not overlap. It is not required that the node & its neighbour transmit at entirely different times.

→ In such case, it is required that they must transmit to different receivers & have to use different frequency /codes.

CASE 1:
Nodes X, Y are not connected so far.
Node X sends invitation.
Node Y answers, telling X that is unconnected to any other node.
Node X tells Y to pick slot/freq. for the link.
Node Y sends back the link spec.




CASE 2:
Node X has some neighbours, Y doesn't
Node X will construct link specificatn and instruct Y to use it.

CASE 3:
Node X has no neighbours, Y has some. Y picks up link specificatn.

CASE 4: both nodes already have links. Nodes exchange their scheduling pick free slot/freq. in mutual agreement.

(§ 3(a).

*) Geographic routing:

→ Routing tables contain information to which next hop a packet should be forwaed.

→ Explicitly constructed.

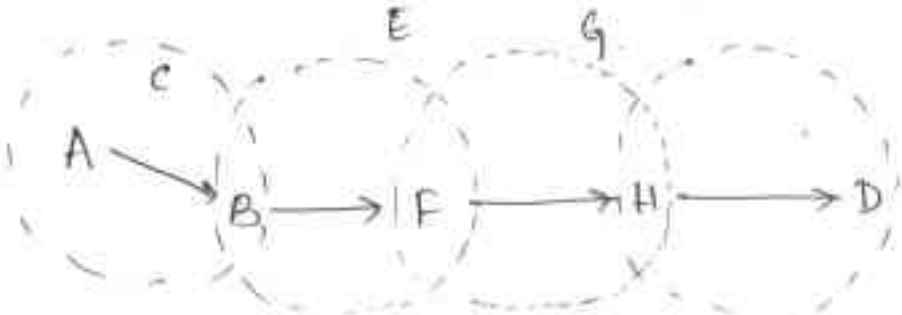
→ Implicitly infer this information from physical placement of nodes.

→ Defines that the position of current node, current neighbours, destination known send to a neighbours in right direction as next hop.
→ Send to any node in given area - geocasting.
→ Use position information to add in routing - position based routing. Might need a location service to map node ID to node.

8) Greedy perimeter stateless routing:
→ It uses 2 forwarding methods - greedy forwarding
- perimeter "

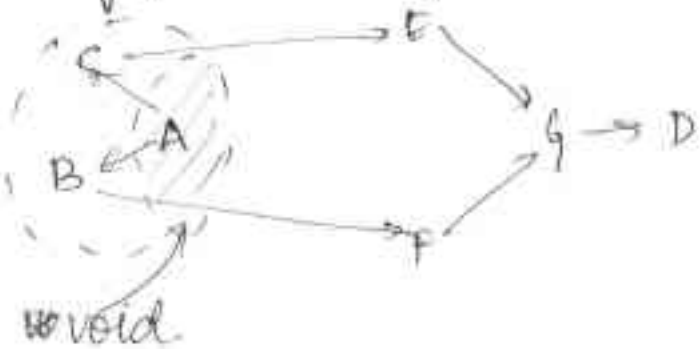
* Greedy forwarding:
→ Every node broadcast its IP address & position periodically. Every node stores position of its 1 hop neighbour.

Ex:




Node A will check the table & find out node B is more near to destination as compared to other nodes.

Greedy forwarding fails.




Node A doesn't have any neighbour nearer to D

Q3(b)

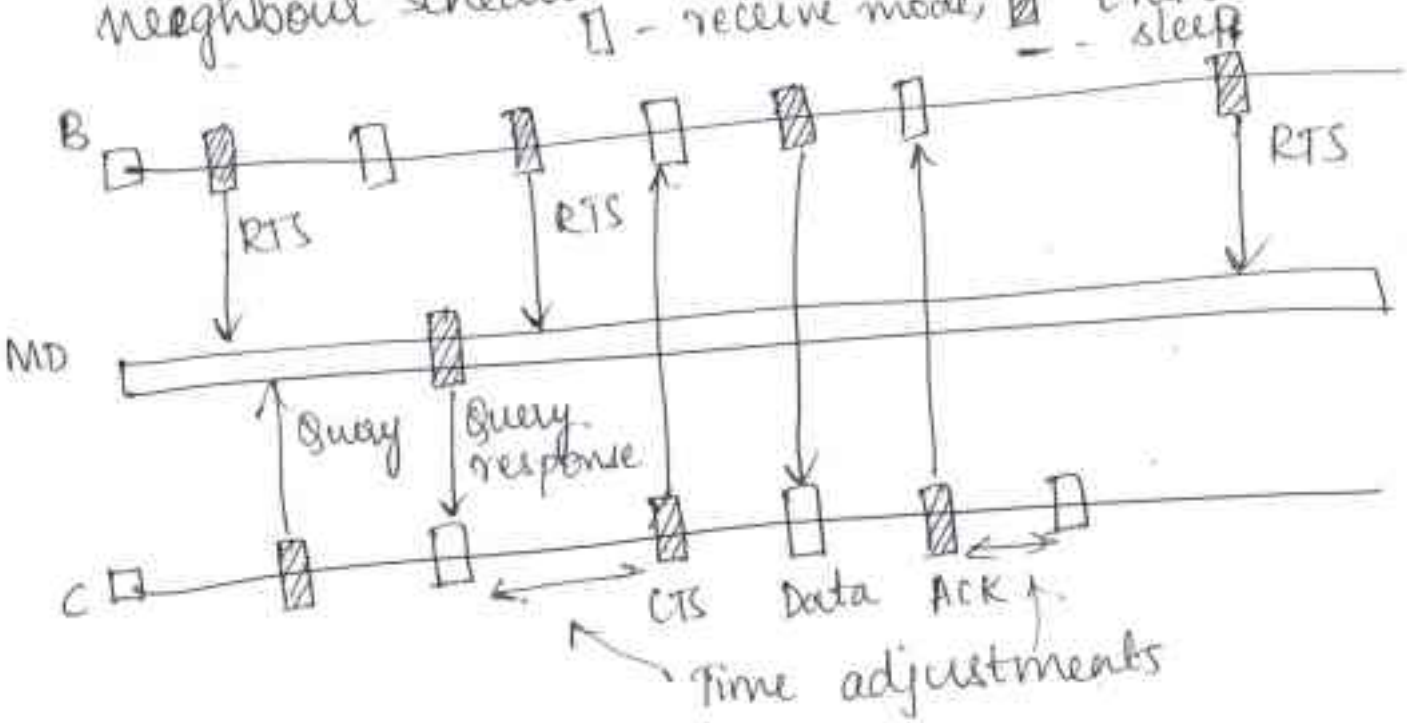
10) Mediation Device Protocol.

→ Avoid useless listening on the channel for messages.

→ Mediation device is used which is available all the time. Allows each node to sleep periodically & to wake up only for short time to receive packets.

→ No global time reference, node does not take care of neighbour schedule. Case MD is unconditional

[] - receive mode; ▨ - transmit mode — - sleep



- B sends RTS to MD
- MD stores the info
- Receiver C sends query to MD
- MD tells 'C' when to wake up.
- 'C' sends CTS to 'B'
- 'B' sends Data to 'C'
- 'C' sends acknowledgement to 'B'
- 'C' returns to old timing.

Advantages:

→ Doesn't require time synchronizatn b/w nodes only MD has to learn time periods of nodes.
→ Most of power burdon is shifted to MD, other devices can be in sleep mode most of the time.
→ Synchronisatn work done by MD, very loco duty cycles can be supported.

Disadvantages:

→ The different nodes might collide repeatedly when node have overlapping wake up periods.
→ MD has to be energy independent & unconstrained

2) (S4(a)) Vulnerabilities in WSNs.

→ Insecure Web Interface
→ Insufficient Authentication / Authorisation
→ Insecure network services
→ Lack of Transport Encryption
→ Privacy concerns
→ Insecure Cloud Interface
→ Insecure Mobile Interface
→ Insufficient Security Configurability
→ Insecure Software / Firmware
→ Poor Physical Security.

3) (S2(b)) IOT security Tomography & layered attack model

→ Security Tomography means finding attack vulnerable sections / subsections.
→ Observations for behaviour using a finite no. of objects or threats in a complex set of subsystems

The layered attack model gives possible attacks on the layers


→ Attack Solutns:

- Layer 1:
  Depends on the devices used, ex, link level provisioning of security.
- Layer 2:
  Programming the network switches to prevent internal node attacks during use of DHCP or spanning tree protocol.
  Additional controls: ARP inspectn, disabling unused ports. Provisions for MAS for security
- Layer 3:
  Use of temper resistant router, use of packet filtering, a firewall for controlling routing messages b/w layer 3 & 4 for reducing risks.

Layer 4:
Port scanning method to identify the vulnerable port. Effective firewall configuring & opening of n/w ports & locking down ports only to those required.
DTLS b/w layer 5 & 4

Layer 5 & 6:
Results of poor coding practices of Applicat$^{n}$ programme
Use HTTPS communication link for Web apps/services

4) Explain Arduino. WAP to display traffic lights.

→ Arduino is a microcontroller board, containing on-board power supply, USB port to communicate with PC, & an Antel micro controller chip.
The board can be programed & connected to systems like sensors & actuators. It is an open source hardware, anyone can get the details of the design & modify it.

CODE: assume LEDs are green for N-S direction & red for E-W direction.

```
int internalLED=13;
int ledR0, ledY0, ledG0, ledR1, ledY1, ledG1, ledR2,
 ledY2, ledG2, ledR3, ledY3, ledG3;

ledR0=2; ledY0=3; ledG0=4
                  ledY3=12, ledG3=14;
```

thereby causing it to crash.

5.a)

- The Arduino platform is used for programming embedded devices.
- It is programmed using avg-gcc toolkit.
- It consists of a preinstalled bootloader which is embedded in its firmware.
- The codes are developed in a graphical cross site platform.
- The board is connected to a computer which runs the IDE.
- The bootloader enhances the loading, which loads the required OS functions and software into the system hardware and networking capabilities into the Arduino board.

The Arduino IDE consists of a set of modules that are used for hardware and software environment development and prototyping of a software for a specific device platform.

for this, an appropriate version of the IDE is downloaded from the Arduino website according to the OS of a computer.

The IDE is then used to push the developed code into the Arduino board.

This IDE consists of C/C++ libraries.

The ARduino IDE is similar to a file editor for code in a processing environment and with libraries.

It indents, highlights the syntax of the code and matches braces.

The simplicity of the Arduino platform can be seen from the necessity of only two functions for an executable code, namely setup() and loop().

The setup() function is used for initialization setting that runs only once.

The loop() function is used for the main execution of the code which runs until the power is turned off.

- The serial monitor is used to display the message from the embedded software of the microcontroller onto the computer screen.

The program for displaying traffic lights is as shown :-

```
int R0, Y0, G0, R1, Y1, G1, R2, Y2, G2, R3, Y3, G2;
R0 = 2; Y0 = 3; G0 = 4; R1 = 5; Y1 = 6; G1 = 7; R2 = 8;
Y2 = 9; G2 = 10; R3 = 11; Y3 = 12; G3 = 14;

void north_south_green ()
{
  digitalWrite (R0, LOW); digitalWrite (Y0, LOW);
  digitalWrite (G0, HIGH);
  digitalWrite (R2, LOW); digitalWrite (Y2, LOW);
  digitalWrite (G2, HIGH);
}

void east_west_red ()
{
  digitalWrite (R1, HIGH); digitalWrite (Y1, LOW); &
  digitalWrite (G1, LOW);
```

```
    digitalWrite (R3, HIGH); digitalWrite (Y3, LOW);
    digitalWrite (G3, LOW);
void  north_south_yellow();
{
    digitalWrite (R0, LOW); digitalWrite (Y0, HIGH);
    digitalWrite (G0, LOW);
    digitalWrite (R2, LOW); digitalWrite (Y2, HIGH);
    digitalWrite (G2, LOW);
}
void  east_west_yellow ()
{
    digitalWrite (R1, LOW); digitalWrite (Y1, HIGH);
    digitalWrite (G1, LOW);
    digital Write (R3, LOW); digitalWrite (Y3, HIGH);
    digitalWrite ( G3, LOW);
}
void . north_south_red (){
{
    digitalWrite (R0, HIGH); digitalWrite (Y0, LOW);
    digitalwrite (G0, LOW);
    digitalWrite (R2, HIGH); digitalWrite (Y2, LOW);
    digitalWrite (G2, LOW);
}
void east_west_green ()
{;     // G1 and G3 will be HIGH
}
```

```
Void setup()
{
  pinMode (R0, OUTPUT);
  pinMode (Y0, OUTPUT);
  pinMode (G0, OUTPUT);
  :
  pinMode (G3, OUTPUT);
  Serial.begin (9600);
  Serial.println ("Traffic Light");
}
void loop()
{
  north_south_green();
  east_west_red();
  delay (30000);
  north_south_yellow();
  delay (10000);
  north_south_red();
  east_west_green();
  delay (30000);
  east_west_yellow();
  delay (10000);
}
```

then 0, 1, 2, 3 corresponds to North, East, South, West directions.

When North - South lights are green, East-West direction lights are red and vice-versa.