USN | | | | | | | | | |

Internal Assessment Test 2 – Aug 2023

Sub:	Internet of Things					Sub Code:	22BETCK25	Branch:	
Date:	10/08/2023	Duration:	90 min's	Max Marks:	50	Sem/Sec:	Physics Cycle		OBE
Answer any FIVE FULL Questions							MARKS	CO	RBT
1	What are the main features of shape memory polymers? Differentiate between hydraulic and pneumatic actuators with examples						5+5	CO2	L2
2	What is IoT? What are the various decision making approaches chosen for offloading data in IoT?						2+8	CO2	L2
3	Discuss about the data format used in IoT and also explain about the importance of processing of data in IoT.						10	CO3	L2
4	Differentiate between structured and unstructured data. Explain how is collaborative processing different from remote processing?						3+7	CO3	L2
5	What factors are to be considered while deciding on the data offload location? What are the pros and cons of on-site and off-site processing?						5+5	CO3	L2
6	What is virtualization? How is it useful for end users? Explain the types of virtualizations.						10	CO3	L2

Faculty Signature CCI Signature HOD Signature

Scheme Of Evaluation
Internal Assessment Test II–Aug 2023

Sub:	INTERNET OF THINGS					Code:	22BETCK25
Date:	10/08/2023	Duration:	90 mins	Max Marks:	50	Sem: II	Branch:

Note: Answer 5 Questions

	Description	Marks Distribution		Max Marks
1	What are the main features of shape memory polymers? Differentiate between hydraulic and pneumatic actuators with examples		5+5	10
	• Features of shape memory polymer • Differences	5 5		
2	What is IoT? What are the various decision making approaches chosen for offloading data in IoT?		10	10
	• Definition of IOT • Decision making approaches listing and explanation	2 8		
3	Discuss about the data format used in IoT and also explain about the importance of processing of data in IoT.		10	10
	• Data formats listing • Explanation • Importance of processing of data in IoT.	2 3 5		
4	Differentiate between structured and unstructured data. Explain how is collaborative processing different from remote processing?		10	10
	• Differences structured and unstructured data	3		
	• is collaborative processing different from remote processing?	7		
5	What factors are to be considered while deciding on the data offload location? What are the pros and cons of on-site and off-site processing?	10		

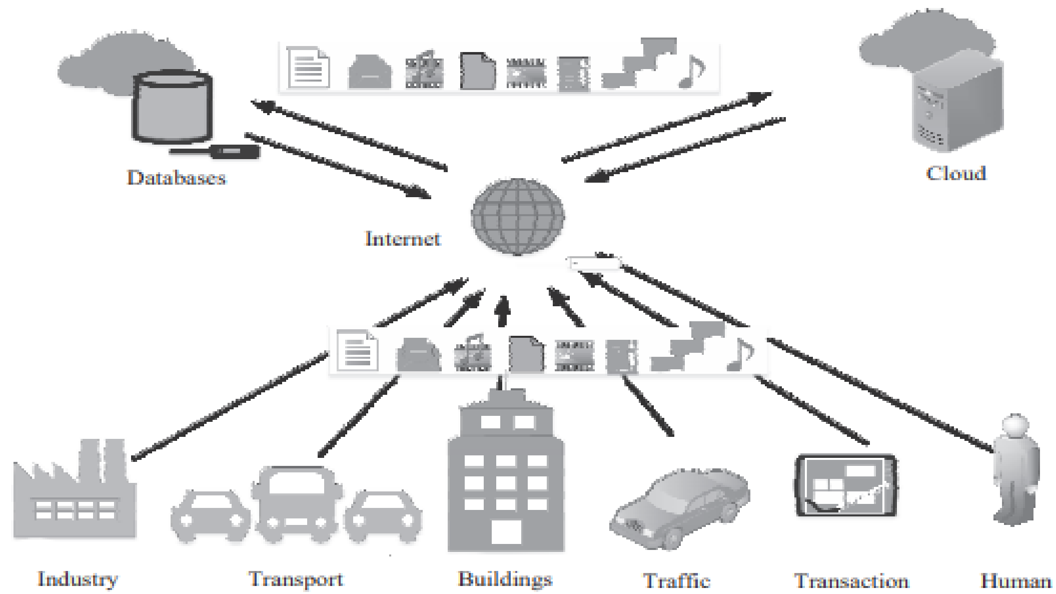
—

	• Factors for data offloading listing • Pros and cons listing	5 5		
6	What is virtualization? How is it useful for end users? Explain the types of virtualizations		10	10
	• Virtualization usage • Types of virtualization listing • Explanation	5 2 3		

CMR INSTITUTE OF TECHNOLOGY
DEPT OF ECE
INTERNET OF THINGS
INTERNAL ASSESSMENT TEST –2
10-08-2023
SOLUTIONS

Q.No.	Question	Marks
1	What are the main features of shape memory polymers? Differentiate between hydraulic and pneumatic actuators with examples	10
Sol	Shape memory polymers (SMP) are considered as smart materials that respond to some external stimulus by changing their shape, and then revert to their original shape once the affecting stimulus is removed. □ Features such as high strain recovery, biocompatibility, low density, and biodegradability characterize these materials. □ SMP-based actuators function similar to our muscles. □ Modern-day SMPs have been designed to respond to a wide range of stimuli such as pH changes, heat differentials, light intensity, and frequency changes, magnetic changes, and others. □ Photopolymer/light-activated polymers (LAP) are a particular type of SMP, which require light as a stimulus to operate. □ LAP-based actuators are characterized by their rapid response times. □ Using only the variation of light frequency or its intensity, LAPs can be controlled remotely without any physical contact. □ The polymer retains its shape after removal of the activating light. In order to change the polymer back to its original shape, a light stimulus of a different frequency has to be applied to the polymer.	5
	(i) Hydraulic actuators: □ A hydraulic actuator works on the principle of compression and decompression of fluids. □ These actuators facilitate mechanical tasks such as lifting loads through the use of hydraulic power derived from fluids in cylinders or fluid motors. □ The mechanical motion applied to a hydraulic actuator is converted to either linear, rotary, or oscillatory motion. □ The almost incompressible property of liquids is used in hydraulic actuators for exerting significant force. □ These hydraulic actuators are also considered as stiff systems. The actuator's limited acceleration restricts its usage (ii) Pneumatic actuators: □ A pneumatic actuator works on the principle of compression and decompression of gases. □ These actuators use a vacuum or compressed air at high pressure and convert it into either linear or rotary motion.	5

	Pneumatic rack and pinion actuators are commonly used for valve controls of water pipes. Pneumatic actuators are considered as compliant systems. The actuators using pneumatic energy for their operation are typically characterized by the quick response to starting and stopping signals. Small pressure changes can be used for generating large forces through these actuators. Pneumatic brakes are an example of this type of actuator which is so responsive that they can convert small pressure changes applied by drives to generate the massive force required to stop or slow down a moving vehicle. Pneumatic actuators are responsible for converting pressure into force. The power source in the pneumatic actuator does not need to be stored in reserve for its operation.	
2	What is IoT? What are the various decision making approaches chosen for offloading data in IoT?	
Sol	One or more Devices interconnected through local or global connectivity and monitor continuous data and transmit for further processing to remote location or onsite is called IOT. The various decision making approaches chosen for offloading data in IoT **Naive Approach**: This approach is typically a hard approach, without too much decision making. It can be considered as a rule-based approach in which the data from IoT devices are offloaded to the nearest location based on the achievement of certain offload criteria. Although easy to implement, this approach is never recommended, especially for dense deployments, or deployments where the data generation rate is high or the data being offloaded in complex to handle (multimedia or hybrid data types). Generally, statistical measures are consulted for generating the rules for offload decision making. **Bargaining based approach**: This approach, although a bit processing-intensive during the decision making stages, enables the alleviation of network traffic congestion, enhances service QoS (quality of service) parameters such as IoT Processing Topologies and Types 125 bandwidth, latencies, and others. At times, while trying to maximize multiple parameters for the whole IoT implementation, in order to provide the most optimal solution or QoS, not all parameters can be treated with equal importance. Bargaining based solutions try to maximize the QoS by trying to reach a point where the qualities of certain parameters are reduced, while the others are enhanced. This measure is undertaken so that the achieved QoS is collaboratively better for the full implementation rather than a select few devices enjoying very high QoS. Game theory is a common example of the bargaining based approach. This approach does not need to depend on historical data for decision making purposes. **Learning based approach:** Unlike the bargaining based approaches, the learning based approaches generally rely on past behavior and trends of data flow through the IoT architecture. The optimization of QoS parameters is pursued by learning from historical trends and trying to optimize previous solutions further and enhance the collective behavior of the IoT implementation. The memory requirements and processing requirements are high during the decision making stages. The most common example of a learning based approach is machine learning.	**2** **2.5** **3** **2.5**
3	Discuss about the data format used in IoT and also explain about the importance of processing of data in IoT.	

Sol	The data can be broadly grouped into two types based on how they can be accessed and stored: 1) Structured data and 2) unstructured data. 	**2**
	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.	**1.5**
	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 IoT Processing Topologies and Types 117 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.	**1.5**

	Importance of Processing in IoT The vast amount and types of data flowing through the Internet necessitate the need for intelligent and resourceful processing techniques. This necessity has become even more crucial with the rapid advancements in IoT, which is laying enormous pressure on the existing network infrastructure globally. Given these urgencies, it is important to decide—when to process and what to process? Before deciding upon the processing to pursue, we first divide the data to be processed into three types based on the urgency of processing: 1) Very time critical, 2) time critical, and 3) normal. Data from sources such as flight control systems [3], healthcare, and other such sources, which need immediate decision support, are deemed as very critical. These data have a very low threshold of processing latency, typically in the range of a few milliseconds. Data from sources that can tolerate normal processing latency are deemed as time critical data. These data, generally associated with sources such as vehicles, traffic, machine systems, smart home systems, surveillance systems, and others, which can tolerate a latency of a few seconds fall in this category. Finally, the last category of data, normal data,can tolerate a processing latency of a few minutes to a few hours and are typically associated with less data-sensitive domains such as agriculture, environmental monitoring, and others.	**5**
4	Differentiate between structured and unstructured data. Explain how is collaborative processing different from remote processing?	**10**
solu	**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 IoT Processing Topologies and Types 117 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. **Remote processing** This is one of the most common processing topologies prevalent in present-day IoT solutions. It encompasses sensing of data by various sensor nodes; the data is then forwarded to a remote server or a cloud-based infrastructure for further processing and analytics. The processing of data from hundreds and thousands of sensor nodes can be simultaneously offloaded to a single, powerful computing platform; this results in massive cost and energy savings by enabling the reuse and reallocation of the same processing resource while also enabling the deployment of smaller and simpler processing nodes at the	**1.5** **1.5** **2**

site of deployment [4]. This setup also ensures massive scalability of solutions, without significantly affecting the cost of the deployment. Figure 6.3 shows the outline of one such paradigm, where the sensing of an event is performed locally, and the decision making is outsourced to a remote processor (here, cloud). However, this paradigm tends to use up a lot of network bandwidth and relies heavily on the presence of network connectivity between the sensor nodes and the remote processing infrastructure.

Collaborative processing This processing topology typically finds use in scenarios with limited or no network connectivity, especially systems lacking a backbone network. Additionally, this topology can be quite economical for large-scale deployments spread over vast areas, where providing networked access to a remote infrastructure is not viable. In such scenarios, the simplest solution is to club together the processing power of nearby processing nodes and collaboratively process the data in the vicinity of the data source itself. This approach also reduces latencies due to the transfer of data over the network. Additionally, it conserves bandwidth of the network, especially ones connecting to the Internet. Figure 6.4 shows the collaborative processing topology for collaboratively processing data locally. This topology can be quite beneficial for applications such as agriculture, where an intense and temporally high frequency of data processing is not required as agricultural data is generally logged after significantly long intervals (in the range of hours). One important point to mention about this topology is the preference of mesh networks for easy implementation of this topology. **2**

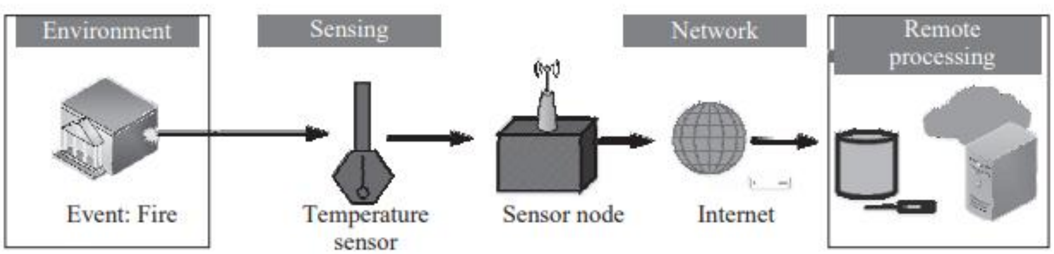


Figure 6.3 Event detection using an off-site remote processing topology **1.5**

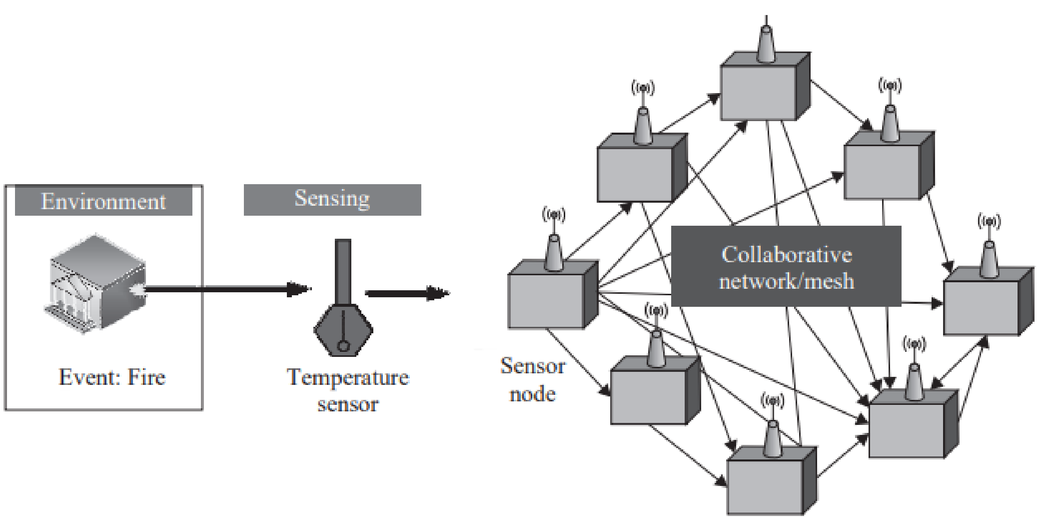


Figure 6.4 Event detection using a collaborative processing topology **1.5**

5	What factors are to be considered while deciding on the data offload location? What are the pros and cons of on-site and off-site processing?	**10**
sol	There are a few offloading parameters which need to be considered while deciding upon the offloading type to choose. These considerations typically arise from the nature of the IoT application and the hardware being used to interact with the application. Some of these parameters are as follows. • **Bandwidth:** The maximum amount of data that can be simultaneously transmitted over the network between two points is the bandwidth of that network. The bandwidth of a wired or wireless network is also considered to be its data-carrying capacity and often used to describe the data rate of that network. • **Latency**: It is the time delay incurred between the start and completion of an operation. In the present context, latency can be due to the network (network latency) or the processor (processing latency). In either case, latency arises due to the physical limitations of the infrastructure, which is associated with an operation. The operation can be data transfer over a network or processing of a data at a processor. • **Criticality:** It defines the importance of a task being pursued by an IoT application. The more critical a task is, the lesser latency is expected from the IoT solution. For example, detection of fires using an IoT solution has higher criticality than detection of agricultural field parameters. The former requires a response time in the tune of milliseconds, whereas the latter can be addressed within hours or even days. **Resources:** It signifies the actual capabilities of an offload location. These capabilities may be the processing power, the suite of analytical algorithms, and others. For example, it is futile and wasteful to allocate processing resources reserved for real-time multimedia processing (which are highly energy-intensive and can process and analyze huge volumes of data in a short duration) to scalar data (which can be addressed using nominal resources without wasting much energy). **• Data volume**: The amount of data generated by a source or sources that can be simultaneously handled by the offload location is referred to as its data volume handling capacity. Typically, for large and dense IoT deployments, the offload location should be robust enough to address the processing issues related to massive data volumes	**5**
	Onsite 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 (real time 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. Offsite processing: The off-site processing paradigm, as opposed to the on-site processing paradigms, allows for latencies (due to processing or network latencies); it is significantly cheaper than on-site processing topologies. This difference in cost is mainly due to the low demands and requirements of processing at the source itself. Often, the sensor nodes are not required to	**5**

<table>
<tr><td></td><td>process data on an urgent basis, so having a dedicated and expensive on-site processing infrastructure is not sustainable for large-scale deployments typical of IoT deployments. In the off-site processing topology, the sensor node is responsible for the collection and framing of data that is eventually to be transmitted to another location for processing. Unlike the on-site processing topology, the off-site topology has a few dedicated high-processing enabled devices, which can be borrowed by multiple simpler sensor nodes to accomplish their tasks. At the same time, this arrangement keeps the costs of large-scale deployments extremely manageable [5]. In the off-site topology, the data from these sensor nodes (data generating sources) is transmitted either to a remote location (which can either be a server or a cloud) or to multiple processing nodes. Multiple nodes can come together to share their processing power in order to collaboratively process the data (which is important in case a feasible communication pathway or connection to a remote location cannot be established by a single node).</td><td></td></tr>
<tr><td>6</td><td>What is virtualization? How is it useful for end users? Explain the types of virtualizations</td><td></td></tr>
<tr><td>Sol</td><td>

2. Virtualization

- The technique of sharing a single resource among multiple end users is known as *"Virtualization"*. It is the key concept of cloud computing.

- In the virtualization process, a physical resource is logically distributed among multiple users. However, a user realizes that the resource is unlimited and is dedicatedly provided to him/her.

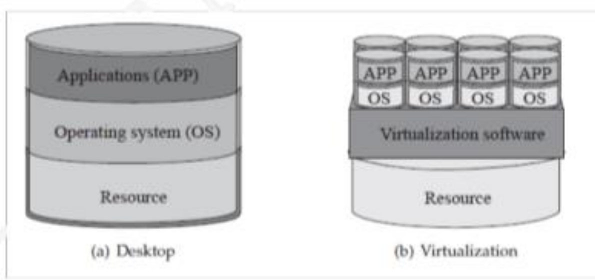


Figure 2.1 Traditional desktop versus virtualization

- Figure 2.2(a) represents a traditional desktop, where an application (App) is running on top of an OS, and resources are utilized only for that particular application.

</td><td>2</td></tr>
</table>

2.1 Advantages of virtualization: There are TWO main entities in a cloud computing architecture: *(1) End users and (2) Cloud Service providers (CSPs).* Both are benefited in several aspects through the process of virtualization. The

(1). Advantages of virtualization for End Users: They are as follows

(a) Variety *(b) Availability*
(c) Portability *(d) Elasticity*

(a) Variety:

– It enables various types of applications based on the requirements.
– It enables end users to access applications, hardware, or software virtually from a variety of devices and networks, regardless of their operating system (OS).

4

(b) Availability:

– Virtualization creates a logical separation of the resources of multiple entities without any intervention from end users.
– It makes available a considerable amount of resources as per user requirements.
– The end users feel that there are unlimited resources present dedicatedly for him/her.

(c) Portability:

– Ability to transfer applications and data between cloud computing environments.
– It enables migration between public and private clouds.
– Portability signifies the availability of cloud computing services from anywhere in the world, at any instant in time.
– It allows individuals to obtain and reuse their data for their purposes across different services.
– It allows them to move, copy or transfer personal data easily from one environment to another in a safe and secure without affecting its usability.
– This has been made possible by such as Google Drive.

(d) Elasticity:

– Elasticity refers to automatically increasing or decreasing cloud resources.
– It automatically adapts to match resources with demand as closely as possible, in real-time.
– An end user can scale up or scale-down resources like processing, memory, and storage resources to meet changing demands.

2.2 Types of virtualization*:** Based on the requirements of the users, virtualization is categorized into FOUR types as shown in Figure below

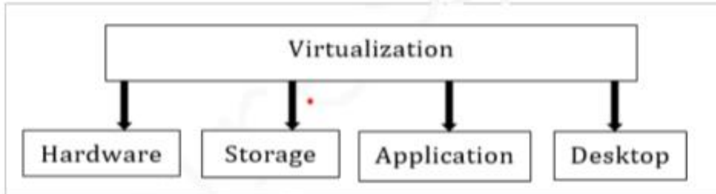


Figure 2.3 Types of virtualization

(i) Hardware Virtualization: Sharing of hardware resources among multiple users. For example, a single processor appears as many different processors in a cloud computing architecture. Different operating systems can be installed in these processors and each of them can work as a stand-alone machine. It uses a virtual machine manager (VMM) called a hypervisor to provide abstracted hardware to multiple guest operating systems, which can then share the physical hardware resources more efficiently.

(ii). Storage Virtualization: In this virtualization, the storage space from different devices is accumulated virtually, and seems like a single storage location. Through storage virtualization, a user's documents or files exist in different locations in a distributed fashion. However, the users are under the impression that they have a single dedicated storage space provided to them.

(iii). Application Virtualization: Application virtualization software allows users to access and use an application from a separate computer from the one on which the application is installed. For example, a single application is stored at the cloud end, but, as per requirement, a user can use the application on his/her local computer without ever actually installing the application.

(iv). Desktop Virtualization: This type of virtualization allows a user to access and utilize the services of a desktop that resides in the cloud. The users can use the desktop from their local desktop.

4