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

Internal Assessment Test 2 – November 2024

Sub:	**Internet of Things**					Sub Code:	**21CS735**	Branch:	**AInDS**
Date:		Duration:	**90 minutes**	Max Marks:	**50**	Sem	**VII**		**OBE**

Answer any FIVE Questions			**MARKS**	**CO**	**RBT**
Q1	a	Explain different data types used in IoT applications with example	[5]	CO3	1
	b	What are the typical data offload locations available in the context of IoT?	[5]	CO3	2
Q 2		Explain role of Processing Topologies and explain on-site processing in detail	[10]	CO3	2
Q 3		What are the main considerations for IoT device Design and Selection	[10]	CO3	2
Q 4	a	Describe the working of a Z-Wave implementation.	[5]	CO4	2
	b	Explain in detail Near Field Communication (NFC)	[5]	CO4	1
Q 5		Draw and Explain Architecture of ISA100.11A	[10]	CO4	2
Q 6		Compare OSI stack with DASH7 stack	[10]	CO4	3

CI **CCI** **HOD**

Q 1 a) Explain different data types used in IoT applications with example

Structured data

These are typically text data that have a pre-defined structure [1]. Structured data are associated with relational database management systems (RDBMS). These are primarily created by using length-limited data fields such as phone numbers, social security numbers, and other such information. Even if the data is human or machine generated, these data are easily searchable by querying algorithms as well as human generated queries. Common usage of this type of data is associated with flight or train reservation systems, banking systems, inventory controls, and other similar systems. Established languages such as Structured Query Language (SQL) are used for accessing these data in RDBMS. However, in the context of IoT, structured data holds a minor share of the total generated data over the Internet.

Unstructured data

In simple words, all the data on the Internet, which is not structured, is categorized as unstructured. These data types have no pre-defined structure and can vary according to applications and data-generating sources. Some of the common examples of human-generated unstructured data include text, e-mails, videos, images, phone recordings, chats, and others [2]. Some common examples of machine-generated unstructured data include sensor data from traffic, buildings, industries, satellite imagery, surveillance videos, and others. As already evident from its examples, this data type does not have fixed formats associated with it, which makes it very difficult for querying algorithms to perform a look-up. Querying languages such as NoSQL are generally used for this data type.

Q 1 b) What are the typical data offload locations available in the context of IoT?

The choice of offload location decides the applicability, cost, and sustainability of the

IoT application and deployment. We distinguish the offload location into four types: **Edge**: Offloading processing to the edge implies that the data processing is
facilitated to a location at or near the source of data generation itself. Offloading to the edge is done to achieve aggregation, manipulation, bandwidth reduction, and other data operations directly on an IoT device.

• **Fog**: Fog computing is a decentralized computing infrastructure that is utilized to conserve network bandwidth, reduce latencies, restrict the amount of data unnecessarily flowing through the Internet, and enable rapid mobility support for IoT devices. The data, computing, storage and applications are shifted to a place between the data source and the cloud resulting in significantly reduced latencies and network bandwidth usage.

• **Remote Server**: A simple remote server with good processing power may be used with IoT-based applications to offload the processing from resourceconstrained IoT devices. Rapid scalability may be an issue with remote servers, and they may be costlier and hard to maintain in comparison to solutions such as the cloud.

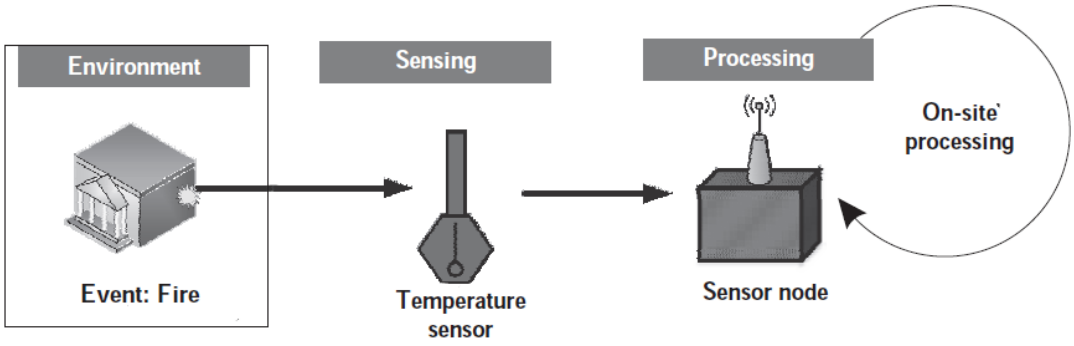
• **Cloud**: Cloud computing is a configurable computer system, which can get access to configurable resources, platforms, and high-level services through a shared pool hosted remotely. A cloud is provisioned for processing offloading so that processing resources can be rapidly provisioned with minimal effort over the Internet, which can be accessed globally. Cloud enables massive scalability of solutions as they can enable resource enhancement allocated to a user or solution in an on-demand manner, without the user having to go through the pains of acquiring and configuring new and costly hardware

Q 2) Explain role of Processing Topologies and explain on-site processing in detail

The identification and intelligent selection of processing requirement of an IoT application are one of the crucial steps in deciding the architecture of the deployment. A properly designed IoT architecture would result in massive savings in network bandwidth and conserve significant amounts of overall energy in the architecture while providing the proper and allowable processing latencies for the solutions associated with the architecture. we can divide the various processing solutions into two large topologies: 1) On-site and 2) Off-site. The off-site processing topology can be further divided into the following: 1) Remote processing and 2) Collaborative processing.

On-site processing

As evident from the name, the on-site processing topology signifies that the data is processed at the source itself. This is crucial in applications that have a very low tolerance for latencies. These latencies may result from the processing hardware or the network (during transmission of the data for processing away from the processor). Applications such as those associated with healthcare and flight control systems (realtime systems) have a breakneck data generation rate. These additionally show rapid temporal changes that can be missed (leading to catastrophic damages) unless the processing infrastructure is fast and robust enough to handle such data. Figure 6.2 shows the on-site processing topology, where an event (here, fire) is detected utilizing a temperature sensor connected to a sensor node. The sensor node processes the information from the sensed event and generates an alert. The node additionally has the option of forwarding the data to a remote infrastructure for further analysis and storage.



Q 3) What are the main considerations for IoT device Design and Selection

The main consideration of minutely defining an IoT solution is the selection of the processor for developing the sensing solution (i.e., the sensor node). This selection is governed by many parameters that affect the usability, design, and affordability of the designed IoT sensing and processing solution. In this chapter, we mainly focus on the deciding factors for selecting a processor for the design of a sensor node. The main factor governing the IoT device design and selection for various applications is the processor. However, the other important considerations are as follows.

• **Size**: This is one of the crucial factors for deciding the form factor and the energy consumption of a sensor node. It has been observed that larger the form factor, larger is the energy consumption of the hardware. Additionally, large form factors are not suitable for a significant bulk of IoT applications, which rely on minimal form factor solutions (e.g., wearables).

• **Energy**: The energy requirements of a processor is the most important deciding factor in designing IoT-based sensing solutions. Higher the energy requirements, higher is the energy source (battery) replacement frequency. This principle automatically lowers the long-term sustainability of sensing hardware, especially for IoT-based applications.

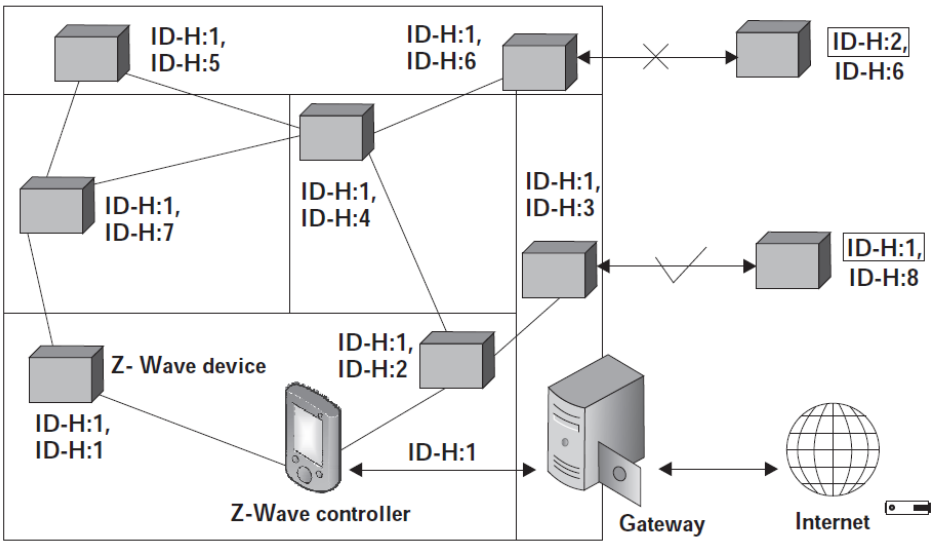
• **Cost**: The cost of a processor, besides the cost of sensors, is the driving force in deciding the density of deployment of sensor nodes for IoT-based solutions. Cheaper cost of the hardware enables a much higher density of hardware deployment by users of an IoT solution. For example, cheaper gas and fire detection solutions would enable users to include much more sensing hardware for a lesser cost.

• **Memory**: The memory requirements (both volatile and non-volatile memory) of IoT devices determines the capabilities the device can be armed with. Features such as local data processing, data storage, data filtering, data formatting, and a host of other features rely heavily on the memory capabilities of devices. However, devices with higher memory tend to be costlier for obvious reasons.

• **Processing power**: As covered in earlier sections, processing power is vital (comparable to memory) in deciding what type of sensors can be accommodated with the IoT device/node, and what processing features can integrate on-site with the IoT device. The processing power also decides the type of applications the device can be associated with. Typically, applications that handle video and image data require IoT devices with higher processing power as compared to applications requiring simple sensing of the environment.

• **I/O rating**: The input–output (I/O) rating of IoT device, primarily the processor, is the deciding factor in determining the circuit complexity, energy usage, and requirements for support of various sensing solutions and sensor types. Newer processors have a meager I/O voltage rating of 3.3 V, as compared to 5 V for the somewhat older processors. This translates to requiring additional voltage and logic conversion circuitry to interface legacy technologies and sensors with the newer processors. Despite low power consumption due to reduced I/O voltage levels, this additional voltage and circuitry not only affects the complexity of the circuits but also affects the costs.

Q 4 a) Describe the working of a Z-Wave implementation.



Z-Wave is an economical and less complicated alternative to Zigbee. It was developed by Zensys, mainly for home automation solutions [11]. It boasts of a power consumption much lower than Wi-Fi, but with ranges greater than Bluetooth. This feature makes Z-Wave significantly useful for home IoT use by enabling inter-device communication between Z-wave integrated sensors, locks, home power distribution systems, appliances, and heating systems. Figure 7.16 shows the network architecture of the Z-Wave protocol.

Z-Wave controller Gateway Internet

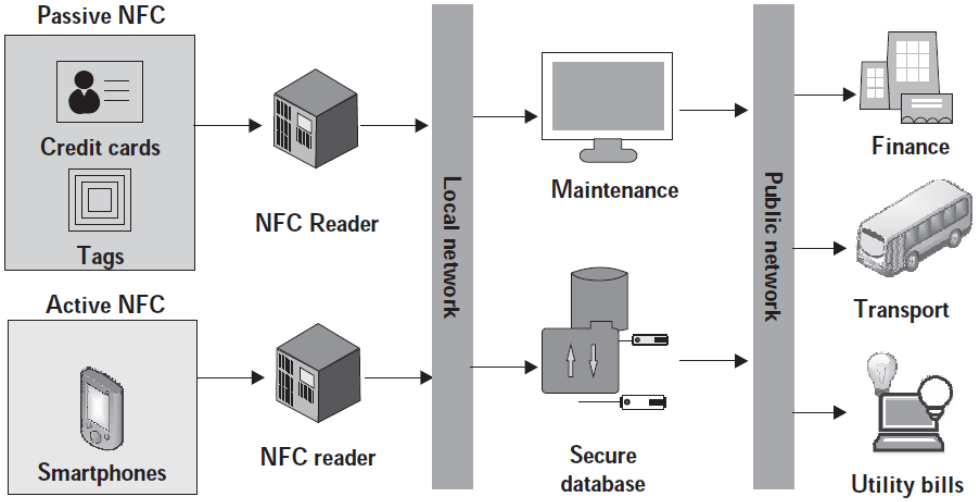
Z- Wave The Z-Wave operational frequency is in the range of 800–900 MHz, which makes it mostly immune to the interference effects of Wi-Fi and other radios utilizing the 2.4 GHz frequency band. Z-wave utilizes gaussian frequency shift keying (GFSK) modulation, where the baseband pulses are passed through a Gaussian filter before modulation. The filtering operation smoothens the pulses consisting of streams of –1 and 1 (known as pulse shaping), which limits the modulated spectrum's width. A Manchester channel encoding is applied for preparing the data for transmission over the channel.

Q 4 b) Explain in detail Near Field Communication (NFC)

Near field communication (NFC) was jointly developed by Philips and Sony as a short-range wireless connectivity standard, enabling peer-to-peer (P2P) data exchange network. Communication between NFC devices is achieved by the principle of magnetic induction, whenever the devices are brought close to one another [9]. NFC can also be used with other wireless technologies such as Wi-Fi after establishing and configuring the P2P network. The communication between compatible devices requires a pair of transmitting and receiving devices. The typical NFC operating frequency for data is 13.56 MHz, which supports data rates of 106, 212, or 424 kbps.

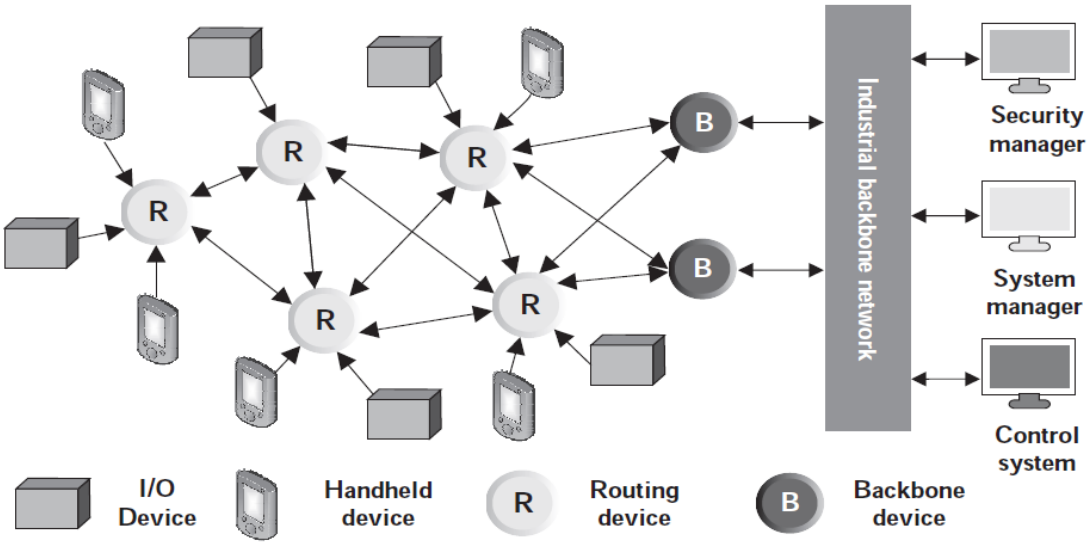
NFC devices can be grouped into two types: 1) passive NFC and 2) active NFC. Figure 7.13 shows the various NFC types, components, and its usage.

A small electric current is emitted by the NFC reader, which creates a magnetic field that acts as a bridge in the physical space between two NFC devices. The generated EM (electromagnetic) field is converted back into electrical impulses through another coil on the client device. Data such as identifiers, messages, currency, status, and others can be transmitted using NFCs. NFC communication and pairing are speedy due to the use of inductive coupling and the absence of manual pairing.



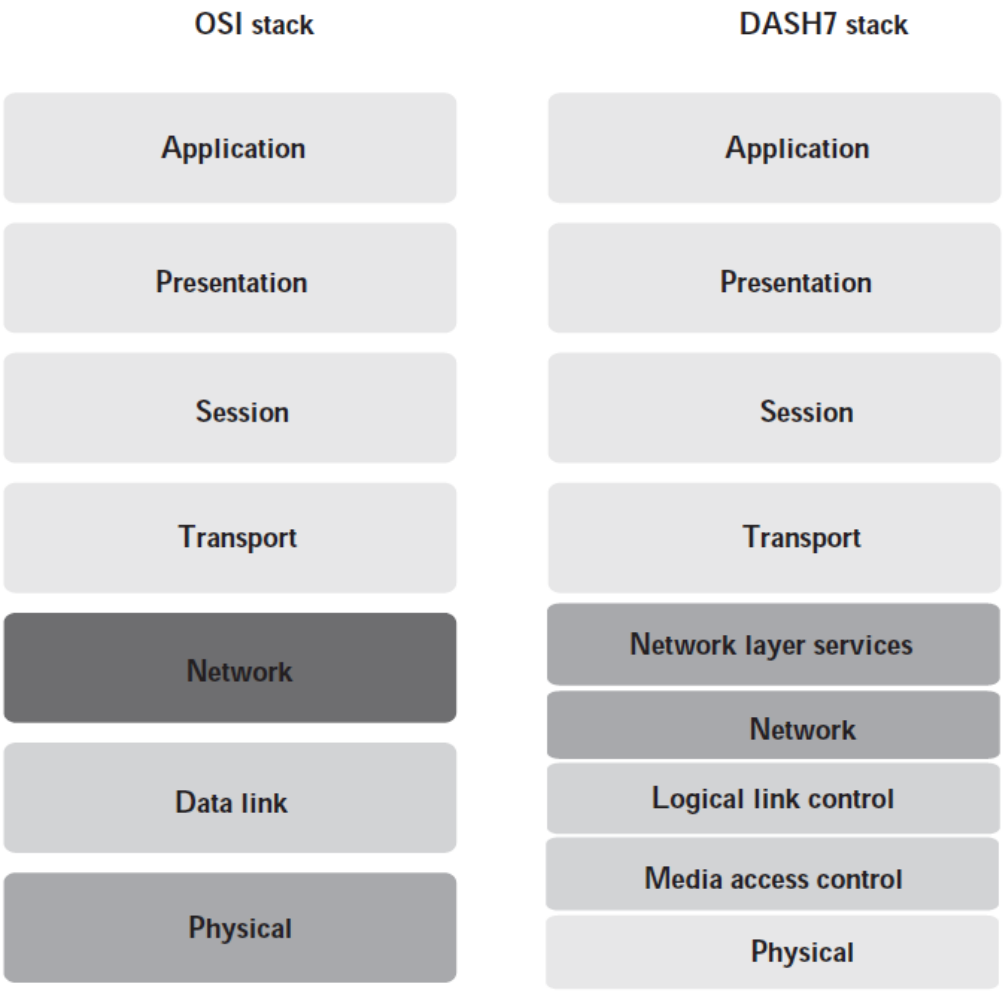
Passive NFC devices do not need a power source for communicating with the NFC reader. Tags and other small transmitters can act as passive NFC devices. However, passive devices cannot process information; they simply store information, which is read by an NFC reader. In contrast, active NFC devices can communicate with active as well as passive NFC devices. Active devices are capable of reading as well as writing data to other NFC terminals or devices. Some of the most commonly used NFC platforms are smartphones, public transport card readers, and commercial touch payment terminals.

Q 5) Draw and Explain Architecture of ISA100.11A



The ISA100.11A architecture consists of
the following: 1) field devices and 2) backbone devices. Field devices may be nonrouting I/O devices, handheld devices, routing devices, which may or may not be fixed or mobile. For industrial usage, the inclusion of portable and mobile devices is highly desirable as it allows floor supervisors and workers to keep checking various parts of the plant without the need for dedicated devices for each part. In contrast, backbone devices include backbone routers, gateways, the system manager, and the security manager, which are kept fixed and not portable. The ISA100.11A architecture provides support for mesh, star, and star–mesh topologies. The connected devices in ISA100.11A are collectively referred to as the downLink (DL) subnet. A wireless industrial sensor network (WISN) gateway connects the ISA100.11A network to the plant network.

Q 6) Compare OSI stack with DASH7 stack



DASH7 stack includes support for cheap processing systems by virtue of its integrated file system. Figure 7.15 shows the protocol stack of DASH 7 in comparison to the ISOOSI stack. DASH7 gateways can query devices in proximity to it without waiting for pre-defined time-slots to listen to end-device beacons.

DASH7 is capable of very dense deployments, has a low memory footprint, consumes minuscule power, and considered by many as a bridge between NFC and IoT communication systems. It can also be used to enable tag-to-tag communication without needing the tags to pass their information through a base station or a tag reader. This feature of DASH7 is quite synonymous with the multinode hopping mesh networks found in Zigbee and Z-wave. The reported range of DASH7 is between 1 to 10 km and a typical querying latency of 1 to 10 seconds.