Q 1) Explain IoT Cloud based data collection, storage and Computing Services using Nimbits

- Nimbits is a platform as a service (PaaS) used to develop software and hardware solutions that seamlessly connect to the cloud and each other.
- Nimbits server runs on powerful cloud platforms like Google App Engine to the smallest Raspberry Pi device.
- Nimbits server is a web portal and API designed to

- Provides time-stamping or geo-stamping on incoming data.
- Store and process that time and location stamped data over cloud (pushing the data over cloud and store them in a data point)
- Provide filtering to incoming data from noise, add important changes to it and then generate trigger events and alerts based on rules and then sending them in real time over internet.
- It provides rule engine for connecting sensors, persons and software to cloud.

The basic services offered by Nimbits are

- Nimbits clients can plot charts and graphs of real time collected data over the internet.

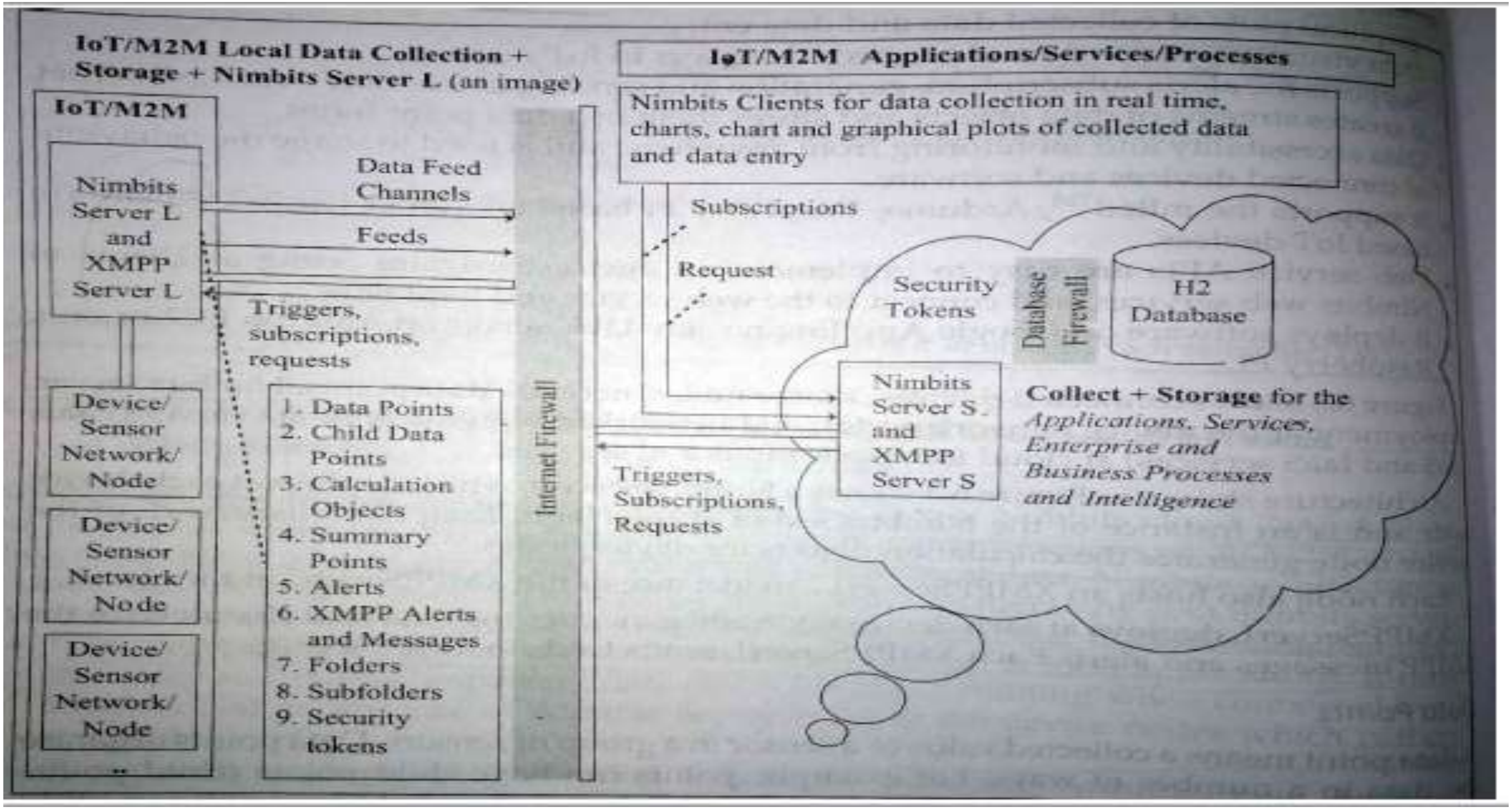
- It supports many format like text, JSON or XML values into the cloud.

- It provides edge computing locally on devices and nodes**.**

- It supports multiprogramming languages, M2M communication and hardware platform of IoT devices like mbed, Arduino, raspberry Pi based etc.

- Nimbits data points can relay data between the software systems or hardware devices like Arduino, using cloud as backend.

- It provides data logging services, access and data monitoring from anywhere.


IoT/M2M Local Data Collection + Storage + Nimbits Server L (an image)
IoT/M2M
Nimbits Server L and XMPP Server L
Device/ Sensor Network/ Node
Device/ Sensor Network/ Node
Device/ Sensor Network/ Node
..
Data Feed Channels
Feeds
Triggers, subscriptions, requests
1. Data Points
2. Child Data Points
3. Calculation Objects
4. Summary Points
5. Alerts
6. XMPP Alerts and Messages
7. Folders
8. Subfolders
9. Security tokens
Internet Firewall
IoT/M2M Applications/Services/Processes
Nimbits Clients for data collection in real time, charts, chart and graphical plots of collected data and data entry
Subscriptions
Request
Security Tokens
Database Firewall
H2 Database
Nimbits Server S and XMPP Server S
Collect + Storage for the Applications, Services, Enterprise and Business Processes and Intelligence
Triggers, Subscriptions, Requests

The Figure shows the 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.

Working

The Nimbits serverL : It is an instance of Nimbits serverS at the cloud

- The Nimbits serverL is deployed at each device node.

- The Nimbits servers first store, then filter and clean the data from noise, add some important changes, provides time-stamping or geo-stamping to it and send events, alerts (like email alerts or push notifications) by using rules an calculations called as **data feeds channels** (A data feed contains latest updates of current information like events, alerts etc) using XMPP messages.
- These data feeds (notifications) are sent over data feed channel using XMPP (Extensible Messaging and Presence Protocol) messages

- This pushing of the **alerts and messages down quickly or repeatedly is called as “Jabbing”.**

- Hence server relay that data up to a website or to a some mobile device

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 (mean subpoints), points can be in folders, the folders can go as deep as like a tree (Tree means a folder having a subfolders and so on).
- Any type of document can be organised with the points.

Child Data Points

- It means the **subpoints stored in the folders** (like subfolder)

Data Feed

- **A data feed** is a special point (an ongoing stream of structured data) that **provides users with updates of current information (like events, alerts etc) from one or more sources**.
- Feed consists of time/ geo-stamped data points, streams and alerts.
- Feed for messages and alerts generated on application of rules for filtering and calculation.

Data feed channels

- **User create data feed channels which shows system events, messages, data alerts also called as feed.**
- The user can subscribe to the data point of other users also and configure the subscriptions to send messages to the feed.

Calculations objects

- A user create calculations objects for a point.
- **The user can apply many formulas for a single data point e.g. for temperature sensor on formula is for increase in last value other formula is for increase over a normal value and then organise the objects in a tree**.

Summary points

- A user can create **summary point which can compute averages, minimum, maximum. Standard deviations, variance and sums** of another point on a specific time interval basis.

Folders, subfolders

- Data points can be stored in folders and subfolder

Security tokens

- Used for authentication and authorization.

Alerts and XMPP alerts and messages

- The **Nimbits servers filter and clean the data from noise,** get important data and then relay that data up to a website or to a some mobile device

- Hence **when the data is passed form sensors to Nimbits server, it adds rules and the formulas calculations to the incoming data and also execute triggers** like email alerts or push notifications **which are sent or broadcasted over XMPP (**Extensible Messaging and Presence Protocol**)** and those triggers generated data i.e. **the data feeds that further cascade with other data feeds** and this cascading result could be an alert which indicates that some relevant changes are happening within this massive system.

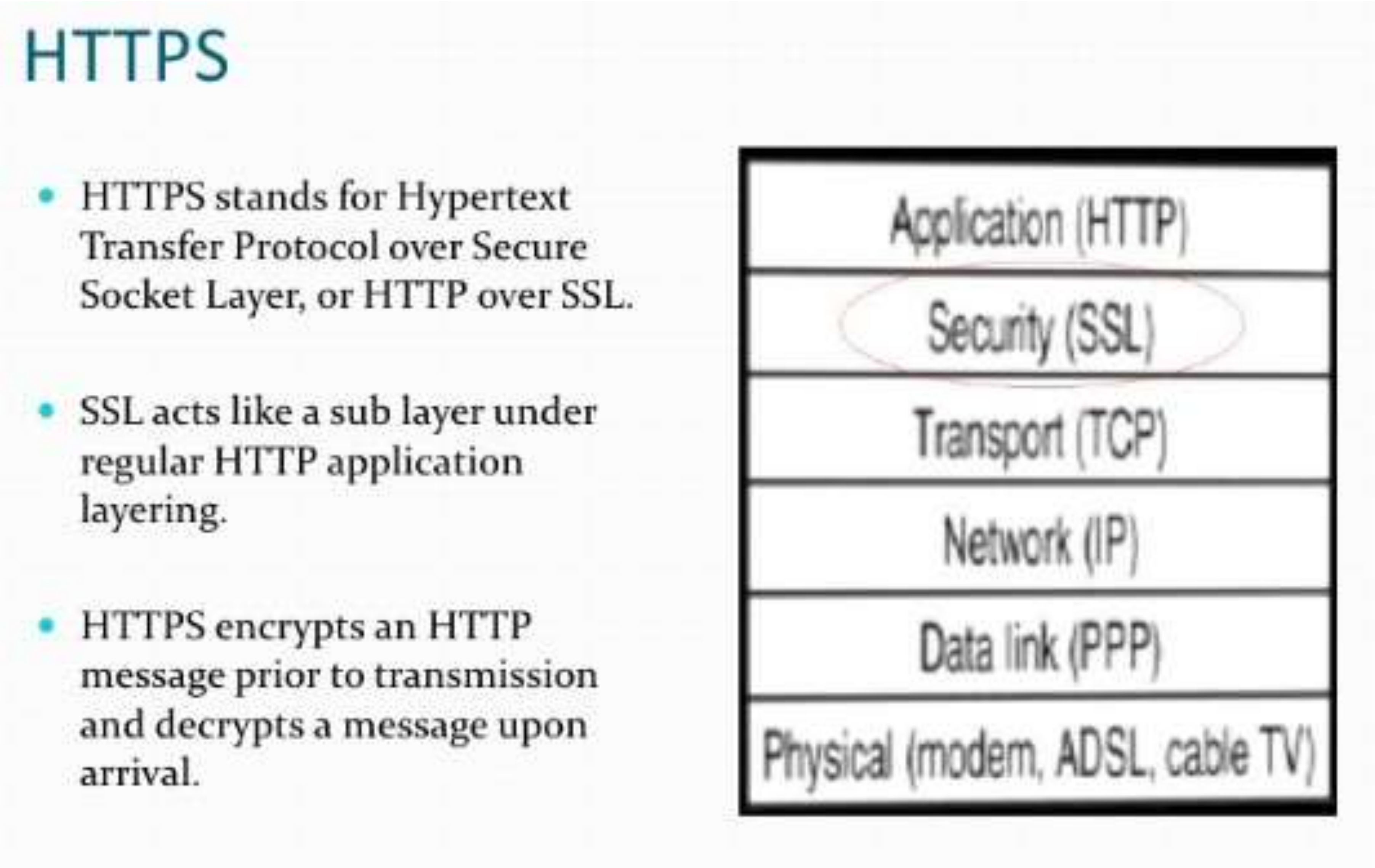
- **The filter item is called as “ah” for XMPP alerts and use custom Jabber IDs.**

Q2 a) Explain about wireless sensor network with neat figure. Explain the challenges for WSN.

- A wireless sensor network (WSN) is a wireless network consisting of **spatially dispersed autonomous devices called sensor nodes, which uses sensors to record and monitor physical or environmental conditions**, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

- These sensors communicate to each other wirelessly like in adhoc manner and **then the collected data is transmitted to central location called gateway** or sink or base station

Q2 b) Explain about HTTPS



HTTPS

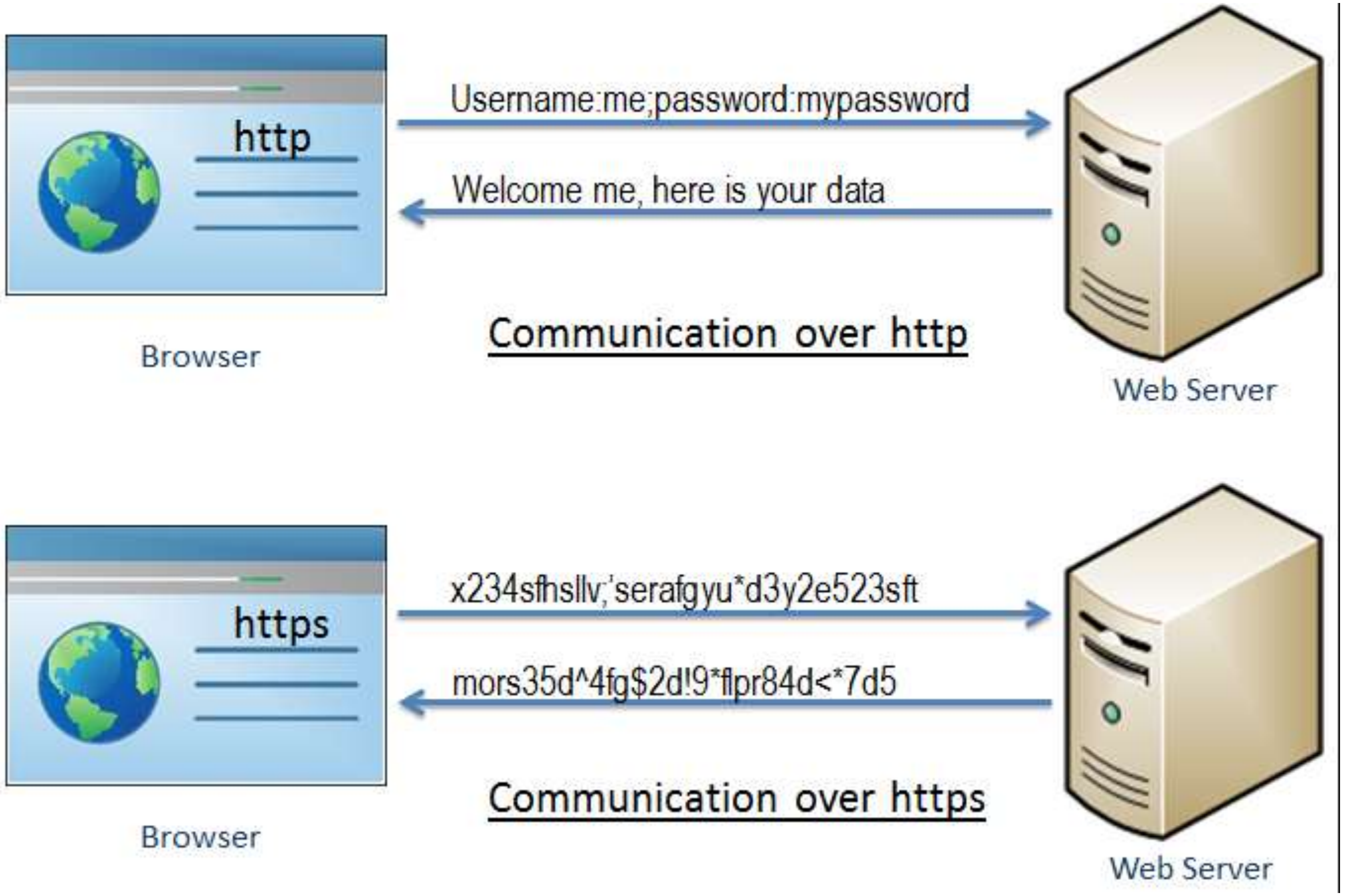
- HTTPS stands for Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL.

- SSL acts like a sub layer under regular HTTP application layering.

- HTTPS encrypts an HTTP message prior to transmission and decrypts a message upon arrival.

Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

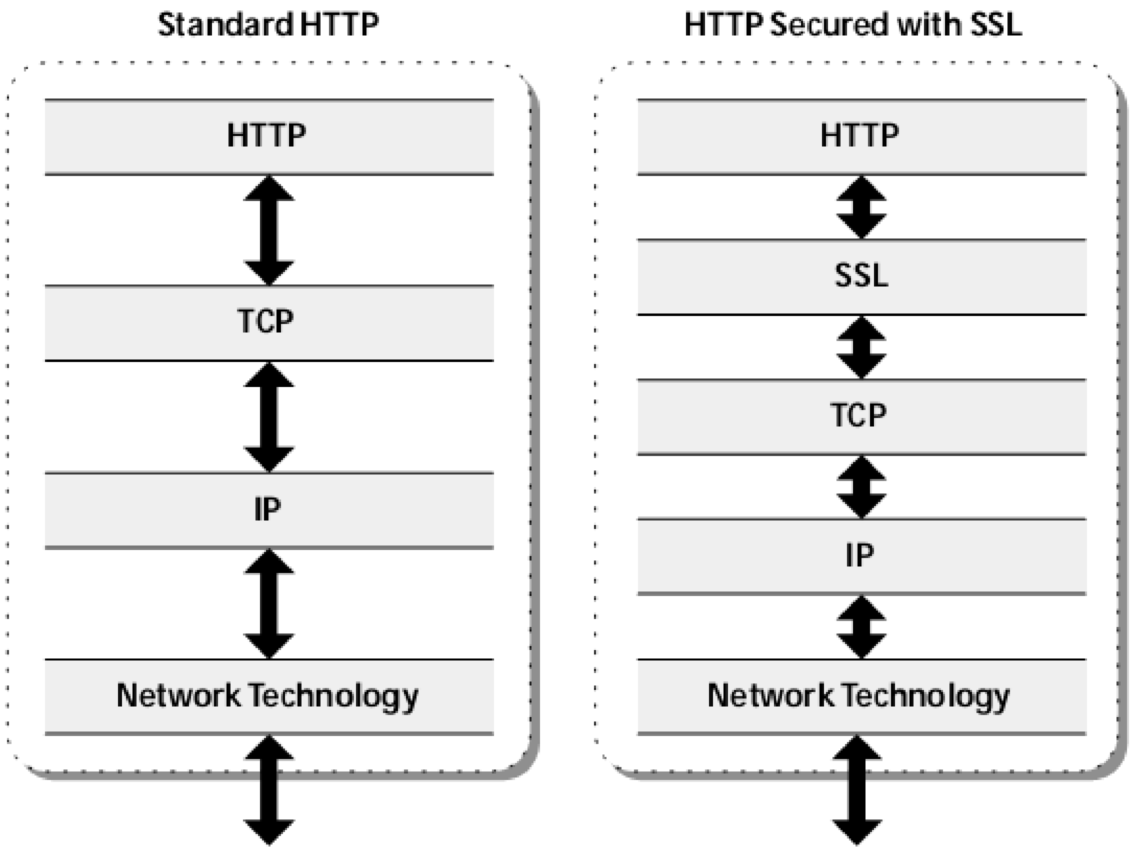
The following figure illustrates the difference between communication over http and https:


http
Username:me;password:mypassword
Welcome me, here is your data
Communication over http
Browser
Web Server
https
x234sfhsllv;'serafgyu*d3y2e523sft
mors35d^4fg$2d!9*flpr84d<*7d5
Communication over https
Browser
Web Server

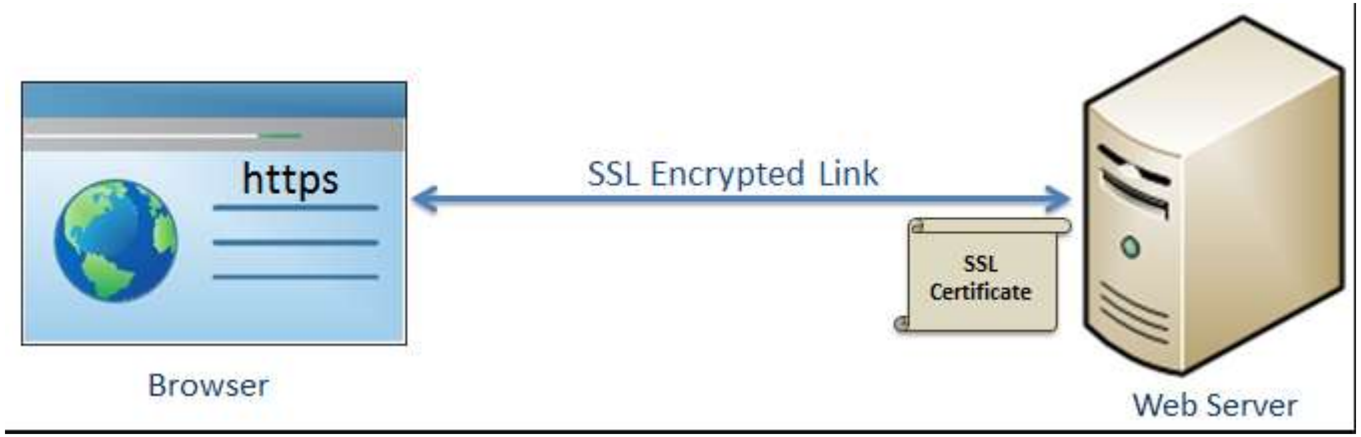
http transfers data between the browser and the web server in the hypertext format, whereas https transfers data in the encrypted format hence https prevents hackers from reading and modifying the data during the transfer between the browser and the web server.

HTTPS

Figure 4.10 ▶
The SSL protocol inserts itself between an application like HTTP and the TCP transport layer. TCP sees SSL as just another application, and HTTP communicates with SSL much the same as it does with TCP.



Secure Socket Layer (SSL)



- ➢ SSL is the standard security technology for establishing an encrypted link between the two systems.
- ➢ These can be browser to server, server to server or client to server.
- ➢ Basically, SSL ensures that the data transfer between the two systems remains encrypted and private.
- ➢ SSL establishes an encrypted link using an SSL certificate which is also known as a digital certificate.
- ➢ HTTPS=HTTP+SSL

Q 3) What is embedded operating system. Write a note on TinyOS with architecture and nesC.

❖ **Embedded operating systems (RTOS)**:

✓ It is a type of operating system that is embedded means specifically configured and programmed for a certain hardware configuration to do specific tasks.

✓ Hardware that uses embedded operating systems are lightweight, compact and operate with a limited number of resources.

✓ Embedded operating systems are also known as real-time operating systems (RTOS).

OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS

- ❑ To support the node operation, it is important to have open-source operating systems designed specifically for WSNs.

- ❑ Such operating systems uses component based architecture that minimize code size as as required by the memory constraints scenario in sensor networks.

❖ TinyOS:

- ✓ **TinyOS is an open-source, flexible and Application-Specific, embedded system operating system for wireless sensor networks.**

- ✓ A wide community uses Tiny OS in simulation to develop and test various algorithms and protocols running on the nodes.

- ✓ WSN consists of a large number of tiny and low-power nodes, each of which executes simultaneous reactive programs that must work with strict memory and power constraints. TinyOS meets these challenges.

❖ TinyOS and nesC

- ✓ **The kernel of TinyOS is developed in nesC.**

- ✓ **nesC:** It is a version of C programming language called network and embedded system programming.

❖TinyOS design models:

1) Component – based model

- ➢ Simple functions are in cooperated in components with clean interface, keeping code size minimum.
- ➢ Complex functions can be implemented by composing components.

2) **Event – based model**

Interact with the outside by events (no command shell)

There are two kinds of events for TinyOS:

a) **External events:** clock events and message events

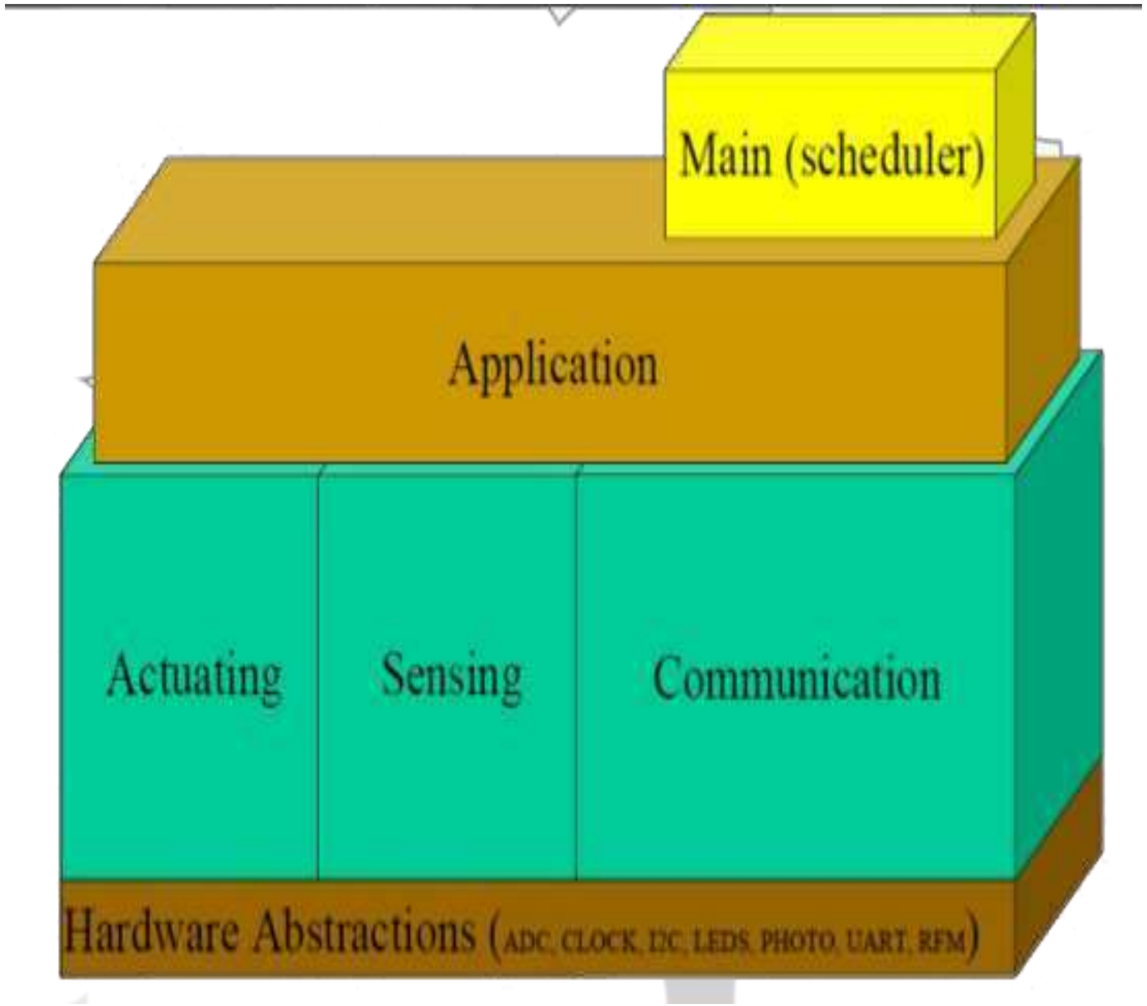
b) **Internal events**: triggered by external events.

Tiny OS Architecture

TinyOS's architecture consists of:

- Scheduler:

- A set of components:


Main (scheduler)
Application
Actuating
Sensing
Communication
Hardware Abstractions (ADC, CLOCK, I2C, LEDS, PHOTO, UART, RFM)

➢ OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

Component

- A component is similar to an object in object-based programming languages:

- A component is made up of:
  - A frame: contains the state information
  - Command handlers : requests
  - Event handlers: Interrupt/ trigger arriving from external.
  - Tasks: The work to be done.

  - They are structured hierarchically with lower level components near to hardware and higher level components closer to application.

  - *Component library: It includes network protocols, distributed services, sensor drivers, and data acquisition tools*

➢ OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

▪Commands:

➢Commands are the task are passed from high level to low level components.

➢When the command arrives from higher component to the lower component, the lower component has to perform the issued task.

▪Events:

➢Event are passed from low level to high level components.

➢When an event (trigger) arrives from the lower component then event handler leaves the information about that event in the frame so that task can be executed later.

➢ OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

❖ TinyOS:

✓ Salient features of TinyOS are

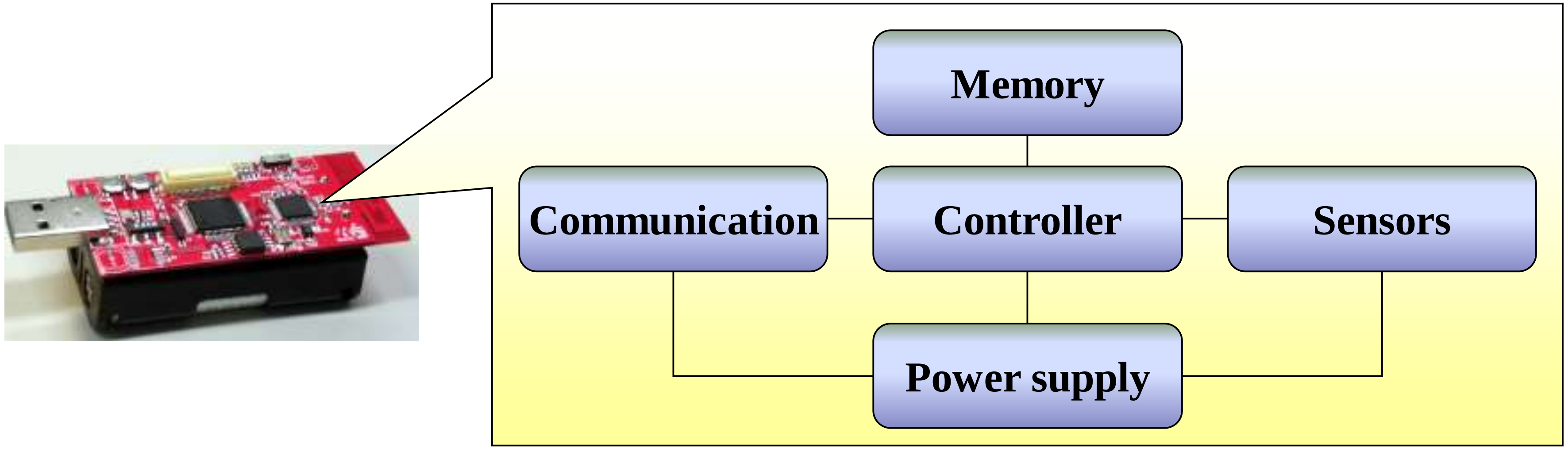
- Has Event-based concurrency model
- Component-based architecture.
- TinyOS’s component library includes network protocols, distributed services, sensor drivers, and data acquisition tools.
- TinyOS’s event-driven execution model.
- T**he kernel of TinyOS is developed in nesC.**

- **nesC** : It is a version of C programming language called **network and embedded system programming.**

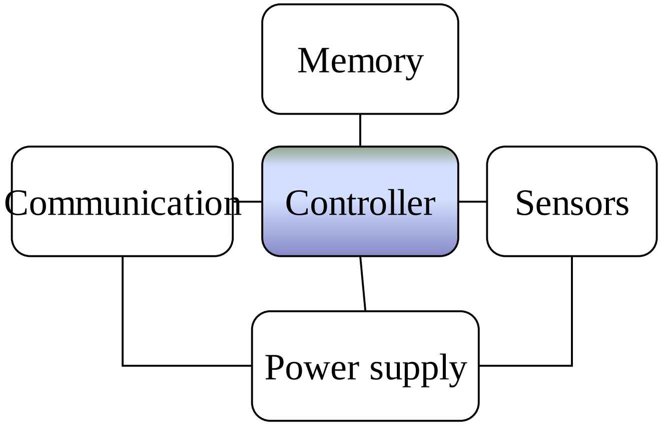
- ✓ Basically between components commands and events are the only way of interaction.
- ✓ Commands and interfaces form an interface between two components.
- ✓ nesC language defines the interface types that define the command and event that belong together.
- ✓ It also introduces the split phase programming (e.g. in first phase sending the command and in second phase executing the task etc.)

Q 4) Explain with neat figure the single node architecture with necessary hardware components

- The main architecture of sensor node includes following components:
    - Controller module
    - Memory module
    - Communication module
    - Sensing modules
    - Power supply module



Controller



Main functionality

- It is core of a wireless sensor network.

- It collects data from the sensors, processes this data, decides when and where to send it, receives data from other sensor nodes and decides on actuator’s behavior.

- It is CPU of sensor node as it executes various programs ranging from time critical signal processing and communication protocols to application protocols.

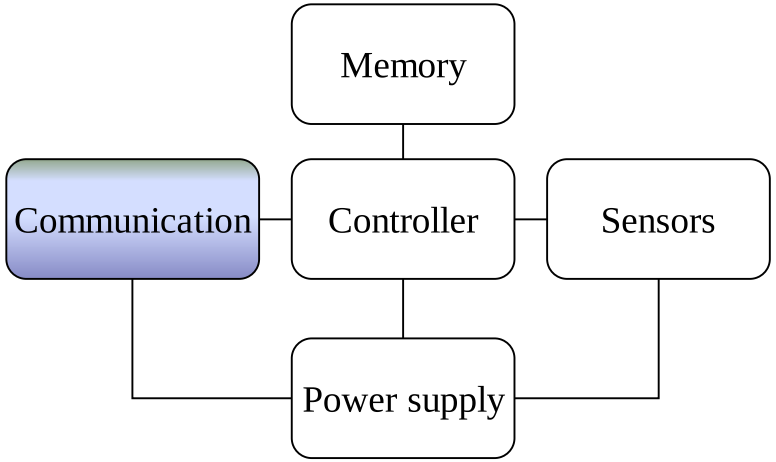
Controller

Microcontroller

Microcontroller

- General purpose processor.
- Optimized for embedded applications,
- Low power consumption by going to sleep state
- Flexible in connecting with the devices like sensors
- They have build in memory.

Communication module



- The communication module of a sensor node is called “Radio Transceiver”.

- The essentially tasks of transceiver is to “transmit” and “receive” data between a pair of nodes.

- Depends upon the
  - a) Choice of transmission medium
  - b) Transceivers

Choice of transmission medium

- Both wired and wireless communication can be used.
- Wired communication:

  It can be carried out by using field buses like LON, CAN etc.

- Wireless communication

✓ It can be radio frequencies, light, ultrasound etc

✓ It provides relatively high data rate and does not require the

✓ line of sight between sender and receiver.

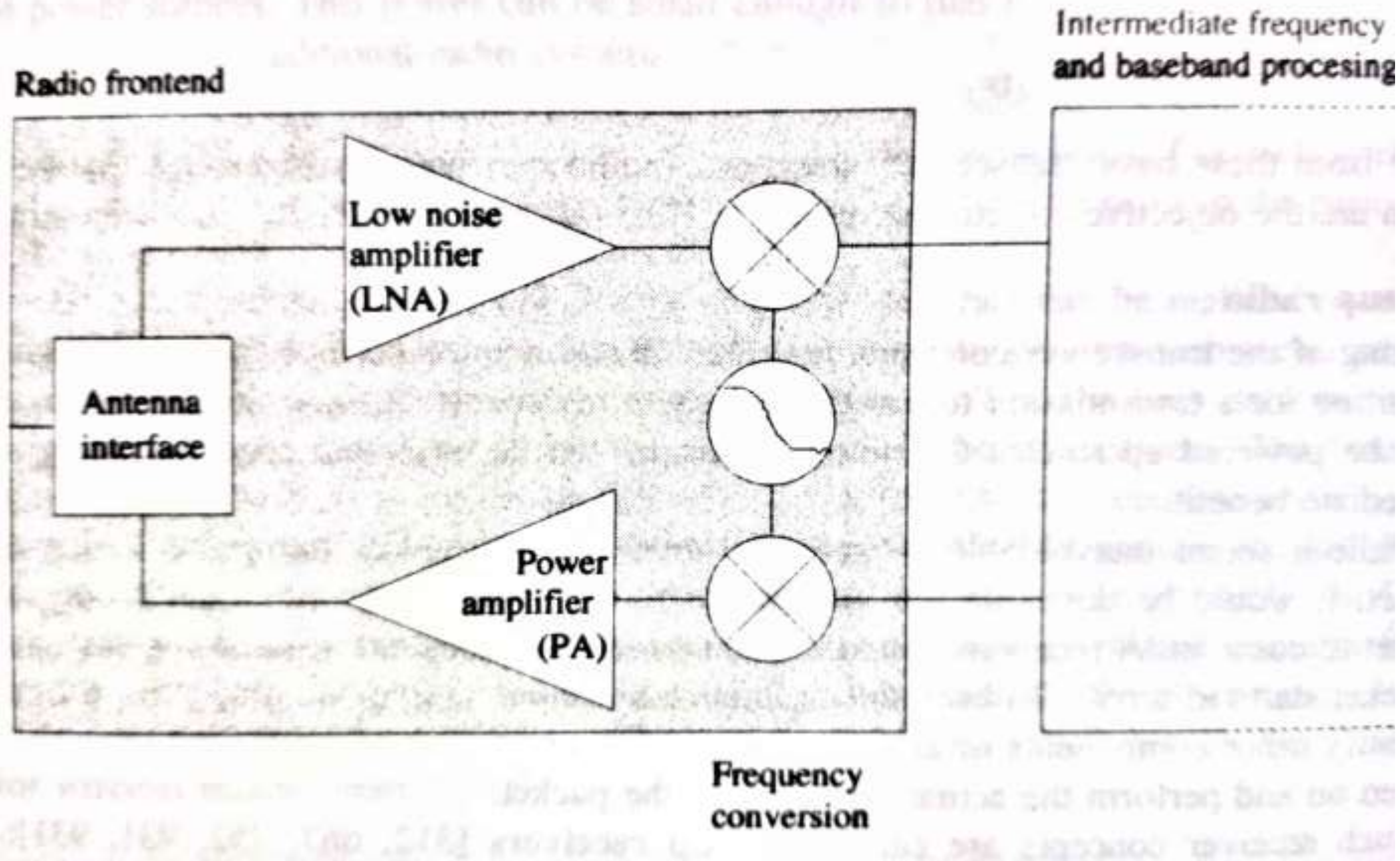
✓ It uses communication frequency between 433 MHz to 2.4

✓ GHz.

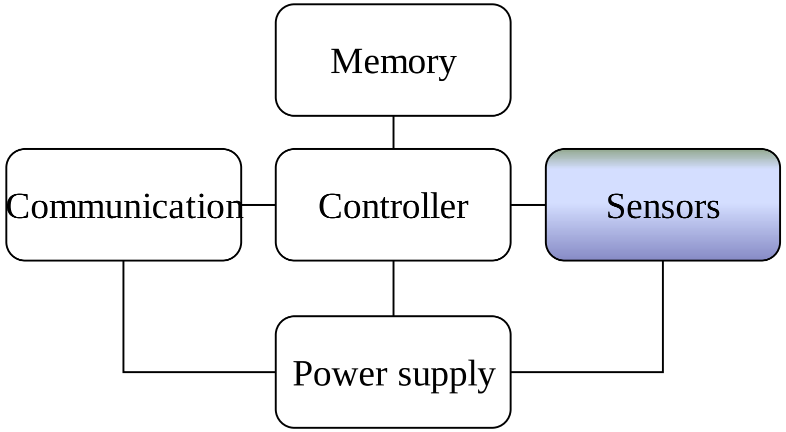
Transceivers

- Transceivers is the combination of transmitter and receiver.
- The main task is convert the bit stream coming from microcontroller to the radio waves.
- They can also perform modulation , demodulation, amplification and so on.

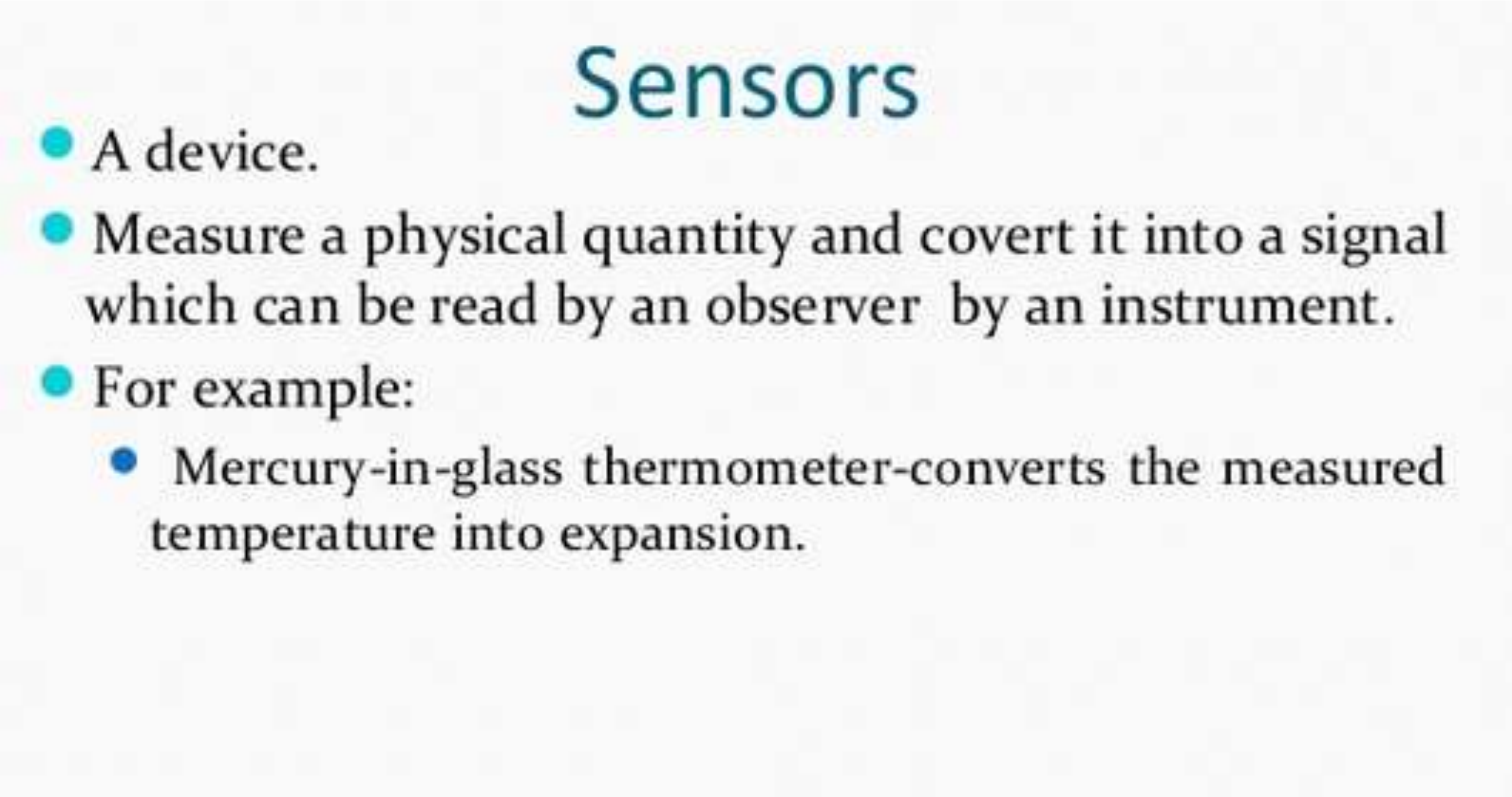
Transceiver structure



- Sensor’s main categories [1]
  - Passive vs. Active
  - Directional vs. Omidirectional



- Some sensor examples
  - Passive & Omnidirectional
    - light, thermometer, microphones, hygrometer, ...
  - Passive & Directional
    - electronic compass, gyroscope, …
  - Passive & Narrow-beam
    - CCD Camera, triple axis accelerometer, infar sensor …
  - Active sensors
    - Radar, Ultrasonic, …



Sensors

- A device.
- Measure a physical quantity and covert it into a signal which can be read by an observer by an instrument.
- For example:
  - Mercury-in-glass thermometer-converts the measured temperature into expansion.

Sensors :

Collect data from environment

- Main categories
  - Passive
  - Passive, narrow-beam
  - Active sensors

Passive Sensor

- Not directional
- They are self powered in the sense that they obtain energy from the environment.
- Examples: thermocouples, magnetic microphones, piezoelectric sensors, light, thermometer, microphones, hygrometer

Other name: self-generating sensors,

Passive, narrow-beam sensor

- Omnidirectional
- They are also self powered in the sense that they obtain energy from the environment
- Example: Camera

Active sensors

- It is a sensor that requires external power to operate.
- Examples: the carbon microphone, thermistors, strain gauges, capacitive and inductive sensors, Radar etc.
- Other name: parametric sensors (output is a function of a parameter - like resistance)

- Important parameter of sensors: <u>Area of coverage</u>

<u>Area of coverage:</u>

It defines the distance or the region between sensor and the object to be detected.

Actuator

A device or mechanism capable of performing a physical action for example motor, light bulb, LEDs etc

➢ Memory:

- ✓ Memory is required to store programs and intermediate data; usually, different types of memory are used in WSN for programs and data.

- ✓ **Random Access Memory (RAM)** to store intermediate sensor readings, packets from other nodes, and so on.

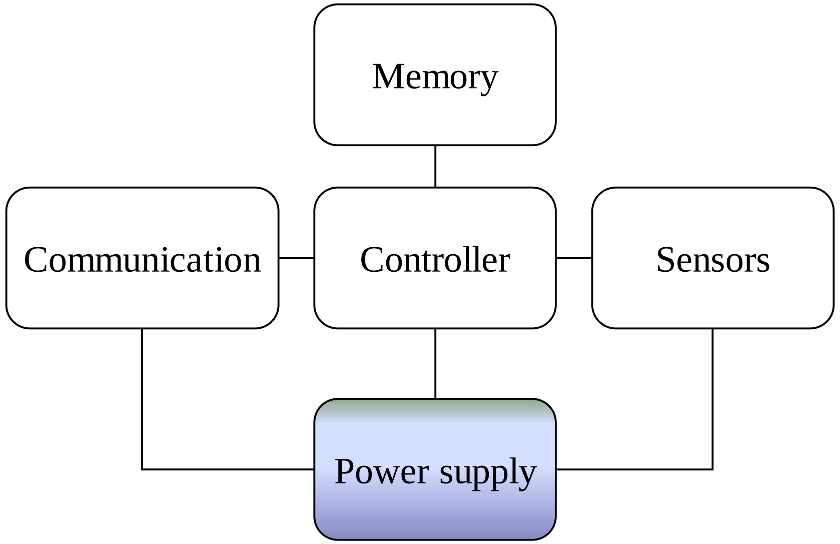
- ✓ RAM is fast, its main disadvantage is that it loses its content if power supply is interrupted.

- ✓ **Read-Only Memory (ROM)** Program code can be stored in Read-Only Memory (ROM) or in Electrically Erasable Programmable Read-Only Memory (EEPROM) or flash memory.

- ✓ **Flash memory** is similar to EEPROM but data can be erased or written in blocks instead of only a byte at a time. It can also serve as intermediate storage of data in case RAM is insufficient or the power supply of RAM should be shut down.

Main Components of a Sensor Node: Power supply module

- **Power supply module**
  - It should provides as much energy as possible
  - includes following requirements



Two Options

a) Through batteries

b) Energy scavenging

Wireless Sensor Network Architecture

Challenges for WSN

Quality of service:

- **QOS** is the description or measurement of the overall performance of a service.
- There are several aspects to measure quality of service such as packet loss, bit rate, throughput, bandwidth, transmission delay, availability, jitter, etc.
- QOS parameters varies according to the application.

Challenges for WSN

Fault tolerance:

- Due to depletion of energy node can fail. In case of node failures network should cope up as soon as possible.
- Fault tolerance is the ability of the network to sustain the problems like sensor node failures and network should function without any interruption.

Challenges for WSN

Scalability:

- Scalability measures the density of the sensor nodes. In some applications, tens of thousands of sensors might be deployed. At any time numbers of nodes can be increased or decreased.

- Performance of network should not be degraded as number of nodes increases.

Challenges for WSN

Production costs:

- The cost of a single node is very important to justify the overall cost of the networks.
- The cost of a sensor node depends upon the application and functionalities.
- Wireless sensor nodes are very small and inexpensive devices due to advanced technologies.

Challenges for WSN

Lifetime:

- Lifetime of a network is a very important figure of merit in WSN.
- Most of the nodes are battery operated and have limited power resources. Therefore power sources like solar cells must be available on a sensor node. Or lifetime of node should be infinite.

Challenges for WSN

Programmability:

- Nodes must be programmable and their programming must be changeable during the operation when the new task become important.

Challenges for WSN

Maintainability:

- Here the nodes must adapt itself according to the changing environments externally as well as internally.
- Survivability in harsh environments

Challenges for WSN

Transmission media:

- In a sensor network, communicating nodes are linked by a wireless medium. To enable global operation, the chosen transmission medium must be available worldwide.
- It can be radio, infrared, optical media etc

Challenges for WSN

Power consumption

Most power is consumed in

- Sensing
- Communication
- Data processing

Design Challenges

- **Heterogeneity**
  - The devices deployed maybe of various types and need to collaborate with each other.
- **Distributed Processing**
  - The algorithms need to be centralized as the processing is carried out on different nodes.
- **Low Bandwidth Communication**
  - The data should be transferred efficiently between sensors.

Q5 a) Write a note on Optimization goals and figure of merit.

- The main challenge for a network is how to optimize a network.
- Optimization and figures of merit depend upon certain parameters like:

  - ❖ Quality of service
  - ❖ Energy efficiency
  - ❖ Scalability
  - ❖ Robustness

- Quality of service involves:

A) Low level networking device observable attributes like: Bandwidth, delay, jitter, packet loss rate

B) **High level, user observable also called as subjective attributes like**: Quality of voice communication or video transmission.

In WSNs, the high level attribute depends upon the application.

❖*Quality of service:* Some generic possibilities are

- ✓ **Event detection/reporting probability :** Means the event that actually happened is detected or not or reported or not to the information sink.

- ✓ **Event classification error-** If events are not only to be detected but also to be classified, the error in classification must be small

- ✓ **Event detection delay -**It is the delay between detecting an event and reporting it to any/all interested sinks

❖*Quality of service:* Some generic possibilities are

- ✓ **Missing reports** -In applications that require periodic reporting, the probability of undelivered reports should be small.

- ✓ **Approximation accuracy-** It defines what is the average/maximum absolute or relative error with respect to the actual function.

- ✓ **Tracking accuracy** Tracking applications must not miss an object to be tracked, the reported position should be as close to the real position as possible, and the error should be small.

❖ *Energy efficiency:*

The Energy efficiency of the WSN can be increased by considering various aspects.

1) **Energy per correctly received bit:**

It defines the average energy consumed in transporting and receiving one bit of information, after considering all possible intermediate hops from source to destination.

2) **Energy per reported event:**

It defines the average energy consumed in reporting one event. Since same event can be reported from various sources. Hence redundant information can be reduced.

3) Delay/Energy trade-off:

In case of reporting of urgent events a huge amount of energy is consumed. Here a trade-off (balance) between Delay/Energy is an important aspect.

4) Network lifetime:

It is the time for which network is operational.
Possible definitions are:

a) Time to first node death:.
b) Network half-life:
c) Time to partition:

a) Time to first node death: The time at which the first node runs out of energy or stop working.

b) Network half-life: The time at which 50% of the nodes runs out of energy or stop working.

c) Time to partition: The time at which network get divided into further networks or there is partition between source and sink.

a) Time to loss coverage:

- It is the time when the nodes stop observing or monitoring an spot in its deployment range.

- Also called as the time at which nodes lost its coverage in deployment range.

b) Time to failure of first event notification:

- It is the time when the unreachable part of the network stop reporting any events.

- It happens *when partition between source and sink has occurred.*

❖ *Scalability:*

- ✓ With WSN potentially consisting of thousands of nodes, the ability to maintain performance characteristics irrespective of the size of the network is referred to as scalability.
- ✓ The need for extreme scalability has direct consequences for the protocol design as the complexity will increase and can effect the performance.

❖ *Scalability:*

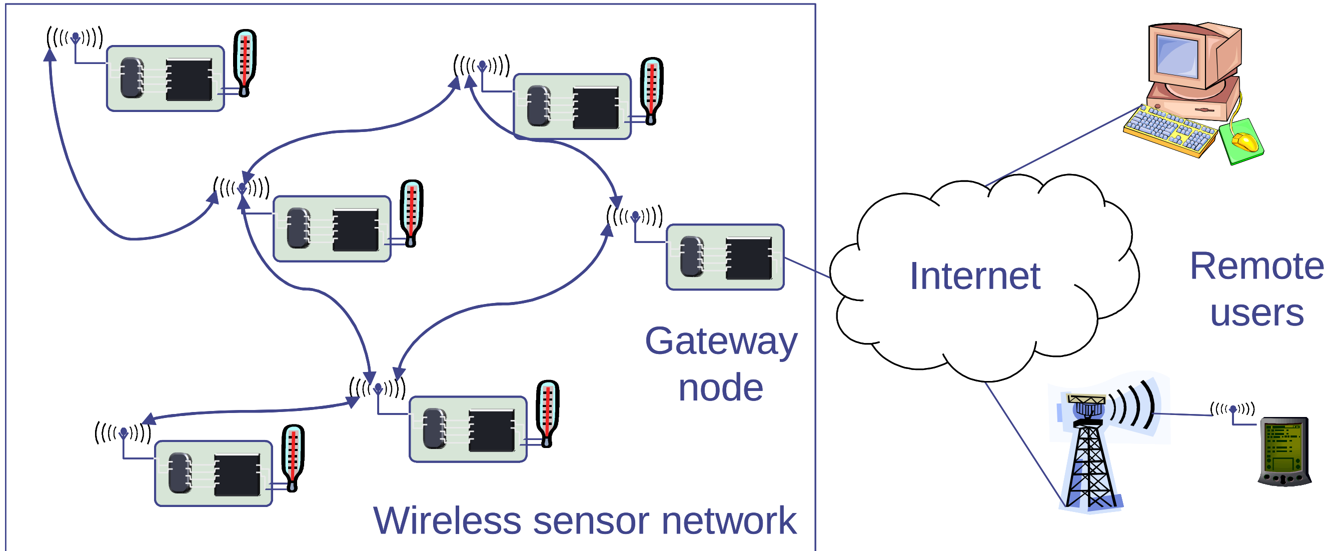
- ✓ With WSN potentially consisting of thousands of nodes.
- ✓ Architectures and protocols should implement appropriate scalability support rather than trying to be as scalable as possible
- ✓ Applications with a few dozen nodes might admit more-efficient solutions than applications with thousands of nodes

❖ *Robustness:*

- ✓ Wireless sensor networks should also exhibit an appropriate robustness
- ✓ They should not fail just because a limited number of nodes run out of energy, or because their environment changes and severs existing radio links between two nodes
- ✓ If possible, these failures have to be compensated by finding other routes.

Q 5 b)Write a short note on network gateway and tunneling.

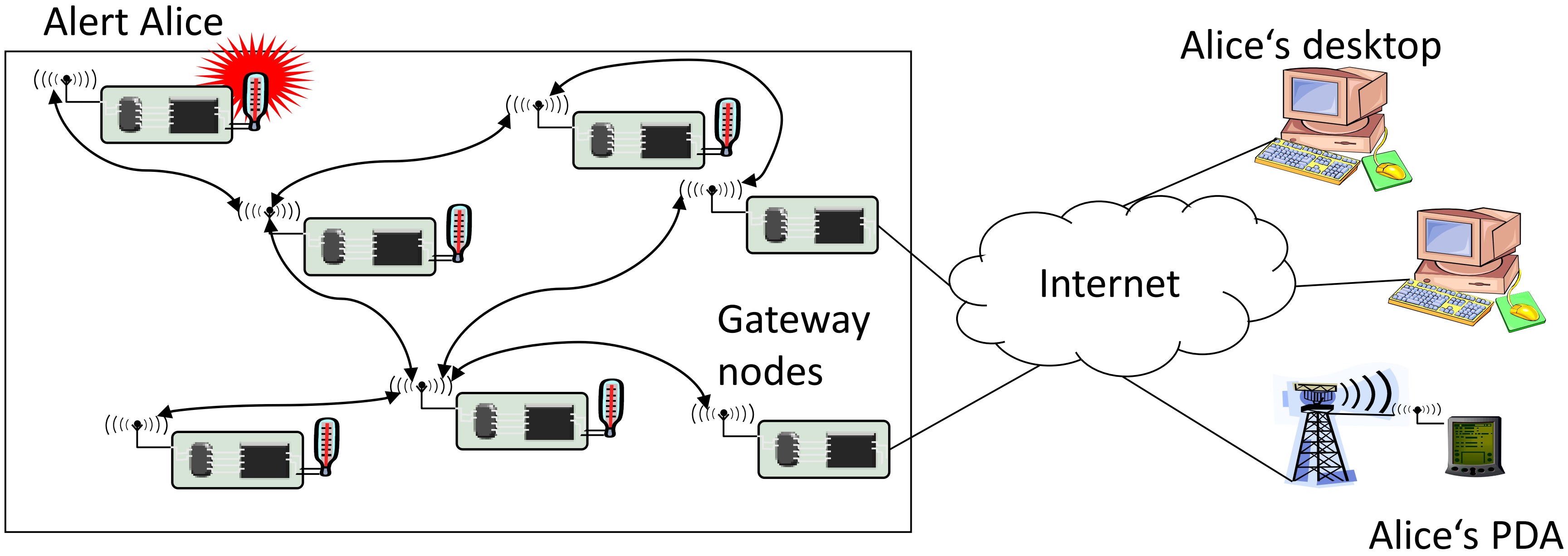
- Gateways allows the WSN to exchange the data with other devices like mobile phones.
- **Gateway node bridges a gap between WSN and other communication devices**
- Gateway is equipped with a radio transceiver or some standard wireless communication technique like IEEE 802.11.



Challenges in WSN to Internet communication

- Let an sensor node ‘ALICE ’wants to deliver an alarm message to some Internet host.

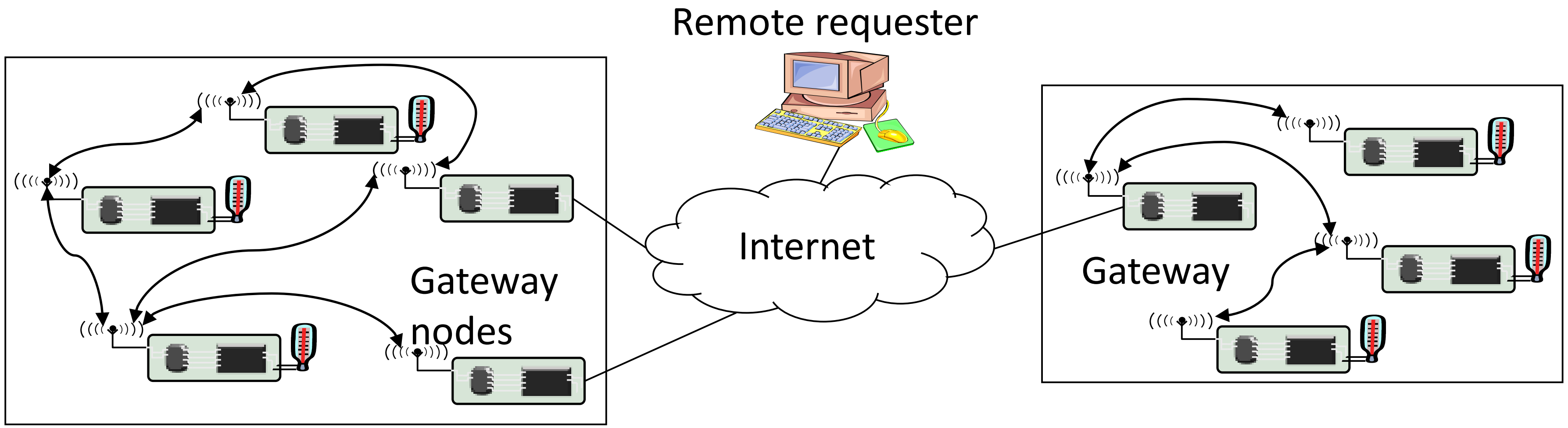
- But here occurs some issues like
  - ➢ How to handle the several gateways.
  - ➢ Choose “best” gateway (integrates routing & service discovery)
  - ➢ Finding the host IP address to which it has to be forwarded.



Internet to WSN communication

- Let internet based entity tries to access the services of WSN.

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



- There also occur many complexities like:

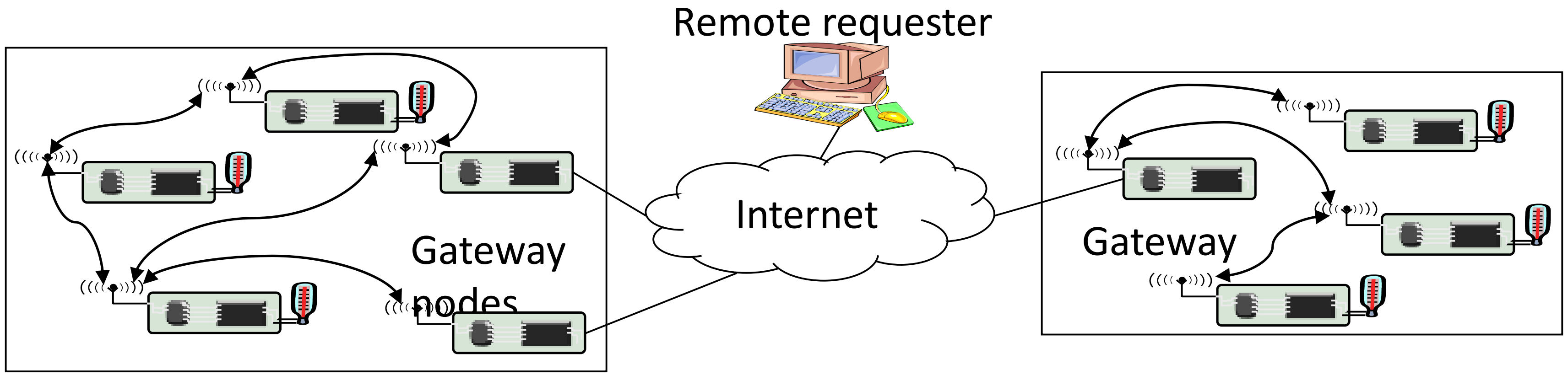
a) Gateway node is required.

b) How to find the right WSN in desired location is another problem.

c) Addressing of the right sensor in network.

d) How to translate from IP protocols to WSN protocols, semantics?

e) How to make WSN services accessible from standard web browser.

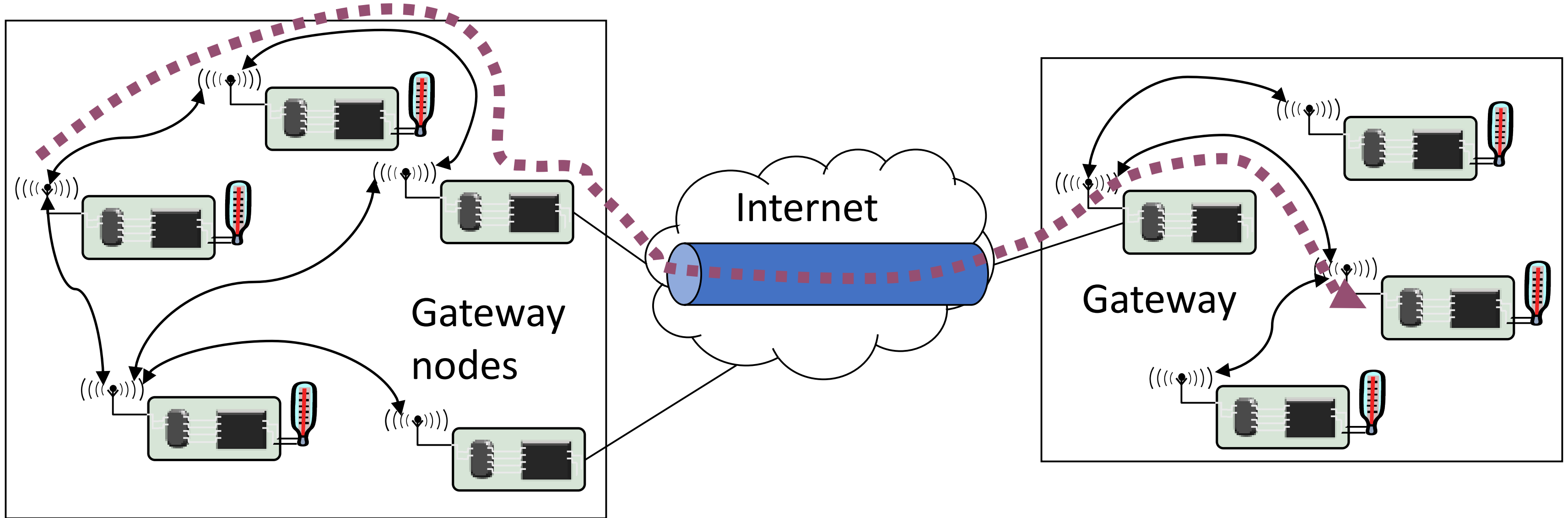


WSN tunneling

- Internet is used to “tunnel” WSN packets between two remote WSNs

✓ The gateways can also act as simple extensions of one WSN to another WSN.

✓ The idea is to build a larger virtual WSN, “tunneling” all protocol messages between two WSNs and simply using the Internet as a transport network



Q 6) Explain about the design principles of WSN and techniques used for In-network processing

Distributed control is used in WSNs for the following reasons:

✓Sensor nodes are prone to failure.
✓For better collection of data.
✓There is also no centralized body to allocate the resources and they have to be self organized.
✓To provide nodes with backup in case of failure of the central node.

In-network processing

- It is called as an In-network processing as the processing is done inside the sensor network close to the source.

- Here an intermediate proxy node is chosen which consolidates (process) the sensor data and route it to the sink node.

- It is energy efficient query processing paradigm as

a) Network is organized in an distributed fashion

b) Nodes not only passes the packets but also decides how to operate the network efficiently.

Techniques for In-network processing

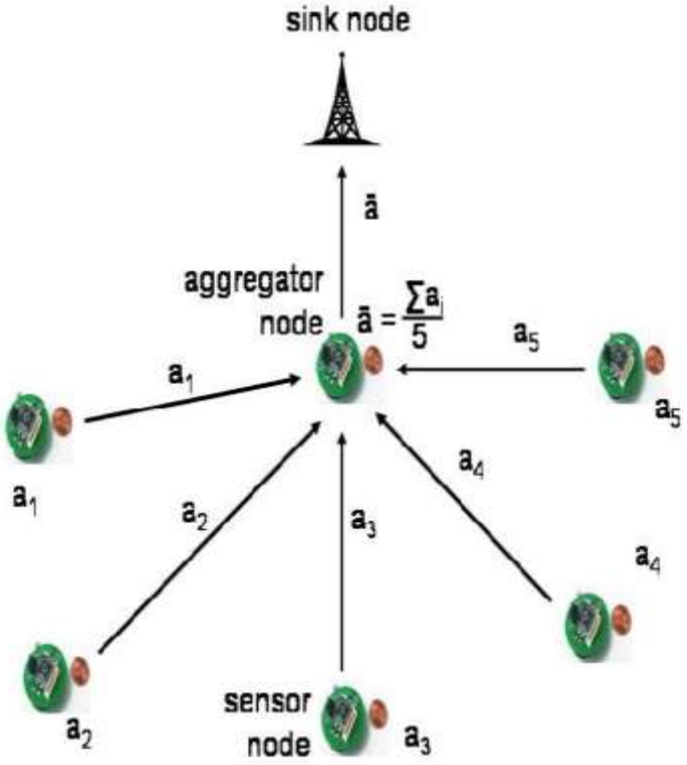
- Aggregation based techniques
- Approximation based techniques.
- Distributed source coding and distributed compression.
- Distributed and collaborative signal processing.
- Mobile code and agent based networking
- Geographic Adaptive Fidelity (GAF).

Aggregation based techniques

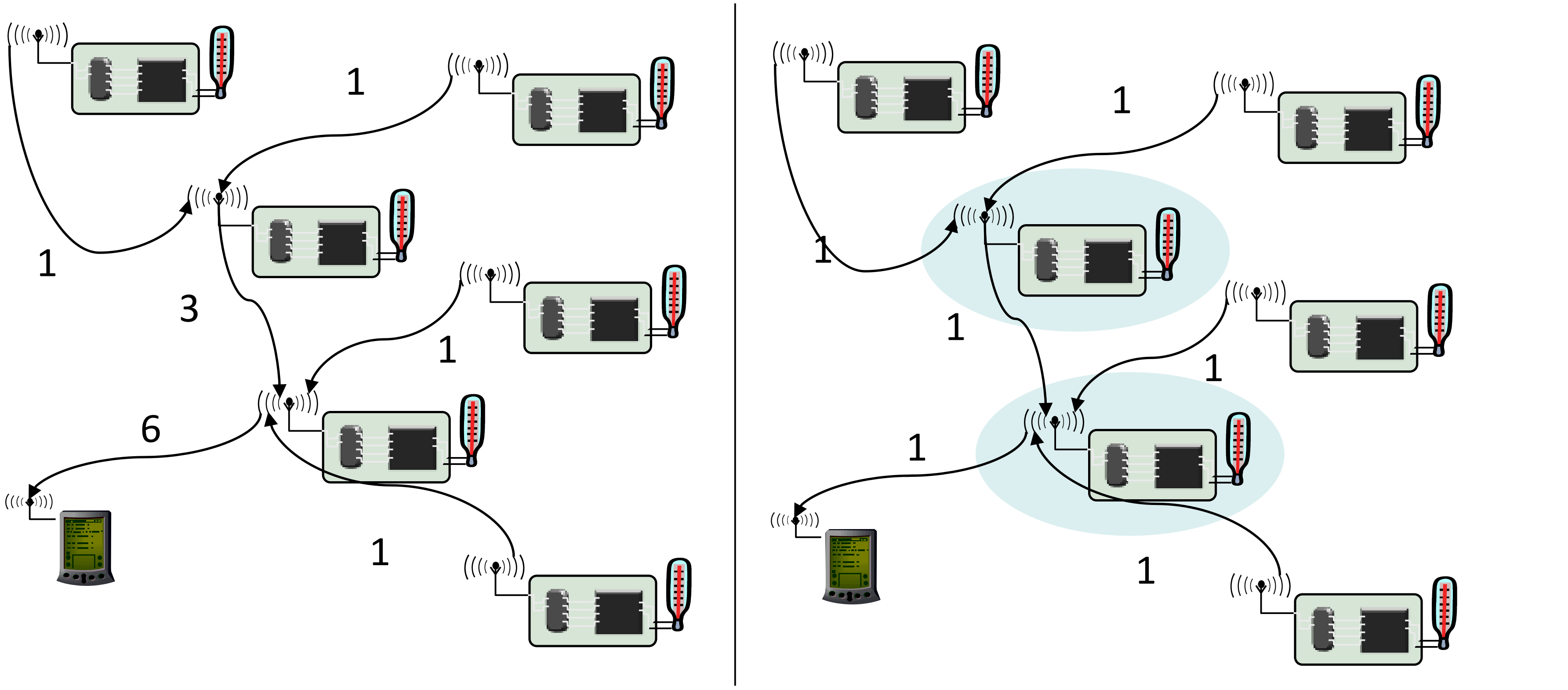
- Let the nodes are periodically measuring the data but it is not required to send all the collected data and it is sufficient to send that data where the average value has changed, or if there is a huge difference between maximum and minimum values.

- Hence aggregation condenses and removes the redundant information by reducing the number of transmitted bits (means not transferring all the information).

- Aggregation reduces the amount of network traffic and energy consumption on sensor nodes



In-network processing: Aggregation example



- In LHS total 13 messages are sent from source to sink
- but in RHS only 6 messages are sent using multihop.

Approximation based techniques

- It is data compression techniques focus on reducing the amount of data packets to be transmitted in-network when the accuracy of data collection is important.

- It defines what is the average/maximum absolute or relative error with respect to the actual function.

- Apart from common computing aggregates such as min, max, average, count, the median (most frequent data values) are also considered.

Distributed source coding and collaborative signal processing.

- Where network is distributed, each node has separate processer and working independently.

- CSP techniques are used for combining the data collected from different nodes within a particular active region.

- If the data collected from two nodes are correlated then data fusion is done.

- If the data collected from two nodes are uncorrelated then decision fusion is done i.e. exchange of likelihood values.

- Hence the measurement should be mixture of correlated and independent components.

Distributed source coding and collaborative signal processing.

The benefits of such an approach are:

(i) The energy efficiency of the overall network can increase significantly.

(ii) The Fast Fourier Transform algorithm is used when complex computation on data is to be done.

(iii) The energy and computational limitations of the individual sensors can be overcome

Mobile code / agent based networking

- ➢ To increase the efficiency of the system a small compact program called as mobile code or software agent is executed at local nodes which decide where to send the data next.

Geographic Adaptive Fidelity (GAF).

- It is one of the most popular location based routing protocol.
- This is it uses the nodes location information to transfer the data.
- It reduces the energy consumption of nodes and increase the network lifetime.
- It reduces the use of energy by turning off the radio of some nodes which share the same functionalities.

Examples of Adaptive fidelity

- Example event detection
  - When there is no event, only very rarely send short “all is well” messages
  - When event occurs, increase rate of message exchanges

- Example temperature
  - When temperature is in acceptable range, only send temperature values at low resolution.
  - When temperature becomes high, increase resolution and thus message length.

Data centric networking

- Address data, not nodes:

➢ It defines that data is the center of attention, not the identity of nodes, called as data centric approach.

Implementation options for data-centric networking

- a) Overlay networks & distributed hash tables (DHT)
- b) Publish/subscribe
- c) Databases

- Overlay networks & distributed hash tables (DHT)

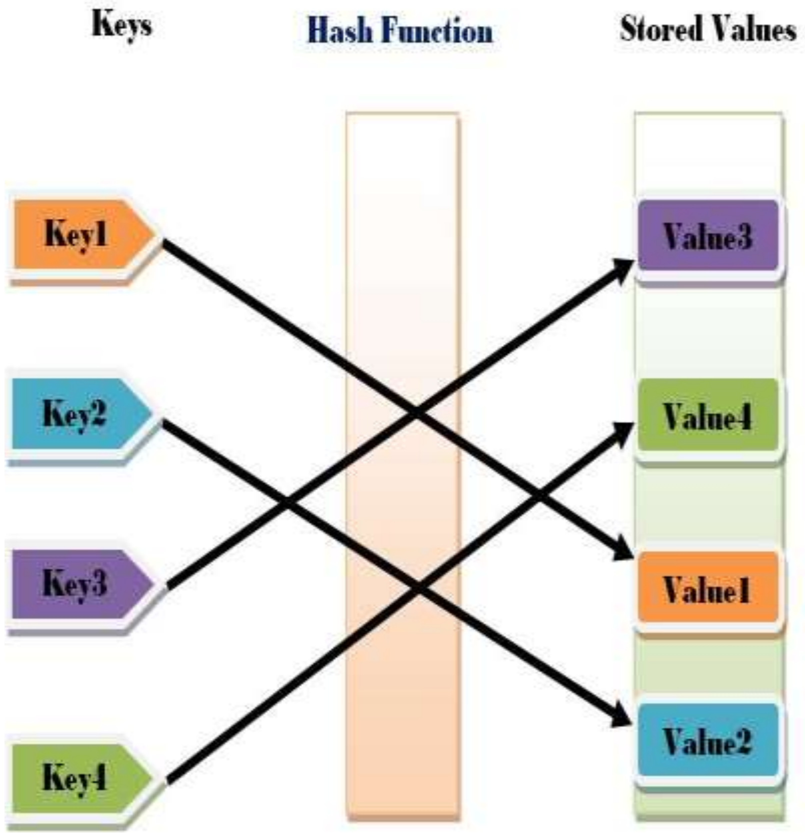
  - In many peer to peer applications the user is only interested in data not its source.

  - Hence, If the data is retrieved from an unknown source then an over lay network is formed by implementing distributed hash tables (DHT).

  - (key, value) pairs are stored in a DHT, where keys (hash) are unique identifiers which map to particular

- values, which in turn can be
- anything like addresses,
- documents, arbitrary data etc.

  - The data is accessed by nodes using these keys.



- 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
  - Subscription and publication are decoupled in time and in identity.

- Databases

  - WSN can be considered as dynamic databases (SQL databases).

  - The sensors are considered as virtual tables to which some relational operators can be applied.

  - Output of sensors can be seen as infinitely-long logical tables.

  - Columns consists of attributes defined in the network like sensor readings, node_id, location, time_stamps, user defined attributes

- For example extract the average readings from all the sensors having temperature greater than 10.

- Select nodeId, timestamp, temp, light
- From   sensors
- Where   light > 10

Further design principles

- Exploit location information
  - ➢ For some applications location information of sensor crucial information.
  - ➢ This simplifies the design and operation of communication protocols.

- Exploit activity patterns
  - ➢ Once an event has happened, it can be detected by a number of sensors, breaking into frenzy activity called as event shower effect.
  - ➢ So protocols should be designed in manner to handle such burst of traffic.

Further design principles

- Exploit heterogeneity

- Nodes can be heterogenous by

a) Construction: Nodes can be of different types in the network

b) Evolution: Some nodes had to perform more tasks and have less energy left; some nodes received more solar energy than others.

Component based protocol stack and cross layer optimization.

- Cross-layer optimization allows the communication between layers by permitting one layer to access the data of another layer to exchange information and enable interaction.

- Protocol component interact with each other in two ways:

- 1)Data packets are passed from one component to another which is processed by different protocols.

- 2)Exchange of cross-layer information i.e. layers can coordinate, interact and perform joint optimization of protocols.

Service interfaces of WSN

- It defines how one component in WSN to interacts and access the service of another component or application using service interface.

- ➢ Structuring application/protocol stack interfaces
- ➢ Requirements for WSN service interfaces

Structuring application/protocol stack interface.

➢ There are two options for interfacing an application to a protocol stack:

✓ Through deliberately designed service interface.

❑ With Service interface application can express the sensing tasks instead of specifying which value to read from sensor.

✓ Application can directly interface the hardware layer-

❑ Here it is easy to introduce the application specific codes to WSN.

❑ Programmer have a fine control over the protocols chosen for specific task.

Structuring application/protocol stack interface.

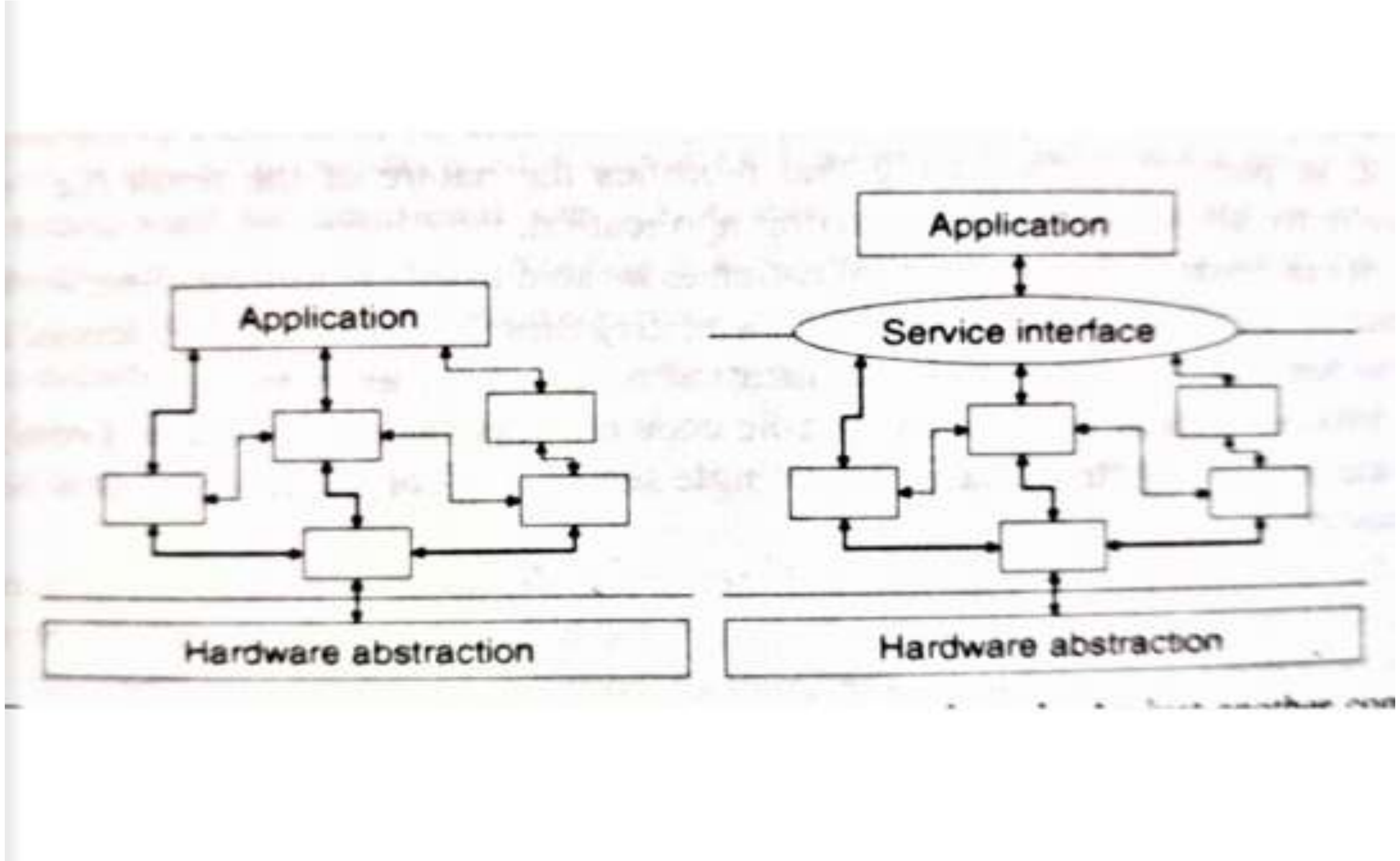


Figure:

Requirements for WSN service interfaces

- Support for simple request/response interactions
- Support for asynchronous event notification
- Different ways for identifying addressee of data
  - By location, by observed values, implicitly by some other form of group membership
  - By some semantically meaningful form – “room 123”
- Easy accessibility of in-network processing functions
  - Formulate complex events – events defined only by several nodes
- Allow to specify accuracy & timeliness requirements
- Access node/network status information (e.g., battery level)
- Security, management functionality…

Q7) Explain XMPP protocol with neat figure.

XMPP uses XML technology for real time communication includes *instant messaging (used in multiuser chat)*

Presence

Collaboration.

- The protocol is *used in constrained environment* for messaging.

- It is also used for publish-subscribe systems, signaling for VoIP, video, file transfer, gaming etc.

XMPP (Extensible Messaging and Presence Protocol)

X- Extensible:

XMPP is designed to be *extensible*, in has been designed to *grow and accommodate changes*.

M-Messaging:

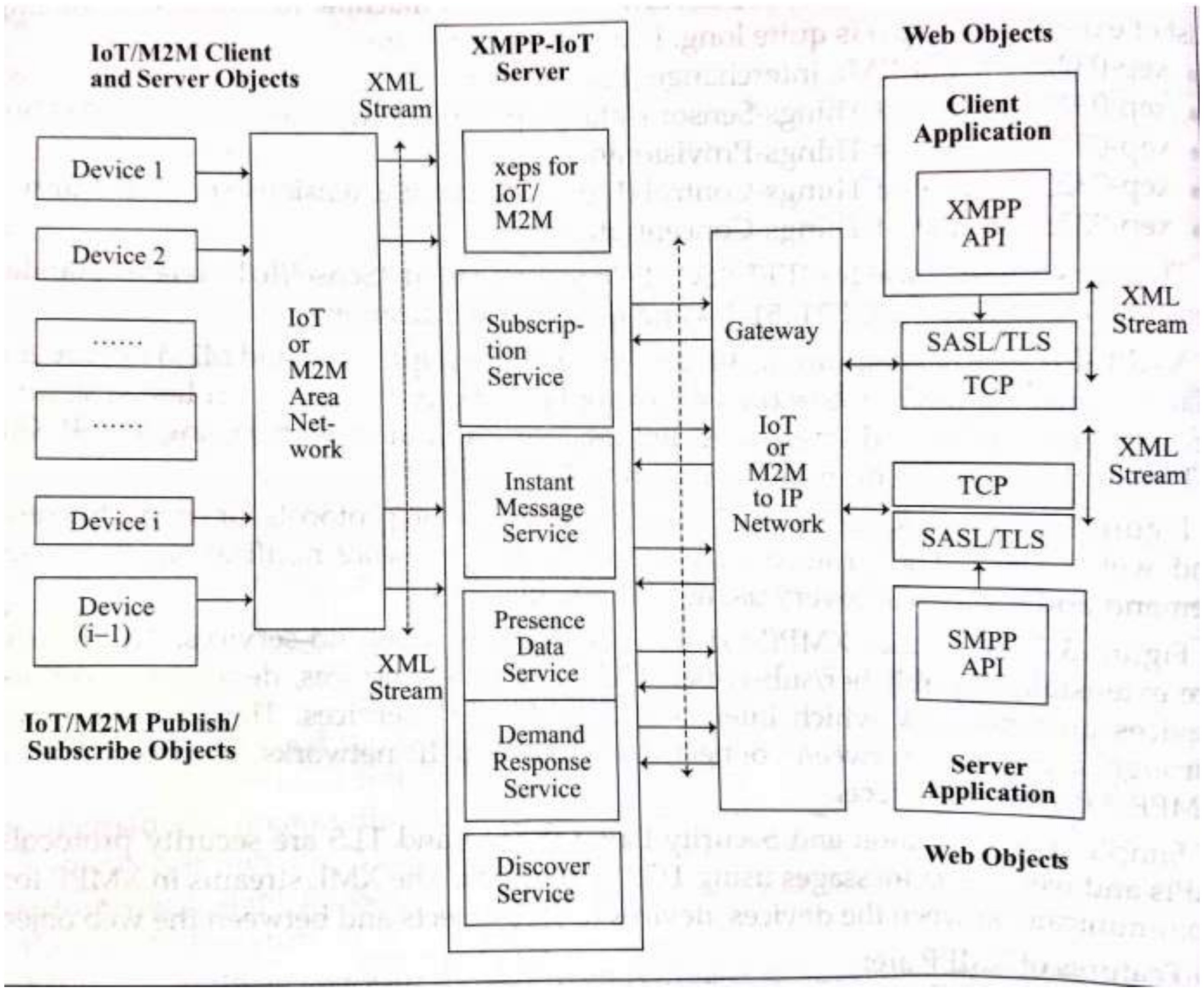
XMPP has been designed to send instant message.

P-Presence:

The presence indicator tells the server that you are online/offline/busy.

Protocol:

XMPP is a protocol; a set of standards to talk to each other. It is widely used across web but is unadvertised.



XMPP-IOT server/ XMPP M2M server is used for exchanging the messages between machines.
SASL (Simple Authentication and Security Layer)

XMPP

- XMPP –IOT server consists of xeps and services like.
- xeps are called as XMPP extension protocols. Examples of xeps are
  - xep-DataForms Format,
  - xep-XHTML-IM (instant messaging)
  - xep-Service Discovery
  - xep-MUC (Multi User Chat)
  - xep-Publish-Subscribe and Personal eventing Protocol
  - xep-File transfer
  - xep-Jingle for voice and video
- Services are like publish, subscribe, instant messaging, MUC etc
- Service discovery identifies the protocols, features supported by client and servers as well as the items associated with an entity, such as the list of rooms hosted at a multi-user chat service.

XMPP Working

- The IOT/M2M Client and Server objects communicate to XMPP-IoT server through the IoT/M2M Area Network.
- Server provides necessary services like publish subscribe, instant messaging, multiuser chat etc.
- Then XMPP server communicates to the XMPP APIs in the web objects through IOT gateway.
- SASL (Simple Authentication and Security Layer) and TLS (Transport layer security) are security protocols for APIs and WEB Objects messages using TCP/IP network.

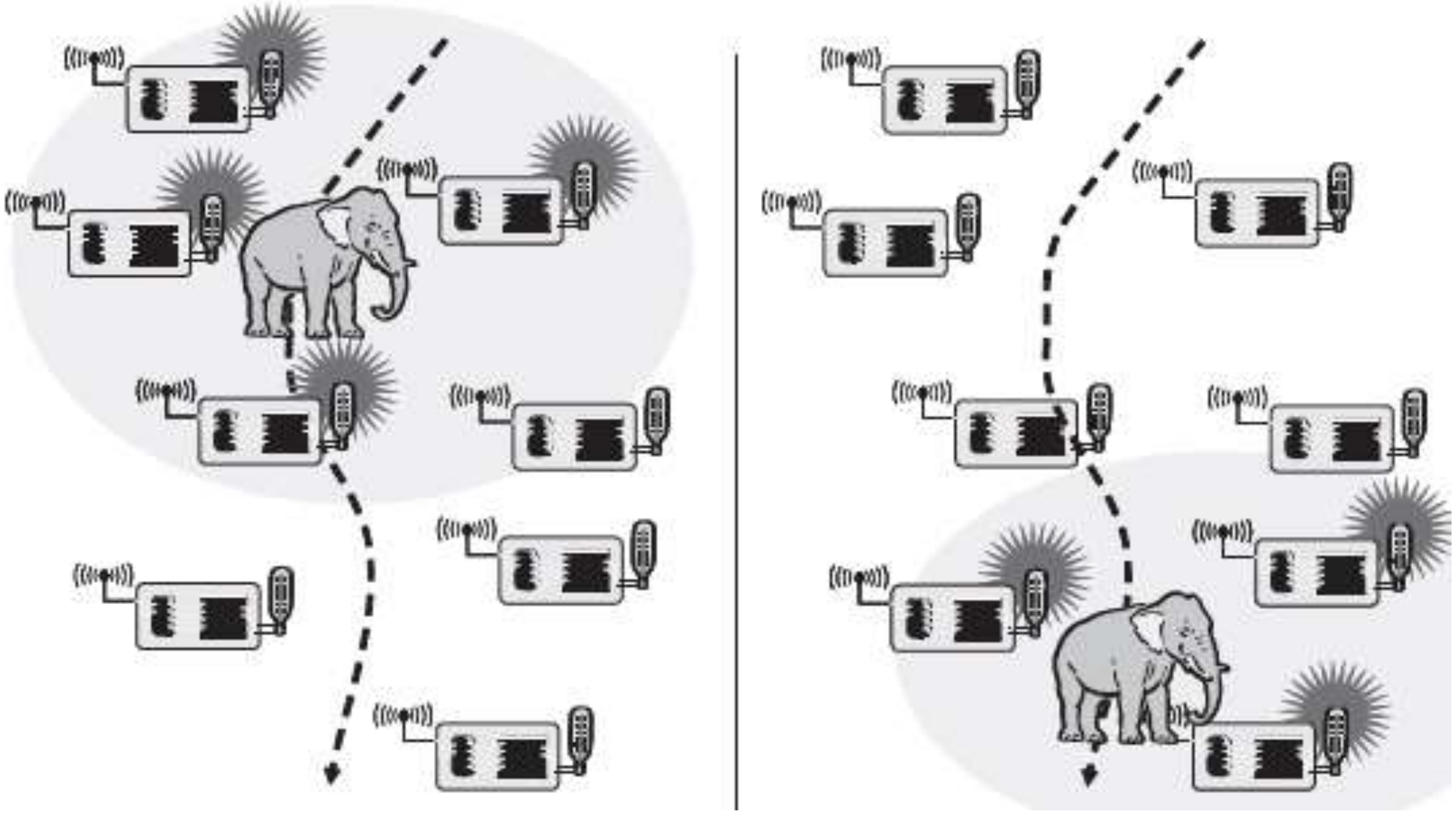
- **Q) 8 b Write a short note on the types of node mobility in WSN.**

- *Node mobility*: The wireless sensor nodes themselves can be mobile e.g in environmental control, livestock survillance.

- *Sink mobility*: The information sinks can be mobile but the mobility as an information sink is not the part of sensor network e.g. a human requests some information while walking in an building.

- *Event mobility*: The objects to be tracked can be mobile. Hence sensor will wake up when object is near, watch their activity and then go back to sleep.

- *Event mobility:* The objects to be tracked can be mobile.



Event mobility: An event(elephant) moves through the network along with the event source (dashed line)

- ✓ Wireless communication supports mobile participants.
- ✓ In WSN, mobility can appear in **three main forms**….

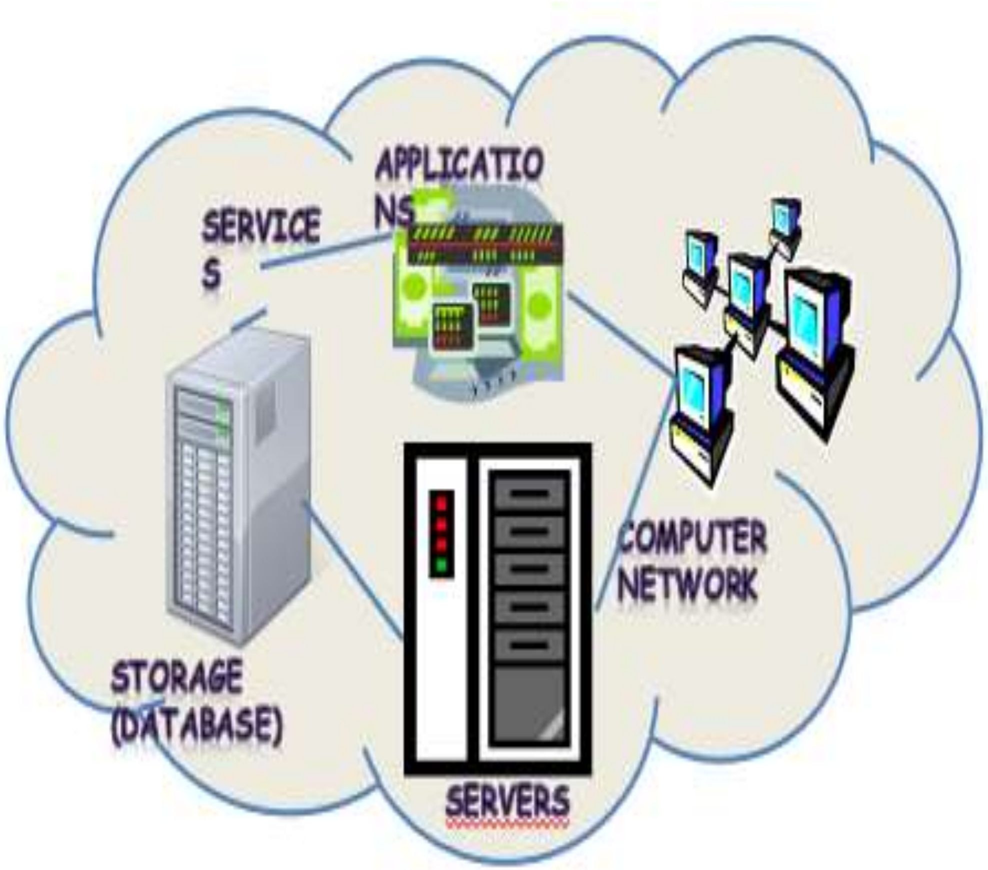
- ○ ***Node mobility***:
- ○ ***Sink mobility:***
- ○ ***Event mobility:***

Q 8 a) Explain cloud computing and explain the cloud service model with neat figure

- A collection of integrated and networked hardware, software, Internet infrastructure (called a platform) with a huge storage, computing capabilities.

- It provides hardware, software and networking services to clients on rent.

- These platforms hide the complexity and details of the underlying infrastructure from users and applications using API



Cloud computing Features and advantages

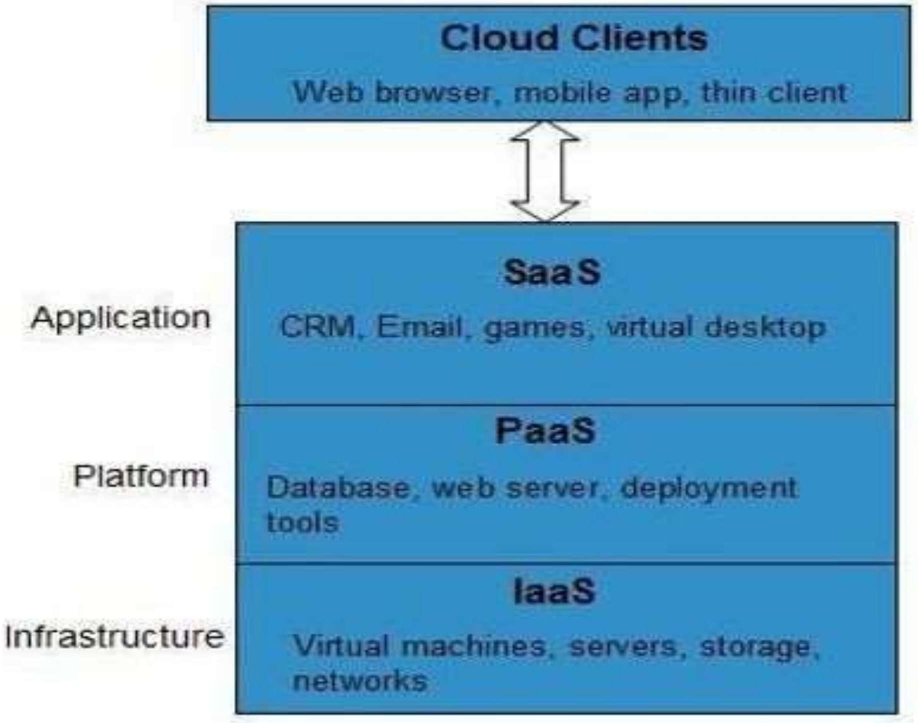
- Provide provision of **unlimited storage, computing servers, software delivery and servers** to the users on rent i.e. on demand.
- Provide **resource pooling** in multi-tenant model.
- Provide **broad network and remote access to heterogeneous users**, clients, systems and devices in virtualised environment.
- More **flexible and efficient allocation of resources**.
- Provides **elasticity, scalability, maintainability of resources**.
- It **lowers the cost** of IT infrastructure.
- Advanced **security and reliability**.
- Enables running **multiple operating system**.
- Always **Latest version availability**

Service Models

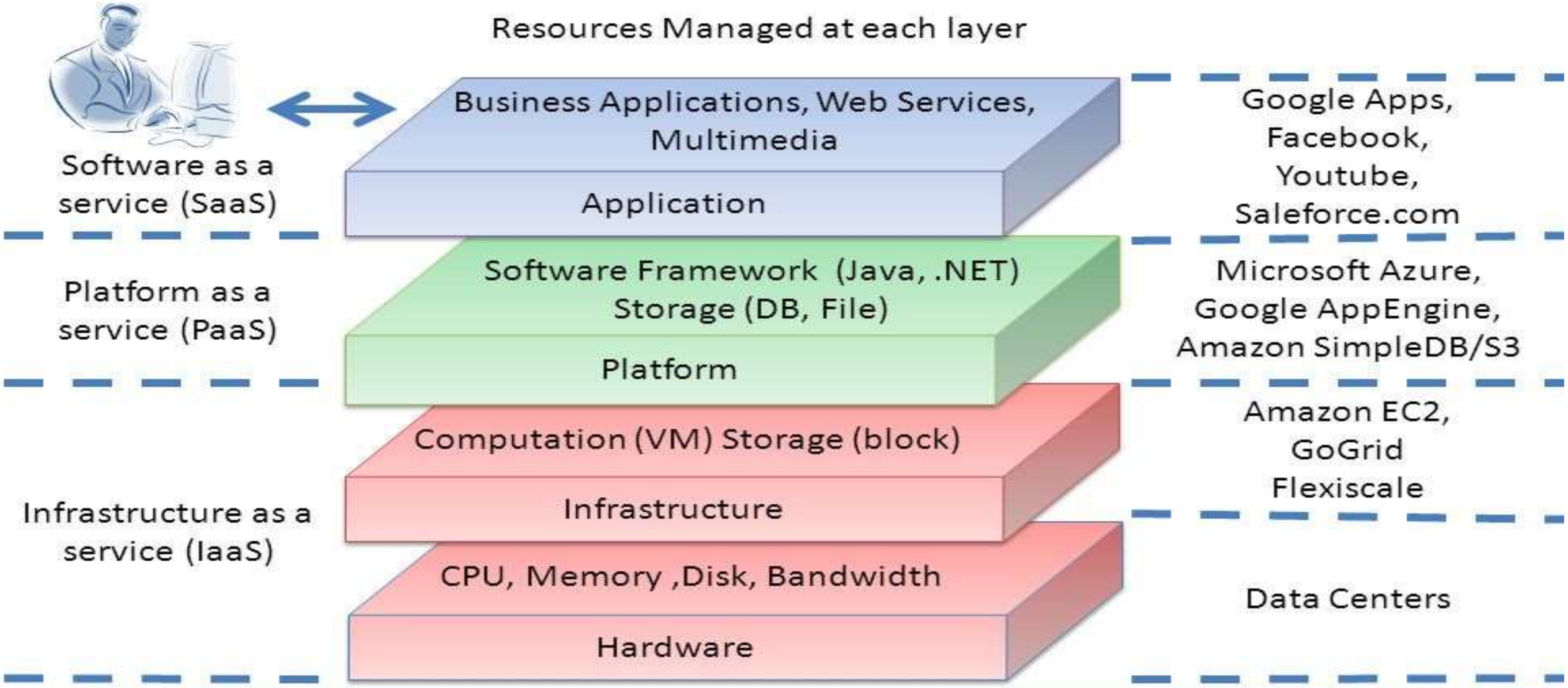
Cloud computing is not a single piece of technology, like a microchip or a cell phone. Rather, it's a system, primarily comprised of three services:

Service Models are the reference models on which the Cloud Computing is based. These can be categorized into three basic service models as listed below:

1. **Infrastructure as a Service (IaaS)**
2. **Platform as a Service (PaaS)**
3. **Software as a Service (SaaS)**


Cloud Clients
Web browser, mobile app, thin client
Application
SaaS
CRM, Email, games, virtual desktop
Platform
PaaS
Database, web server, deployment tools
Infrastructure
IaaS
Virtual machines, servers, storage, networks

Cloud Computing Layers



Resources Managed at each layer

Software as a service (SaaS)

Business Applications, Web Services, Multimedia

Application

Google Apps, Facebook, Youtube, Saleforce.com

Platform as a service (PaaS)

Software Framework (Java, .NET) Storage (DB, File)

Platform

Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3

Infrastructure as a service (IaaS)

Computation (VM) Storage (block)

Infrastructure

Amazon EC2, GoGrid Flexiscale

CPU, Memory ,Disk, Bandwidth

Hardware

Data Centers

Software as a Service (SaaS)

- **SaaS** is a method for delivering software applications over the Internet, on rent (i.e. on demand) to the end users typically on a subscription basis.

- There are several SaaS applications, some of them are listed below:

- Billing and Invoicing System
- Help Desk Applications
- Human Resource (HR) Solutions
- Customer Relationship Management (CRM) applications

Benefits of Software as a service (SaaS):

➢SaaS supports **multitenant environment** i.e. it allows access to programs and recent software to large number of users through browser.

➢A SaaS provider **gives access to applications to multiple clients and users over web** by hosting and managing the given application in their or leased datacenters.

➢Provides **centralized management of data** like software control, maintenance, updation to new version, infrastructure, platform and resource requirements etc.

Platform as a Service (PaaS)

- Here instead of delivering software online, it supplies or rent a platform over some time for creating developing, testing, delivering and managing software applications like web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development

Google's App Engine, Force.com are examples of PaaS offering vendors.

Platform as a Service (PaaS)

Benefits :

- LOWER **ADMINISTRATIVE OVERHEAD**
- LOWER **COST OF OWNERSHIP**
- **SCALABLE SOLUTIONS**
- **MORE CURRENT SYSTEM SOFTWARE**

Issues :

- LACK OF PORTABILITY BETWEEN PAAS CLOUDS
- EVENT BASED PROCESSOR SCHEDULING
- SECURITY ENGINEERING OF PAAS APPLICATIONS

Infrastructure as a Service (IaaS)

- It rent complete IT infrastructure to the user like—servers and virtual machines (VMs), storage, data center, networks, operating systems through IP-based connectivity as part of an on-demand service.

- Cloud paradigm also serves as a business model apart from technology.

Infrastructure as a Service (IaaS)

IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

Apart from these resources, the IaaS also offers:

- Virtual machine disk storage
- Virtual local area network (VLANs)
- Load balancers
- IP addresses
- Software bundles

Infrastructure as a Service (IaaS)

Benefits :

- **Full Control through Administrative Access to VMs** on the computing resource.
- **Flexible and Efficient renting** of Computer Hardware.
- **Portability, Interoperability** with Legacy Applications.

Issues :

- COMPATIBILITY WITH LEGACY SECURITY VULNERABILITIES
- VIRTUAL MACHINE SPRAWL
- DATA ERASE PRACTICES

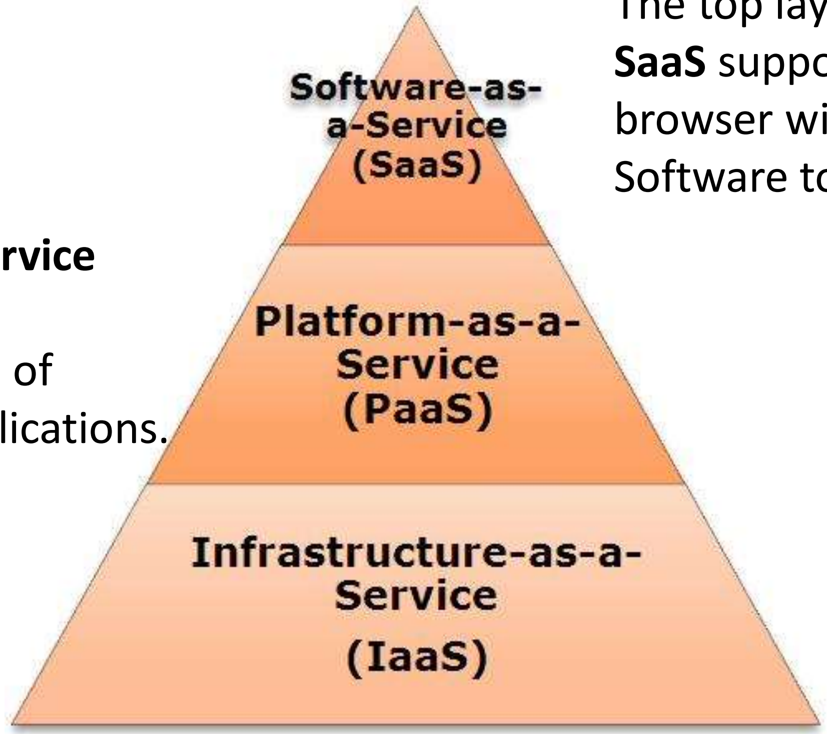
CLOUD SERVICE MODEL:

Layers in Cloud:

Different layers are outlined based on the kind of services provided by the Cloud

The top layer is **Software-as-a-Service (SaaS)**.
SaaS supports accessing user’s applications through a browser without the knowledge of Hardware or Software to be installed.

Middle layer is **Platform-as-a-Service (PaaS)** which mainly supports deployment and dynamic scaling of .NET, Python and Java based applications. One such an example of **PaaS is Google App Engine.**


Software-as-a-Service (SaaS)
Platform-as-a-Service (PaaS)
Infrastructure-as-a-Service (IaaS)

Bottom layer contains basic hardware resources like Memory, Storage Servers. Hence it is denoted as **Infrastructure-as-a-Service (IaaS). For example Amazon easy Storage Service (S3) and Amazon Elastic Compute Cloud (EC2).**

- **Cloud Computing can be considered as = SaaS+ PaaS+IaaS**

- Therefore Cloud connects the devices, data, applications, service, persons and business.

- Cloud services are considered as **distribution service -a service** for linking the resources (computing functions, data store, processing functions, networks, servers and applications) and provisions of coordinating between the resources.