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

Internal Assessment Test 2 – Sept. 2018

Sub:	IOT &WSN					Sub Code:	15EC752	Branch:	TCE
Date:	17-10-18	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7		OBE

Answer all FIVE Questions

1 (a)	Explain in detail the features of Nimbits.	[10]	CO2	L2
	OR			
(b)	Explain in brief about the energy consumption in transceiver (transmitter and receiver) section in sensor node.	[10]	CO4	L2
2 (a)	What is Wireless sensor network. Explain the challenges for WSN.	[10]	CO4	L2
	OR			
(b)	Differentiate between single hop and multiple hop networks and explain the types of node mobility.	[10]	CO4	L2
3 (a)	Write a short note on embedded operating system and explain about different programming paradigms.	[10]	CO4	L2
	OR			
(b)	Write a explanatory note on TinyOS and nesC.	[10]	CO4	L2

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

Internal Assessment Test 2– Sept. 2018

Sub:	IOT &WSN					Sub Code:	15EC752	Branch:	TCE
Date:	17-10-18	Duration:	90 min's	Max Marks:	50	Sem / Sec:	7		OBE

Answer all FIVE Questions

1 (a)	Explain in detail the features of Nimbits.	[10]	CO2	L2
	OR			
(b)	Explain in brief about the energy consumption in transceiver (transmitter and receiver) section in sensor node.	[10]	CO4	L2
2 (a)	What is Wireless sensor network. Explain the challenges for WSN.	[10]	CO4	L2
	OR			
(b)	Differentiate between single hop and multiple hop networks and explain the types of node mobility.	[10]	CO4	L2
3 (a)	Write a short note on embedded operating system and explain about different programming paradigms.	[10]	CO4	L2
	OR			
(b)	Write a explanatory note on TinyOS and nesC.	[10]	CO4	L2

4 (a)	Explain the single node architecture with necessary hardware components	[10]	CO4	L2
	OR			
(b)	Write a short note on design principles of WSN .	[10]	CO4	L2
5 (a)	Write a short note on Optimization goals and figure of merit.	[6]	CO4	L2
(b)	Write a short note on network gateway	[4]	CO4	L2
	OR			
(c)	Write a short note on energy scavenging	[4]	CO4	L2
(d)	Write short node on operational state of sensor node with different power consumption.	[6]	CO4	L2

4 (a)	Explain the single node architecture with necessary hardware components .	[10]	CO4	L2
	OR			
(b)	Write a short note on design principles of WSN	[10]	CO4	L2
5 (a)	Write a short note on Optimization goals and figure of merit.	[6]	CO4	L2
(b)	Write a short note on network gateway	[4]	CO4	L2
	OR			
(c)	Write a short note on energy scavenging	[4]	CO4	L2
(d)	Write short node on operational state of sensor node with different power consumption. .	[6]	CO4	L2

controllers_Monitors. user_id can be text or number, such controller12345.

6.4.2 IoT Cloud-based Data Collection, Storage and Computing Services Using Nimbits

Nimbits enables IoT on an open source distributed cloud. Nimbits cloud PaaS deploys an instance of Nimbits Server at the device nodes. Nimbits functions as an M2M system data store, data collector and logger with access to historical data. Nimbits architecture is a cloud-based Google App Engine. Nimbits server is a class hierarchy com.nimbits.server.system.ServerInfo of java.lang.Object. Nimbits PaaS services offer the following features:

- Edge computing locally on embedded systems, built up of local applications. It runs the rules and pushes important data up to the cloud running when connected over Internet and an instance of Nimbits Server hosts at the device nodes which is then enabled.
- It supports multiple programming languages, including Arduino, new Arduino library, push functions from Arduino cloud, JavaScript, HTML or the Nimbits.io Java library.
- Nimbits server functions as a backend platform. Nimbits data point can relay data between the software systems; or hardware devices such as Arduino, using the cloud as a backend.
- An open source Java library called nimbits.io enables easy development of JAVA, web and Android solutions (Nimbits data, alerts, messages on mobile).
- It provides a rule engine for connecting sensors, persons and software to the cloud and one another. Rules can be for calculations, statistics, email alerts, xmpp messages (Section 3.3.3), push notifications and more.

- It provides a data logging service and access, and stores the historical data points and data objects.
- Storage in any format that can be serialised into a string, such as JSON or XML.
- It filters the noise and important changes sent to another larger central instance.
- It processes a specific type of data and can store it.
- Time- or geo-stamping of the data.
- Nimbits clients provide over Internet, data collection in real time, charts, chart and graphical plots of collected data and data entry.
- Data visualisation for data of connected sensors to IoT devices.
- Supports the alerts subscription, generation and sending in real time over the Internet.
- It creates streams of data objects and stores them in a data point series.
- Data accessibility and monitoring from anywhere, and is used to shape the behaviour of connected devices and software.
- It supports the mBed™, Arduino, Raspberry Pi based and other hardware platform based IoT devices.
- Web service APIs are easy to implement on device hardware acting as clients to Nimbits web services, and connect to the web service and send data.
- It deploys software on Google App Engine, any J2EE server on Amazon EC2 or on a Raspberry Pi.

Figure 6.3 shows connected devices, sensor nodes, network data points, Nimbits server, deployment at the device network nodes, and networked with the Nimbits Server (PaaS, SaaS and IaaS services) at cloud for applications and services.

Architecture shown in Figure 6.3 shows a NimbitsServerL which deploys at each device node and is an instance of the NimbitsSeverS at the cloud. Each NimbitsServerL of the device node generates the calculation objects for device nodes.

Each node also hosts an XMPPServerL, an instance of the XMPPSeverC at the cloud.

XMPPServerL deploys at each device node and generates the data feed channels for the XMPP messages and alerts.Each XMPPServerL sends feeds to XMPPSeverC.

Data Points

Data point means a collected value of a sensor in a group of sensors. Data points organise the data in a number of ways. For example, points can have child points (child points mean subpoints; *for example, if light level is a data point then light on or off is a child point and light level above or below the threshold can be another child point.*)

Points can be in the folders. The folders can go as deep as like in a tree (Tree means a folder having several subfolders, a subfolder having several subfolders, till the leaf folder).

Any type of document can upload and organise them with the points. Files can be shared publicly or with the connections.

A subscription data feed is a special point for each user that logs system messages, alerts from other points which are subscribed by a service and more.

Q2(a)

1) What is wireless sensor network? Explain the challenges for WSN.

→ Wireless Sensor networks are networks that consists of sensors which are distributed in an ad hoc manner.

Challenges for WSN's:

- Characteristic requirements
- Required mechanisms

- Characteristic requirements:

→ Type of service

→ Quality of service

→ Fault tolerance

→ Lifetime

→ Scalability

→ Wide range of densities

→ Maintainability

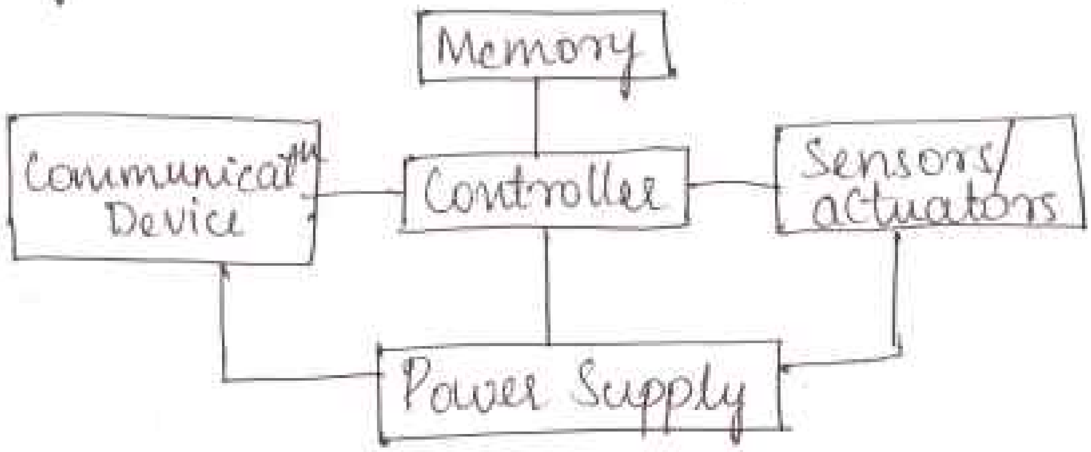
→ Programmability

- Required Mechanisims.

→ Multihop wireless communication
→ Energy-efficient operation
→ Auto-configuration
→ Collaboration and in-network processing
→ Data centric
→ Locality
→ Exploit trade-off's.

Q4(a).
2) Explain the single node architecture with necessary hardware components.

→ A basic sensor node contains 5 main components such as controller, memory, sensors and actuators, communication devices and power supply unit.



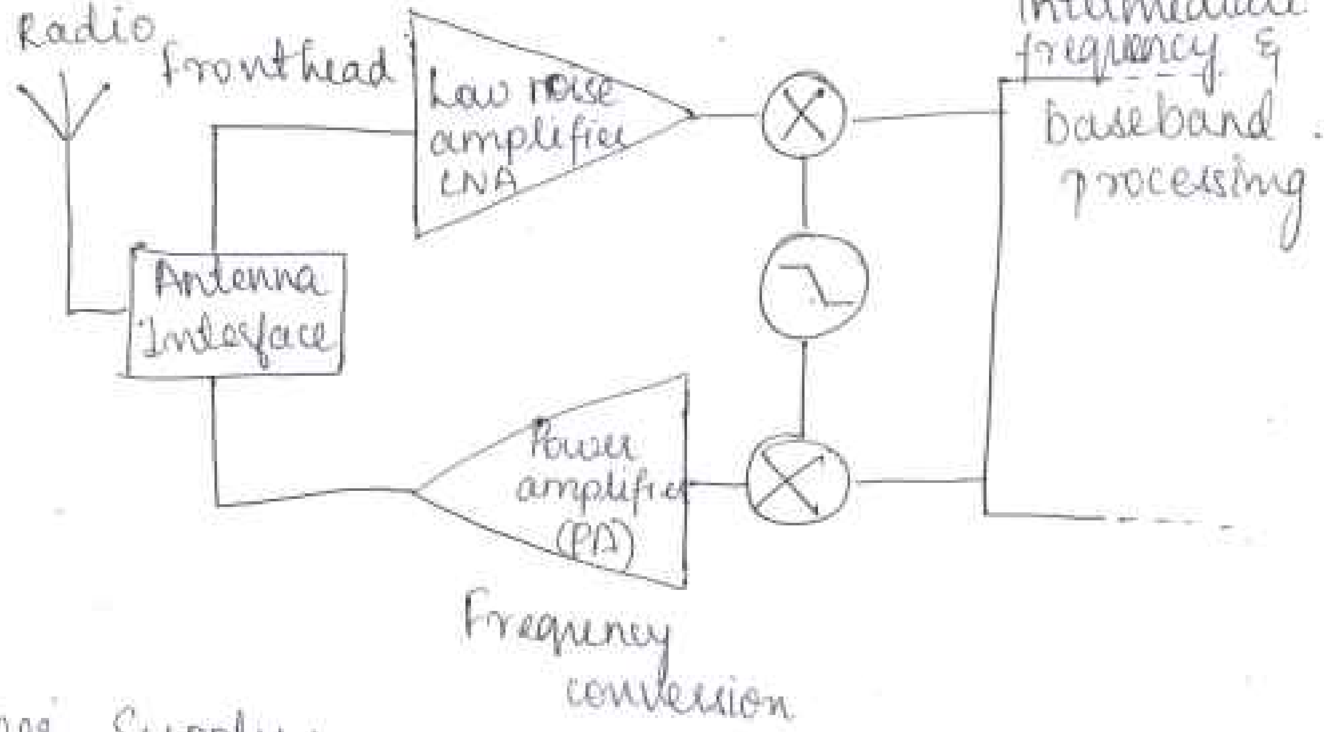
Sensor node Hardware components.

i] <u>Controller:</u>
The controller is the core of a wireless sensor node, it process all the relevant data, capable of executing arbitrary code.
It collects data from sensors, processes this data, decides when and where to send it, similarly receives data from other sensor nodes and decides on the actuator's behaviour.

ii] <u>Memory</u>: Memory is required to store programs and intermediate data; usually, different types of memory are used for programs and data. In WSN there is a need for RAM to store intermediate sensor readings, packets from other nodes & so on. RAM is fast, but loses content if power supply is interuppted. Hence we use RAM or EEPROM.

iii] Sensors and Actuators:
They are the actual interface to the physical world, they observe or control physical parameters of the environment.
Sensors can be categorized as active, passive omni directional, and passive narrow beam sensors.
Actuators are just about as diverse as sensors yet. It is used for converting electrical signals into physical phenomenon.

iv] Communication Device:
To turn nodes into a n/w, a device is required for sending & receiving info over a wireless channel. The communication device is used to exchange data b/w individual nodes. RF based communication is by far the most relevant one as it fits all requirements of most WSN applications.
For communication, both transmitter & receiver are required in a sensor node to convert a bit stream from a controller & convert them to & from radio waves. For 2 tasks a combined device called transceiver is used.



v] Power Supply:
No tied power supply is available, hence some form of batteries are necessary to provide energy.
There are essentially 2 features:
1) Storing energy 2) Energy Scavenging

Q5(c)

4) Write a short note on energy scavenging.

→ Depending on application, high capacity batteries that last for long times with negligible self-discharge rate and that can efficiently provide small amounts of current. For this energy scavenging is used, which is the process of recharging battery with gathered from the environment. For example,

→ Photovoltaics : solar cells

→ Temperature gradient : difference in temperature converted to electrical energy

→ Vibrations : mechanical energy

→ Flow of air / liquid : Windmills & turbines

Q5(d)

3) Operational state of sensor with diff. power consumpt^n

→ Microcontroller energy consumpt^n: For a controller, typical states are active, idle & sleep. The energy saving in microcontroller is denoted given by

$$E_{saved} = (t_{event} - t_1) P_{active} - (\tau_{down} (P_{active} + P_{sleep})/2 + (t_{event} - t_1 - \tau_{down}) P_{sleep})$$

The energy overhead is denoted by $E_{overhead}$ $\tau_{up} (P_{active} + P_{sleep})/2$

Ex Intel Strong ARM.

- In normal mode, the power consumption is upto 400 mW.

- In the ~~deepest sleep~~ idle mode (~~[illegible]~~) ~~only~~ consumes ~~[illegible]~~ 100mW.
- In idle sleep mode, power consumpth is upto 50μW

→ Atmel ATmega:
In atmel ATmega, power consumpth varies b/w 6mW & 15mW, in idle and active modes & is about 75μW in power down modes

Q1(b)

5) Energy consumption in transceivers

→ A radio transceiver has essentially 2 tasks as transmitting & receiving data between a pair of nodes. To maintain low energy consumption, the tranceivers should be turned off most of the time and only be activated when necessary.

→ The energy consumed by a transmitter is due to RF signal generation & due to electronic components necessary for frequency synthesis, conversion, filters and so on.

→ Similar to the transmitter, the receiver can be either turned off or turned on.

6) Write a short note on embedded OS & explain

→ An operating system (OS) is system software that manages computer hardware & software resources, ie, acts as an intermediary between programs & computer hardware. An embedded system is some combination of hardware or software, that is specifically designed for a particular function.

→ Embedded operating system (EOS) are designed to be used in embedded computer systems. EOS are able to operate with a limited no of resources. They are compact & efficient.

Q3(b)

7) Note on TinyOS

→ It is an opensource, flexible & application specific OS for wireless sensor n/w. WSN consists of a large no of tiny & low-power nodes, each of which executes simultaneous & reactive programs that must work with strict memory & power consumptn. TinyOS meets these challenges.

Features:

- Event based concurrency model.
- Component-based architecture.
- TinyOS's component library includes network protocols, distributed services, sensor drivers, etc.
- TinyOS's event-driven execution model.

Q2(b)

8) Differences b/w single hop & multiple hop network & explain the types of node mobility.

→ In wireless sensor, n/w mobility can appear in 3 main frames:

- Node Mobility - The wireless sensor nodes themselves can be mobile. The meaning of such mobility is highly applicatⁿ dependant

- Sink mobility. The information sinks can be mobile while this can be a special case of node mobility the important aspect is the mobility of an info sink that is not part of the sensor n/w.

- Event Mobility. In applications like event detection and in particular tracking applicatⁿˢ the cause the events or the objects to be tracked can be mobile.

Single vs multi hop

→ From the basics of radio communicatⁿ & the inherent power limitation of radio comm follows a limitatⁿ on the feasible distance b/w a sender & receiver. Because of this limited distance, the simple, direct communicatⁿ b/w source & sink is not always possib specifically in WSNs, which are intended to cover a lot of ground or that operate in difficult radio environments with strong attenuation.

To overcome such limited distances, an obvious way out is to use relay statⁿˢ, with data packets taking multi hops from the source to the sink. This concept of multihop n/w is particularly attractive for WSNs as the sensor nodes themselves can act as such relay nodes, foregoing the need for additional equipment.

10) ~~Short note on design principles of WSN.~~

Q5(a)

9) Short note on optimizat^n goals & figure of merit.

→ For all WSN sceenaniors & application types have to face the challenges such as:

- How to optimize a network, compare solut^ns?
- How to decide the better approach
- How to turn inaccurate optimizat^n goals into measurable figures

General answers for the above questions:

* Quality of service
* Energy efficiency
* Scalability
* Robustness.

i] Quality of service: WSNs differ from other conventional n/w in the type of service they offer. These n/w essentially only move bits from one place to another

Generic possibilities are:-

→ Event detection/ reporting probability
→ Event classification error.
→ Event detection delay
→ Missing reports
→ Approx accuracy
→ Tracking accuracy.

ii] Scalability : The ability to maintain performance characteristics irrespective of the size of n/w referred to scalability. With WSN potentially consisting of thousands of nodes, scalability is an obviously essential requirement. The need for extreme scalability has direct consequences for the protocol design.

iii] Robustness:
wireless sensor n/w should also exhibit an appropriate robustness. They should not fail just because a limited no. of nodes run out of energy or bcz their environment changes. If possible, these failures have to be compensated by finding others routes.

Q4(b)
10) Design principles of WSN:

- Distributed organizat^n.

→ Distributed means every part of organizat^n is acting separately as an autonomous unit, there is no central controller w/o interference of the other.

→ If WSN used centralized architecture, then if the central node fails, then the entire n/w will collapse.

→ Hence a distributed control is to increase reliability.

- In-network processing
It is called in-network processing as processing is done inside the sensor network close to the source.
It is divided into 2 categories:-

* Aggregatn based techniques:
Let nodes periodically measure data, but it is not necessary to send all the collected data. Data with change in average value, or a huge difference can be sent.
Hence aggregatn condenses & removes redundant info by reducing the no. of bits.

* Approx techniques: It focuses on reducing amount of data packets to be transmitted in-network when the accuracy of data collection is important. It defines what is the average/max absolute or relative error w.r.t the actual function.

- Distributed source coding & collaborative signal processing:
Where networking is distributed, each node has separate processor & working independently. CSP techniques are used for combining the data collected from different nodes with in a particular active region.

- Geographic Adaptive Fidelity (GAF): It is one of the most popular location based routing protocol. It uses the nodes location info to transfer the data. It reduces the use of energy consumption of nodes and increase the n/w lifetime.

- Data centric networking:
It defines that data is the center of attention, not the identity of nodes, called the data centric approach.

- Overlay n/w & distributed hash tables (DHT):
If the data is retrieved from an unknown source, like in peer-to-peer n/w then use DHT. The desired data is identified by a key (hash) and then DHT provides one source for the data associated with key.

- Publish/subscribe: Nodes can publish data, can subscribe to any particular kind of data. Once data of a certain type has been published, it is delivered to all subscribes.

- Databases: WSN can be considered as dynamic database (SQL). The sensors are considered as virtual tables to which some relational operators can be applied.

- Exploit location info: For some applications location info of sensors is crucial info.

- Exploit activity patterns: Once an event has happened, it can be detected by a no. of sensors, breaking into frenzy activity called as event shower effect.

- Exploit heterogeneity.

Q.5(b)

11) Write a short note on n/w gateway.

→ The n/w has to be able to interact with other info devices for ex. to read the temperature sensors in one's home while travelling & accessing the internet via a wireless

→ Gateway allows the WSN to exchange the data with other devices like mobile phones.

→ Gateway node bridges a gap b/w WSN and other communication devices.

→ Gateway is equipped with a radio transceiver or some standard wireless communicatⁿ technique like IEEE 802.11.

Challenges in WSN to internet comm.

→ Let sensor node 'ALICE' wants to deliver an alarm msg to an internet host.

→ But some issues occur like:
- How to handle several gateways.
- Choose 'best' gateway
- Finding host IP address to which it should be sent.

→ Let internet based entry try to access the services of WSN.

→ If requesting terminal can directly communicate like mobile then no particular treatment is neccessary, but if this is not the case, complexity increases.

3a) Embedded Operating Systems and different programming paradigms.

→ An operating system is a symtem which interfaces both the hardware and software components of a computer system.

→ They act as a backbone, making it easier for the user to access the hardware and software components.

→ An Embedded Operating system (EOS) is the operating system that interface the hardware components of the embedded system with its software components.

→ One of the most common EOS is TinyOS. It satisfies all the requirements an embedded operating system needs for its functioning.

→ EOS has different programming paradigms. They are:

Concurrent Programming paradigm:
In concurrent programming, multiple functions / programs are carried out simultaneously, without lower the speed or affecting the quality. Concurrent programming reduces the time consumption as multiple programs are running at the same time

Parallel concurrency:
Parallel concurrency is extremely useful as it speeds up the process. The main advantage of this over threads is that if the threads are lost once, they cannot be regained back where as here, we can recover our program by simply turning off and turning on the system again.

Event-based:
Event-based works on the basis of the current event that has occured. For example click of a mouse.
The next action is based on the output of the recently occured event.

CMR

★ Interface paradigm:

GUI - Graphic user interface is the most common example.

It gives control of events to the user.

This interface has graphic icons, ~~which~~ each of which have their own functionality.