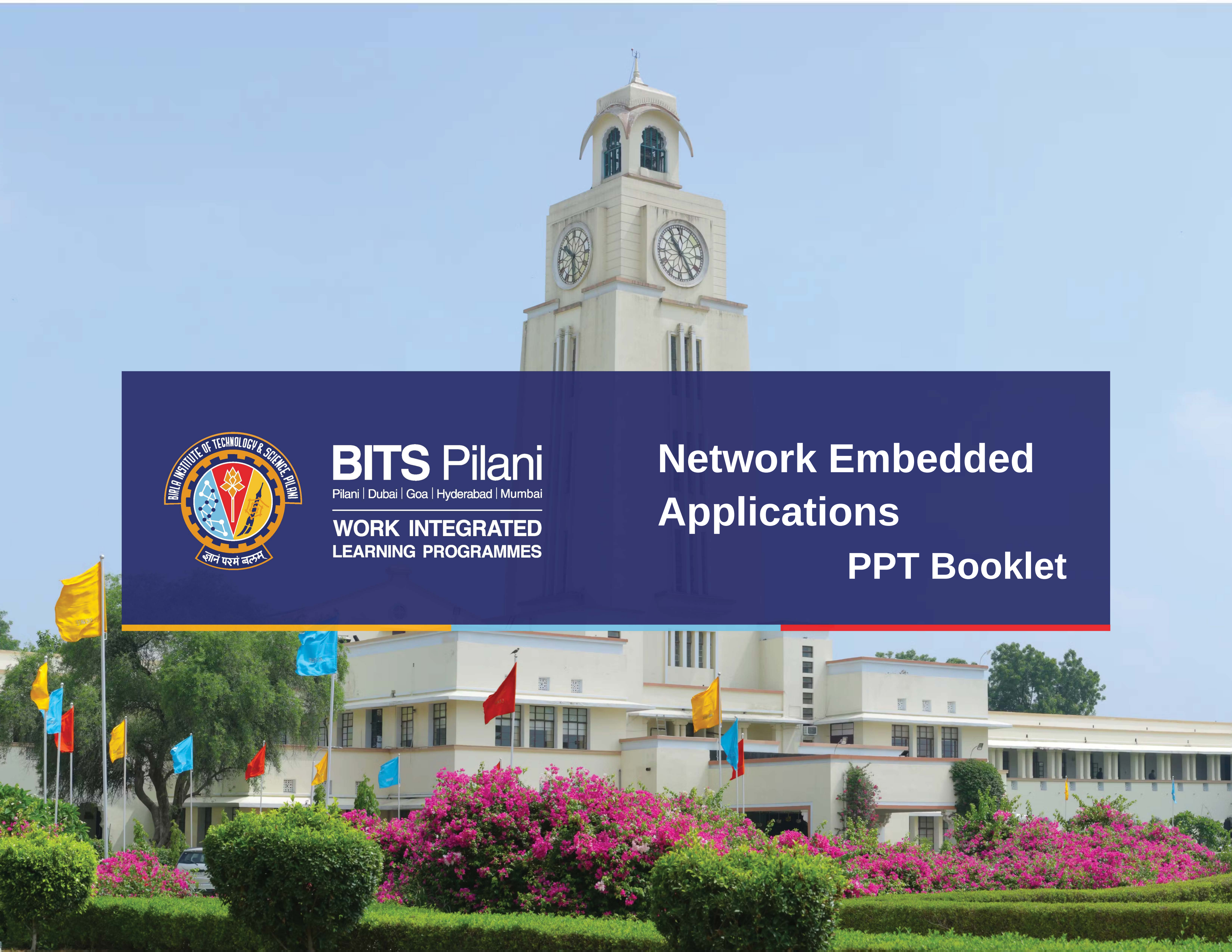


BITS Pilani

Pilani | Dubai | Goa | Hyderabad | Mumbai

WORK INTEGRATED LEARNING PROGRAMMES

Network Embedded Applications

PPT Booklet

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

IMP Note to Self

Start Recording

Network Embedded Applications (CSIW ZG656)

Guruprasad Shenai
gpshenai@wilp.bits-pilani.ac.in
+91- 9820319484

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

http://taxila-aws.bits-pilani.ac.in/

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

CS1: Introduction to the course

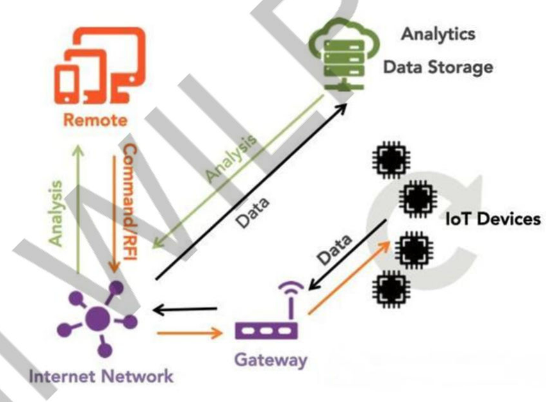
Industry 4.0

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

What is IoT?

IoT Ecosystem



IoT Layers

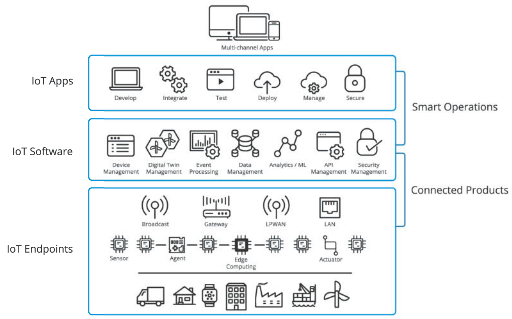
- Visualization (representation in graphs)
- Storage, Analytics and Query (also called Edge IT)
- Data integration (data cleaning, normalization and structuring)
- Network & Connectivity elements (wired/ wireless gateways, protocol converters)
- Devices / sensors / things / remote data providers

IoT Model

Devices	Send and receive data interacting with the
Network	Where the data is transmitted, normalised and filtered using
Edge Computing	Before landing in
Data Storage / Databases	Accessible by
Applications	Which process it and provide it to people who will
Act & Collaborate	

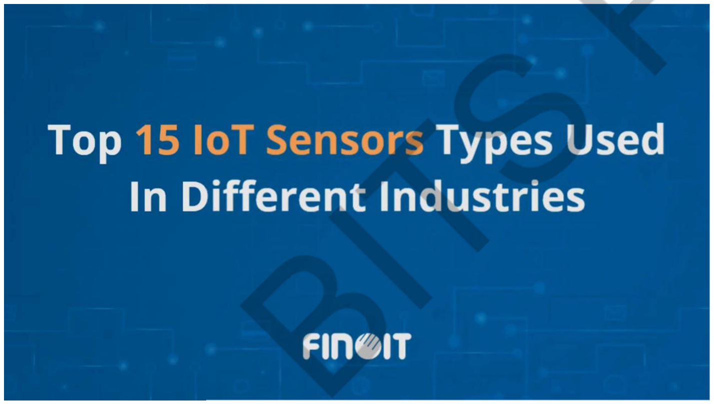
innovate achieve lead

IoT 3 Layer model



Sensors

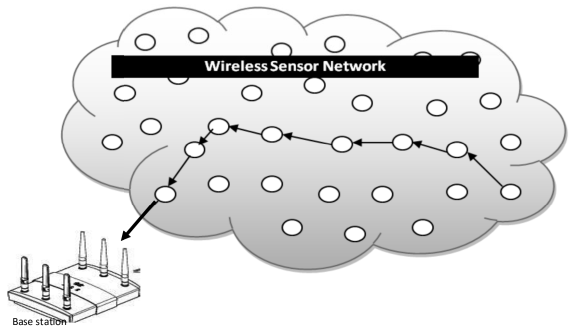
IoT Sensors



Wireless Sensor Networks

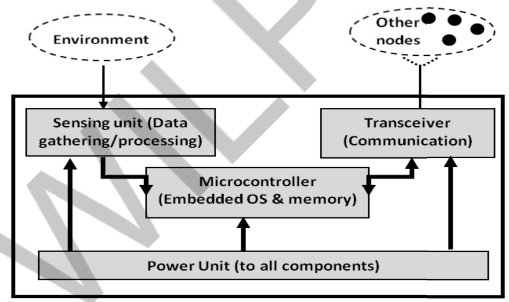
- Wireless Sensor Networks are networks that consists of sensors which are distributed in an ad hoc manner.
- These sensors work with each other to sense some physical phenomenon and then the information gathered is processed to get relevant results.
- Wireless sensor networks consists of protocols and algorithms with self-organizing capabilities.

Wireless Sensor Network



A sensor network is a wireless network that consists of thousands of very small nodes called *sensors*.

Wireless Sensor Network



WSN **Sensors** are equipped with sensing, limited computation, and wireless communication capabilities.

WSN V/S Ad-Hoc network

- Wireless sensor networks mainly use **broadcast** communication while ad hoc networks use **point-to-point** communication.
- Unlike ad hoc networks wireless sensor networks are **limited by sensors** limited power, energy and computational capability.
- Sensor nodes may **not have global ID** because of the large amount of overhead and large number of sensors.

Characteristics of WSN

- Wireless Sensor Networks mainly consists of **sensors. Sensors** are -
  - low power
  - limited memory
  - energy constrained due to their small size.
- Wireless networks can also be deployed in **extreme environmental** conditions and may be prone to enemy attacks.
- Although deployed in an ad hoc manner they need to be **self organized** and **self healing** and can face constant reconfiguration.

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

Design Challenges…

Large Scale Coordination
- The sensors need to coordinate with each other to produce required results.

Utilization of Sensors
- The sensors should be utilized in a ways that produce the maximum performance and use less energy.

Real Time Computation
- The computation should be done quickly as new data is always being generated.

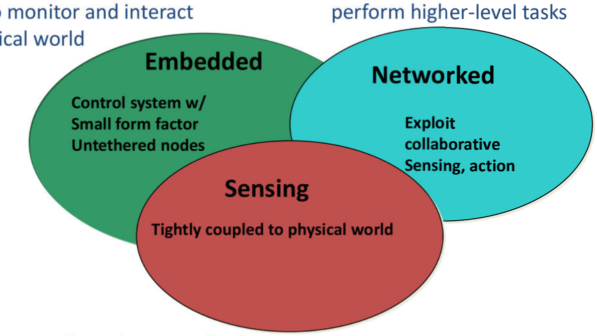
Operational Challenges

- Energy Efficiency
- Limited storage and computation
- Low bandwidth and high error rates
- Errors are common
  - Wireless communication
  - Noisy measurements
  - Node failure are expected
- Scalability to a large number of sensor nodes
- Survivability in harsh environments
- Experiments are time- and space-intensive

WSN Enablers

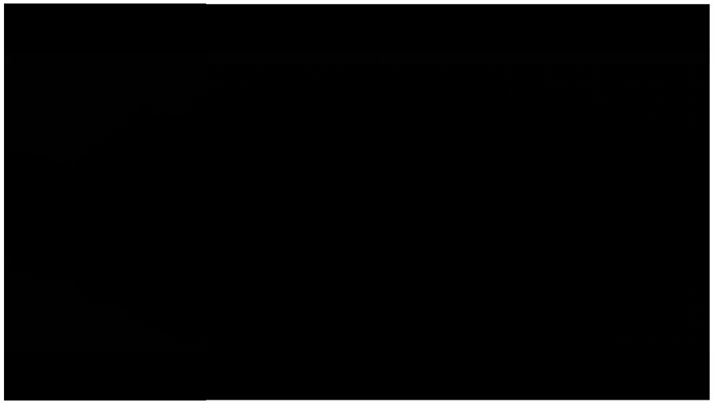
Embed numerous distributed devices to monitor and interact with physical world

Network devices to coordinate and perform higher-level tasks

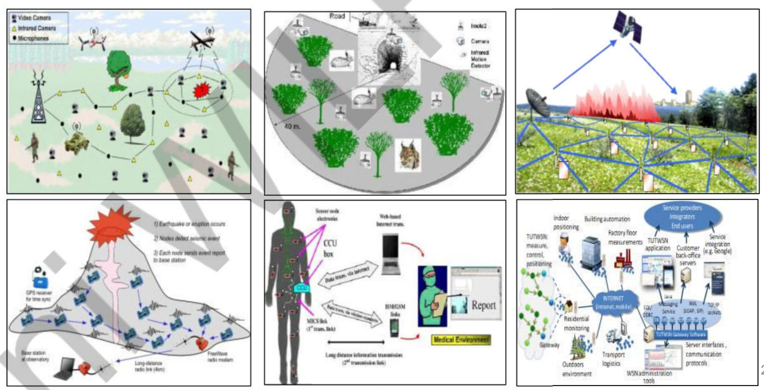


Exploit spatially and temporally dense, in situ, sensing and actuation

WSN Application



WSN Applications



WSN Applications…

The applications can be divided in three categories:

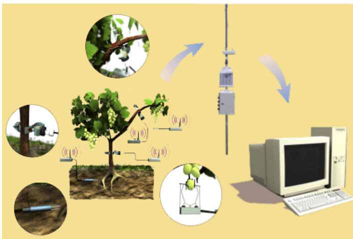
1. Monitoring of objects.
2. Monitoring of an area.
3. Monitoring of both area and objects.

Area Monitoring

- Environmental and Habitat Monitoring
- Precision Agriculture
- Indoor Climate Control
- Military Surveillance
- Treaty Verification
- Intelligent Alarms

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Area monitoring Example



- Precision agriculture aims at making cultural operations more efficient, while reducing environmental impact.
- The information collected from sensors is used to evaluate optimum sowing density, estimate fertilizers and other input needs, and to more accurately predict crop yields.

Monitoring Objects

- Structural Monitoring
- Eco-physiology
- Condition-based Maintenance
- Medical Diagnostics
- Urban terrain mapping

Monitoring Interactions between Objects and Space

- Wildlife Habitats
- Disaster Management
- Emergency Response
- Ubiquitous Computing
- Asset Tracking
- Health Care
- Manufacturing Process Flows

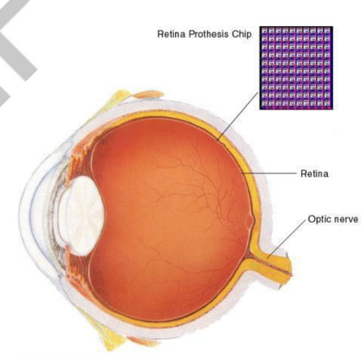
Example

innovate achieve lead

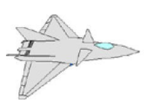
LoRaWAN

Biomedical

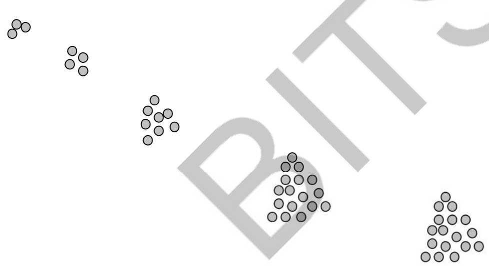
- Health Monitors
  - Glucose
  - Heart rate
  - Cancer detection
- Chronic Diseases
  - Artificial retina
  - Cochlear implants
- Hospital Sensors
  - Monitor vital signs
  - Record anomalies



Military

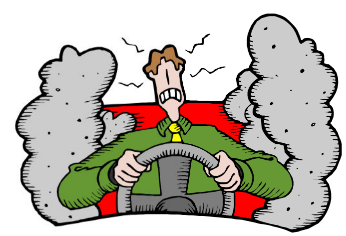


Remote deployment of sensors for tactical monitoring of enemy troop movements.



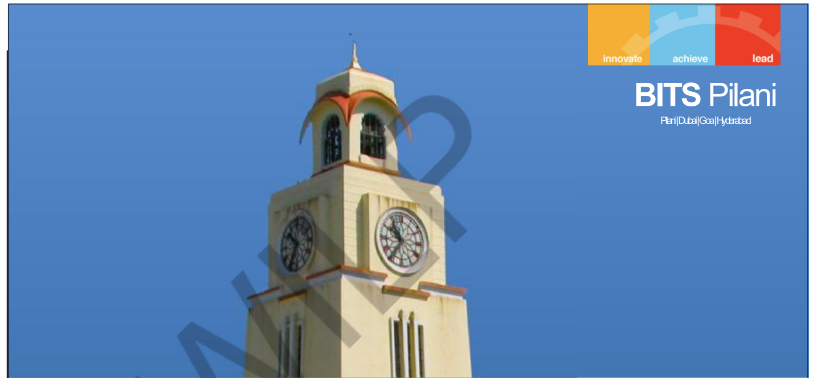
Traffic Management

- Future cars could use wireless sensors to:
  - Handle Accidents
  - Handle Thefts



✓Sensors embedded in the roads to:
- Monitor traffic flows
- Provide real-time route updates

Future World



Questions?

IMP Note to Self

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Network Embedded Applications (CSIW ZG656)

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

http://taxila-aws.bits-pilani.ac.in/

---

innovate achieve lead

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

CS2: Wireless Sensor Networks

---

innovate achieve lead

Wireless Sensor Network

---

innovate achieve lead

Protocol Stack

Appln
Transport
Network
Data Link
Physical

- ISM – 2.4GHz
- ZigBee
- LoRa, UWB
- Simple Modulation Technique
- Energy Aware - Short Range

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Protocol Stack

- Appln
- Transport
- Network
- Data Link
- Physical

- Energy Aware
  - Collisions
  - Idle Listening
  - Overhearing
  - Sleep state
    - Over- emitting
- Multi Channel

Protocol Stack

- Appln
- Transport
- Network
- Data Link
- Physical

- Data Centric
  - Energy Aware
  - Data Aggregation
- Query- Response

Protocol Stack

- Appln
- Transport
- Network
- Data Link
- Physical

- Appears only when WSN connects to standard infrastructure
- Nodes – Gateway
- Gateway needs Transport

Protocol stack

- Appln
- Transport
- Network
- Data Link
- Physical

- One single application
- Whole protocol stack designed for a special appln
- Whole network is seen as an instrument
- Application layer distributed along the whole protocol stack
- Not appear explicitly
- Explicit application
  - Sensor management
  - Task management
  - Data advertisement
  - sensor query & data extraction

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Protocol Stack - Issues

- Dynamic environment
- Power control - Longetivity
- Protocol place in the sensor node architecture
- Protocol availability

Dynamic Environment

- Sensor nodes address a dynamic environment
  - Nodes have to reconfigure themselves –
  - to adapt to the changes.
- resources are very limited
- Network - adapts its functionality to a new situation -lower the use of the scarce energy & memory - maintain the integrity of its operation

Error Control

- Normally resides in all protocol layers – worst case scenarios are handled
- WSN this redundancy- too expensive
- Adopting a central view on how error control is performed and cross-layer design reduces the resources spent for error control

Power Control

- Traditionally done only at the physical layer,
- Energy consumption- is a major design constraint - found in all

innovate achieve lead

Protocol Place

Habitat Monitoring

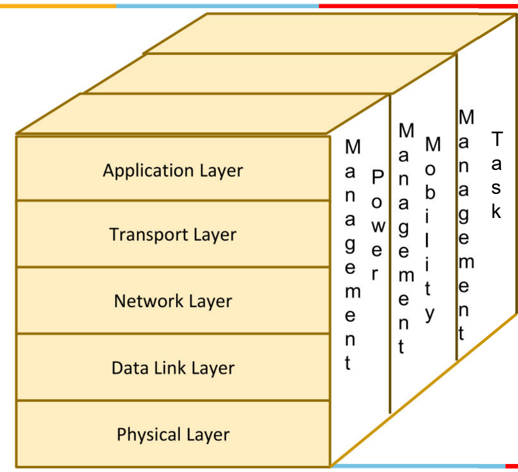
Protocol Place

- Time synchronization, localization, calibration.
- Shift their place in the protocol stack - transient phase is over
- Data produced by some of these algorithms might make a different protocol stack more suited for the sensor node
- Localization algorithm for static sensor networks might enable a better routing algorithm

Protocol Availability

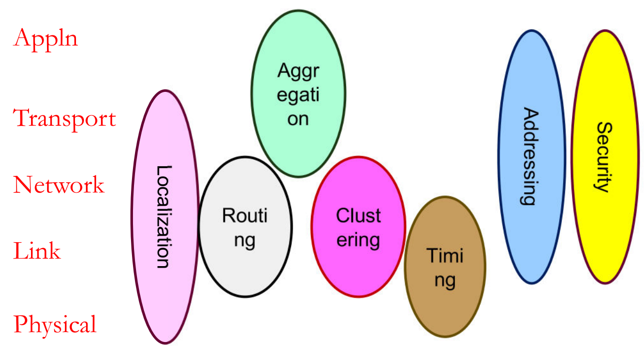
- New protocols might become available after network deployment
- In specific conditions- some of the sensor nodes might use a different protocol stack that better suits their goal & the environment
- Changing or Updating at run time parts of the software on the nodes is important

3D Protocol Stack



BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Protocol Placement



Sensor Management Protocol

Rules related to data aggregation, attribute based naming, clustering of sensor nodes	Data related to sensor location algorithm	Time synchronisation of sensor nodes
Turning sensors on and off	Querying sensor network configuration, status of nodes and reconfiguration of sensor network	Security management

Power management plane

- Manage sensor power
- Alert neighbours to low power status
  - Does not participate in routing
  - Power is used for sensing

Mobility management plane

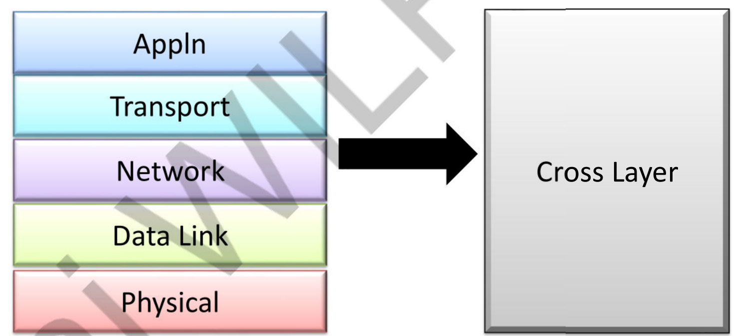
- DETECTS AND REGISTERS MOVEMENTS OF NODES
- KEEP TRACK OF NEIGHBOURS
- BALANCE POWER AND TASK USAGE

innovate achieve lead

Task management plane

- Balances and schedules the sensing tasks in the region
- Not all sensor nodes in the region participate in tasks

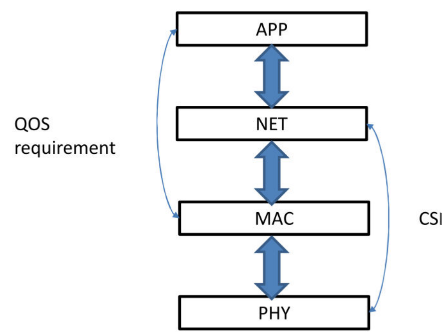
Cross Layer Protocol



Cross Layer Protocol

- Non adjacent layers interact
- Application layer – MAC layer interact to achieve better scheduling
- Physical layer exchanges Channel Status Information to improve routing

Cross layer protocols



Challenges of cross layer protocols

- Complexity – increased system design effort
- Scalability Issues – Does not work for large network
- Interoperability – May face integration issues with standardised systems
- Security risks – Potential security risk due to open layer communication

Questions?

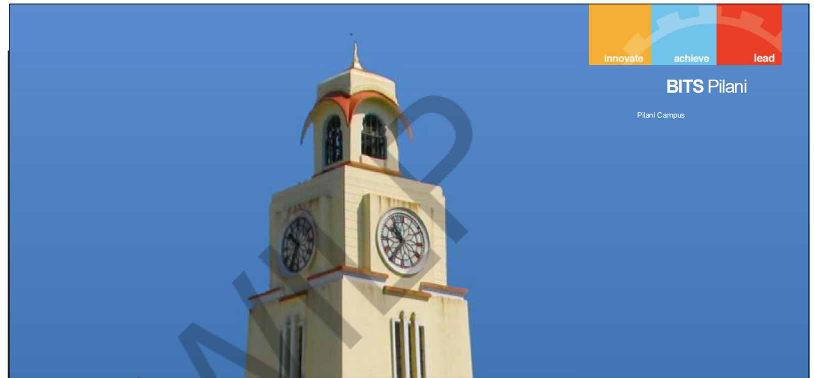
BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Network Embedded Applications (CSIW ZG656)

BITS Pilani
Pilani|Dubai|Goa|Hyderabad

Prof. Guruprasad Shenai

http://taxila-aws.bits-pilani.ac.in/



Course Name (Network Embedded Applications)

Lecture No- 03

Start Recording

Recap of CS #02

What We Learned:

- Introduction to Wireless Sensor Networks (WSNs)
- **Definition:** WSNs consist of multiple devices (sensor nodes) with processing, sensing, and wireless communication capabilities.

Traditional Monitoring Challenges:

- Expensive, bulky devices
- Manual deployment and calibration
- Limited data retrieval and risk of data loss

Modern Approach:

- Small, inexpensive, energy-efficient sensor nodes
- Wireless data transmission
- Easier deployment and self-organizing networks

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

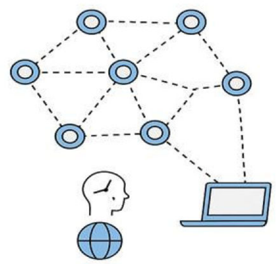
Key Features & Importance of WSNs

Key Features Short-range wireless communication (radio, infrared, etc.)

- Localized processing and data transmission
- Self-configuring and maintenance algorithms

Importance in Computing & Monitoring

- Supports real-time environmental monittoring
- Enhances decision-making for humans and machines



CS: #03 Agenda

Module	Number	Topic
M2	CS3	NTP, HBS, TDP, RBD

Recorded Lectures Review

Serial No	Topic
RL 2.4.1	WSN Time Synchronization
RL 2.4.2	Sender- Receiver Synchronization
RL 2.4.3	Receiver- Receiver Synchronization
RL 2.5.1	WSN Localization, Need for Localization and Issues

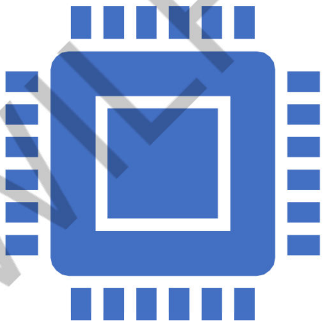
CS: 3 | Module: 2

NTP, HBS, TDP, RBD

innovate achieve lead

WSN – structural health monitoring

Synchronisation in network embedded system



Sensor Deployment

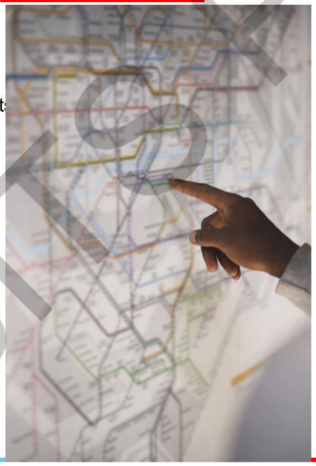
Deployed in homes, plantations, rivers, streets

Events

- Target tracking
- Speed estimation
- Ocean/river current monitoring etc.

Knowledge of time is a necessity

- Time stamp for security
- Measure the phenomenon etc
- Synchronisation of nodes



Synchronisation

- What is synchronisation?
- Why do we need synchronisation?
- What are the challenges in synchronisation of clocks?

Time synchronization in Wireless Sensor Networks (WSNs)

Time synchronization in Wireless Sensor Networks (WSNs)

- **R**efers to the process of ensuring that all sensor nodes in the network maintain a common time reference within a specified accuracy.
- It is essential for coordinated operations such as event detection, data fusion, secure time stamping, and energy-efficient communication.
- Time synchronization helps overcome challenges like clock drift, network delays, and environmental interference to maintain accurate and reliable sensor data across distributed nodes.

Time Synchronization in Wireless Sensor Networks

Applications

- Small intelligent devices will be deployed in homes, oceans, highways
- Target tracking, speed estimation and ocean current montoring
- Timestamped data packets for security
- Fecures synchroni-zations

Importance

- Data fusion, voice and video synchronization)
- Medium access schemes (TDMA
- Energy-saving nodes can be turned off

Why is Time Synchronization Important?

- Ensures **accurate event detection** (e.g., target tracking, speed estimation).
- Enables **secure time stamping** for data integrity.
- Supports **energy-efficient communication** (e.g., TDMA scheduling).
- Allows seamless **fusion of data** (voice, video, sensor readings) from multiple nodes.

Challenges in Time Synchronization

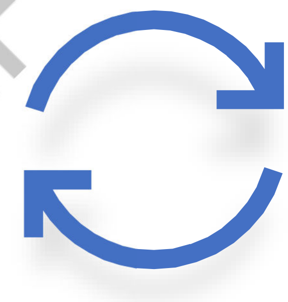
- **Clock drift** over time causes synchronization errors.
- **Multihop delays** affect time consistency across nodes.
- **Energy constraints** limit synchronization frequency.
- **Unattended deployments** (deep space, ocean floors) make manual adjustments impractical.
- **Communication delays** due to software and medium access variations.
- Nodes must be **energy-efficient, low-cost, and compact**.

Objective

1. MAINTAIN SIMILAR TIME
2. ASSIST IN MAINTAINING ENERGY EFFICIENCY, LOW COST ETC
3. ADDRESS DRIFT IN CLOCKS

Design challenges

- Application driven
- Time synchronisation is dictated by the application
- Some issues to be addressed
  - Robust: Sensor nodes may fail, and the failures should not have significant effect on the time synchronization error
  - Dependency on a specific master may lead to cascade effect of failure
  - Failures should be contained and not cascaded



Design Challenges & Factors Affecting Time Synchronization

Challenges in Sensor Networks:

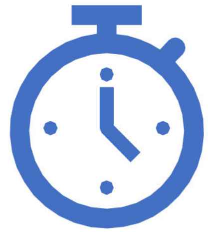
- Mapping sensor network time to **Internet time (e.g., UTC)**.
- **Energy constraints** require lightweight synchronization.
- **Frequent node failures** disrupt network synchronization.
- **Interference** can cause network fragmentation, leading to **desynchronized regions**.
- **Large propagation delays** in hierarchical synchronization structures.

Factors Affecting Synchronization

- **Environmental interference** causing communication delays.
- **Clock drift** over time affecting accuracy.
- **Multihop transmission delays** leading to timing inconsistencies.

Design challenges

- Energy aware: A time synchronization protocol should use the minimum number of messages to synchronize the nodes in the earliest time.
  - Load of time synchronisation should be shared
  - If a partition fails, danger of parts of network drifting apart is high
- Server-less: Precise time server may not be available. Sensor nodes should be able to synchronize to a common time without the precise time servers.



Design Challenges

Lightweight: The complexity of the time synchronization protocol has to be low in order to be programmed into the sensor nodes

- The synchronization protocol may be programmed into a FPGA or designed into an ASIC.
- Tight integration of time synchronisation protocol with hardware, the delay and variation of the processing may be reduced.
- Expensive

Tunable service: Should be switched ON or OFF depending on the application requirement

Factors affecting time synchronisation

- Temperature
- Phase noise
- Frequency noise
- Asymmetric delay
- Clock glitches

Software Access delay

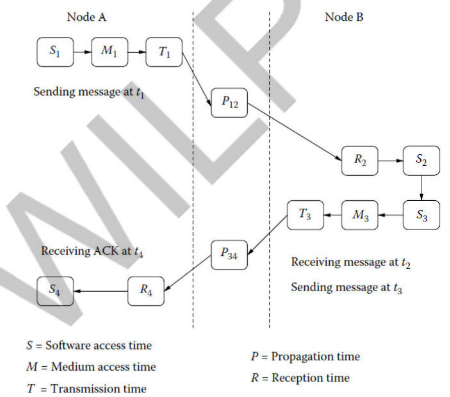
What is the granularity of a clock running at 500 KHz vs a clock running at 1MHz?

A Mica mote is running at 4MHz having clock granularity of 0.25µs. If the node is 80% loaded and it takes 100 cycles to obtain the time, calculate the software access delay

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Mica Mote

Round trip time

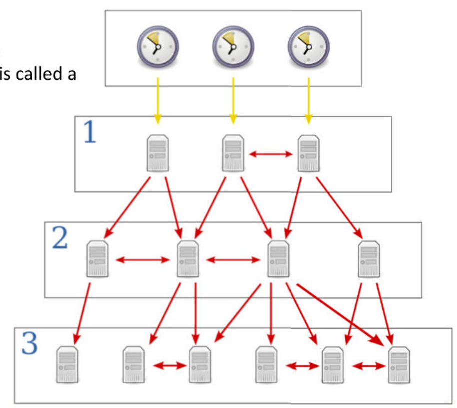


Time Synchronization Protocols

- **Network Time Protocol (NTP)** – Millisecond-level accuracy but **energy-intensive**.
- **Timing-sync Protocol for Sensor Networks (TPSN)** – Hierarchical approach for synchronization.
- **Reference-Broadcast Synchronization (RBS)** – Uses broadcast reference messages for accuracy.
- **Time-Diffusion Synchronization Protocol (TDP)** – Distributes time updates efficiently.
- **Adaptive Clock Synchronization & ARSP** – Adjusts based on clock drift and network conditions.
- **H-Sensor Broadcast Synchronization (HBS)** - Achieve precise time synchronization in large-scale WSNs.
- **Rate-Based Diffusion (RBD) Protocol** - nodes synchronize their clocks by adjusting their local time based on the rate of clock drift observed over multiple communication rounds.

NTP(Network Time Protocol)

- Based on Hierarchical Client – Server Architecture
- Each level of this hierarchy is called a **Stratum**



innovate achieve lead

NTP

Stratum 0

These are high-precision timekeeping devices such as atomic clocks, GPS or other radio clocks.

They generate a very accurate pulse per second signal that triggers an interrupt and timestamp on a connected computer.

Stratum 0 devices are also known as reference clocks.

Stratum 1

These are computers whose system time is synchronized to within a few microseconds of their attached stratum 0 devices.

Stratum 1 servers may peer with other stratum 1 servers for sanity check and backup.

They are also referred to as primary time servers

NTP

Stratum 2

- These are computers that are synchronized over a network to stratum 1 servers.
- Often a stratum 2 computer queries several stratum 1 servers. Stratum 2 computers may also peer with other stratum 2 computers to provide more stable and robust time for all devices in the peer group.

NTP

Stratum 3

- These are computers that are synchronized to stratum 2 servers.
- They employ the same algorithms for peering and data sampling as stratum 2, and can themselves act as servers for stratum 4 computers, and so on.

Time Stamps

The 64-bit timestamps used by NTP consist of:

32-bit part for seconds and a 32-bit part for fractional second, giving a time scale that rolls over every 2^{32} seconds (136 years)

Theoretical resolution of 2^{-32} seconds (233 picoseconds).

NTP uses an epoch of January 1, 1900.

Therefore, the first rollover occurs on February 7, 2036.

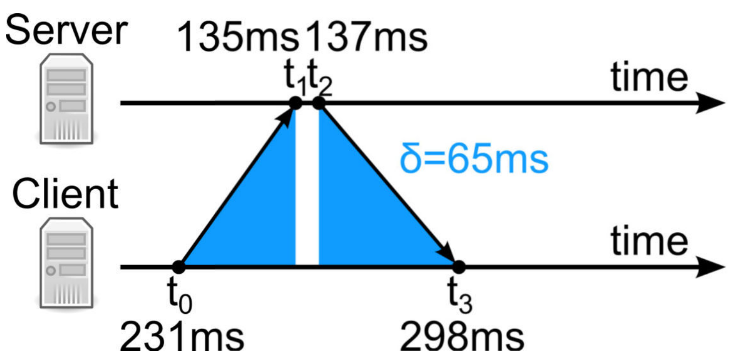
NTPv4 introduces a 128-bit date format:

64 bits for the second and 64 bits for the fractional-second.

The most-significant 32-bits of this format is the Era Number which resolves rollover ambiguity in most cases.

The 64-bit value for the fraction is enough to resolve the amount of time it takes a photon to pass an electron at the speed of light.

Working



Legend

- t0 is the client's timestamp of the request packet transmission,
- t1 is the server's timestamp of the request packet reception,
- t2 is the server's timestamp of the response packet transmission and
- t3 is the client's timestamp of the response packet reception.

Synchronising clocks

NTP client regularly polls one or more NTP servers. The client must compute its time offset and round-trip delay.

Time offset θ, the difference in absolute time between the two clocks, is defined by

$$\theta = \frac{(t_1 - t_0) + (t_2 - t_3)}{2}$$

Clock drift is calculated as:

offset/time

offset = ((t1-t0)+(t2-t3))/2

time = time difference between two syncs

$$\delta = (t_3 - t_0) - (t_2 - t_1)$$

Problem1: Calculate Offset in NTP

- A client sends a request to an NTP server at **T1 = 12:00:00.500** and receives a response from the server at **T4 = 12:00:01.000**. The server logs the arrival of the request at **T2 = 12:00:00.600** and sends the response at **T3 = 12:00:00.800**. **Find the clock offset (θ) and round-trip delay (δ).**

Solution:

1. Round-Trip Delay (δ)

$$\delta = (T4 - T1) - (T3 - T2)$$

$$\delta = (12:00:01.000 - 12:00:00.500) - (12:00:00.800 - 12:00:00.600)$$

$$\delta = (0.500) - (0.200) = 0.300 \text{ sec}$$

2. Clock Offset (θ)

$$\theta = \frac{(T2 - T1) + (T3 - T4)}{2}$$

$$\theta = \frac{(12:00:00.600 - 12:00:00.500) + (12:00:00.800 - 12:00:01.000)}{2}$$

$$\theta = \frac{(0.100) + (-0.200)}{2} = \frac{-0.100}{2} = -0.050 \text{ sec}$$

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Problem2: Calculate Synchronization Error Due to Network Delay

- A system uses NTP to synchronize with a server, but due to network congestion, the round-trip time fluctuates. If the server processes the request in **0.020 sec** and the total observed round-trip delay is **0.500 sec**, what is the estimated one-way delay?

Solution:

Assuming symmetric delay, one-way delay is:

$$\text{One-way delay} = \frac{\text{Total round-trip delay} - \text{Processing time at server}}{2}$$

$$\text{One-way delay} = \frac{0.500 - 0.020}{2}$$

$$\text{One-way delay} = \frac{0.480}{2} = 0.240 \text{ sec}$$

Limitation of NTP for embedded systems?

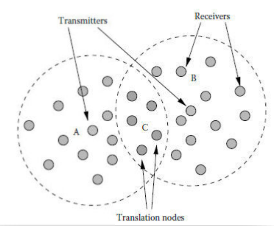
- Computation intensive
- Requires precise time servers
- Energy consumption not taken into account
- May suffer from large time delays

Reference Broadcast Synchronisation

- Receiver uses the physical layer broadcasts for comparing the clocks.
- RBS allows nodes to synchronize their clocks to the resolution necessary for example for wireless sensor network applications.
- Rather than broadcasting a timestamp in a synchronization packet, RBS allows the nodes receiving the synchronization packets to use the packet's time of arrival as a reference point for clock synchronization.
- Because most of the non-deterministic propagation time involved in transmitting a packet over a wireless channel lies between the construction of the packet and the sender's transmitter (e.g., sender's queue delay, MAC contention delay, etc.), by timestamping only at the receiver, RBS removes most delay uncertainty involved in typical time synchronization protocols.

RBS

Receivers communicate with each other
- –Receivers record time of arrival
- –Communicate the time of arrival with
- –The offset in arrival is calculated



innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

RBS Translation nodes

- Sensor nodes within broadcast region of both transmitter region are known as translation nodes
- On occurrence of an event:
  - Message describing the event with timestamp is translated by translation node
  - The translate message is routed back to the sink
  - This message is used for synchronisation

Problem 1: Calculate Clock Offset

Using RBS

- A reference node broadcasts a reference message to two receiver nodes, **Node A** and **Node B**. The recorded arrival times of the message at both nodes are:
- **Node A:** 10.250 seconds
- **Node B:** 10.275 seconds

Find the clock offset between Node A and Node B.

Solution :

The clock offset **θ** between two nodes is given by:

$$\theta = T_B - T_A$$

$$\theta = 10.275 - 10.250 = 0.025 \text{ sec}$$

- Clock offset between Node A and Node B: 0.025 seconds
- This means Node B's clock is ahead of Node A by 0.025 seconds.

Problem 2: Adjusting Clocks Using RBS

A reference node sends a broadcast message. Three nodes record the arrival times:

- **Node P:** 12.500 sec
- **Node Q:** 12.530 sec
- **Node R:** 12.515 sec

Find the adjustments required for each node to synchronize with the average clock time.

Solution :

1. Compute the **average arrival time**:

$$T_{\text{avg}} = \frac{T_P + T_Q + T_R}{3}$$

$$T_{\text{avg}} = \frac{12.500 + 12.530 + 12.515}{3} = \frac{37.545}{3} = 12.515 \text{ sec}$$

2. Compute adjustments for each node:
   - **Node P Adjustment:** $12.515 - 12.500 = +0.015$ sec (slow, needs to be increased)
   - **Node Q Adjustment:** $12.515 - 12.530 = -0.015$ sec (fast, needs to be decreased)
   - **Node R Adjustment:** $12.515 - 12.515 = 0.000$ sec (already synchronized)

Node P: Increase time by **0.015 sec**
Node Q: Decrease time by **0.015 sec**
Node R: No adjustment needed

Advantages of Reference-Broadcast Synchronization (RBS)

- **High Synchronization Accuracy:** Removes **send and receive uncertainties**, leading to better precision.
- **No Need for Global Time Source:** Synchronization is achieved **locally** among sensor nodes.
- **Energy-Efficient:** Reduces the need for **frequent clock adjustments**, saving node battery life.
- **Multi-Hop Synchronization:** Supports **distributed synchronization**, making it scalable for large networks.
- **Resistant to Clock Drift:** Nodes continuously synchronize based on shared reference broadcasts.
- **No Need for Dedicated Time Servers:** Works without relying on **external time sources** like GPS or NTP.

Challenges of Reference-Broadcast Synchronization (RBS)

- **Requires Extra Communication Overhead:** Additional reference messages are needed, **increasing network traffic**.
- **Limited to Nodes in the Same Broadcast Domain:** Only nodes **receiving the same reference signal** can synchronize.
- **Vulnerable to Wireless Interference:** Packet **loss or delays** due to interference can reduce accuracy.
- **Multi-Hop Synchronization Complexity:** Synchronizing nodes over **multiple hops** is challenging.
- **Not Ideal for Real-Time Systems:** The reliance on reference messages may introduce **latency** in time-critical applications.
- **Potential Security Risks:** Broadcast signals can be **hijacked or manipulated**, leading to false synchronization.

H-Sensor Broadcast Synchronisation

- Designed for clusters
- Assumes cluster heads are high power sensors
  - Capable of accessing GPS signal
  - Message encryption
- Each timing message tagged with
  - Sequence number
  - Message authentication code using shared key
- Public/private keys are preloaded and are protected by tamper resistant hardware

H-Sensor Broadcast Synchronisation

Authentication of timing message
- Neighbouring cluster heads check authentication code using stored public key
- In case of unauthentic message, an alarm is broadcast

Advantages of HBS in WSN

- **High Accuracy:** H-sensors provide a reliable time reference, reducing synchronization errors.
- **Energy Efficiency:** Regular sensor nodes do not need to send frequent synchronization messages, **saving battery life**.
- **Scalability:** Suitable for **large WSN deployments** with minimal overhead.
- **Robustness:** Reduces **clock drift** by using periodic broadcast synchronization.
- **Low Communication Overhead:** Fewer messages exchanged compared to peer-to-peer synchronization methods.
- **Faster Convergence:** Synchronization occurs quickly due to **one-to-many broadcasting**.

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

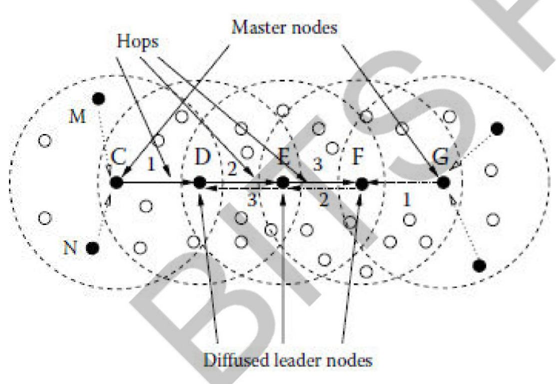
Challenges of HBS in WSN

- **Dependency on H-Sensors:** If H-sensors fail, synchronization accuracy may degrade.
- **Limited Coverage Area:** H-sensors must be strategically placed for optimal **coverage and efficiency**.
- **Environmental Interference:** Wireless interference can cause **packet loss**, affecting time synchronization.
- **Initial Deployment Cost:** Requires additional high-powered H-sensors, **increasing costs** compared to fully homogeneous networks.
- **Multi-Hop Synchronization Delay:** Nodes farther from H-sensors may experience slight synchronization **delays**.
- **Security Concerns:** H-sensor signals can be vulnerable to **spoofing or jamming attacks**.

Time Diffusion Synchronisation Protocol

- Maintains time throughout network within certain tolerance
- Tolerance levels adjusted as per application requirement
- Automatic self configuration performed by electing master nodes
  - Election process sensitive to energy requirements
- Deployed in unattended areas

TDP



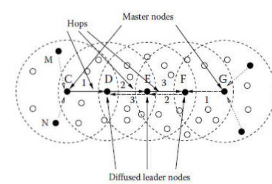
TDP

Elected nodes C & G

Master node measure round trip delay with their neighbours

Neighbours then self determine if to become diffuse leader node

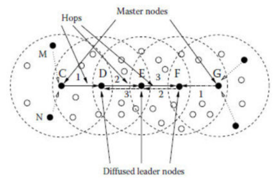
Elected diffuse leaders send reply to master and send message to neighbours to measure RTT



innovate achieve lead

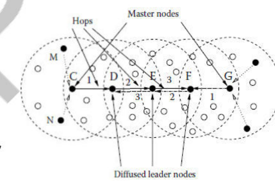
TDP

- Nodes M,N and D are diffuse leader nodes of C
- On receipt of reply, master calculates RTT and SD of RTT
- One-way delay is RTT/2
- Master send time stamped message to neighbour containing SD



TDP

- On receipt of time stamped mssg diffuse leader nodes adjust time delay and broadcast time stamped mssg.
- This mssg accounts for one-way delay and SD of RTT from diffuse leader.
- Process continues n times, where n is number of hops from master node
- What are the number of hops in the scenario?



TDP

On receipt of multiple time stamped message from masters, receiver uses sd as weight to determine the time diffused and calculate the new time

Master nodes are autonomously elected

If nodes die, other nodes self determine to become masters

TDP can operate as a serverless system

Problem 1: Calculating Average Time in a Diffused Network

A Time Diffusion Synchronization Protocol (TDP) is used in a wireless sensor network where **four nodes (A, B, C, and D)** receive a synchronization message and record the following timestamps:

- **Node A:** 10.500 sec
- **Node B:** 10.450 sec
- **Node C:** 10.525 sec
- **Node D:** 10.475 sec

Using the TDP approach, find the average network time to which all nodes should synchronize.

Solution :

The average time **T_avg** is calculated as:

$$T_{\text{avg}} = \frac{T_A + T_B + T_C + T_D}{4}$$

$$T_{\text{avg}} = \frac{10.500 + 10.450 + 10.525 + 10.475}{4}$$

$$T_{\text{avg}} = \frac{41.95}{4} = 10.4875 \text{ sec}$$

Problem 2: Synchronization Adjustment for Nodes

- Using the previous problem, calculate the adjustment required for each node to synchronize with **T_avg = 10.4875 sec**.

Solution :

Each node's adjustment is:

$$\Delta T = T_{\text{avg}} - T_{\text{node}}$$

- **Node A:** $10.4875 - 10.500 = -0.0125$ sec (decrease time)
- **Node B:** $10.4875 - 10.450 = +0.0375$ sec (increase time)
- **Node C:** $10.4875 - 10.525 = -0.0375$ sec (decrease time)
- **Node D:** $10.4875 - 10.475 = +0.0125$ sec (increase time)

Advantages of TDP

Advantages of TDP in WSN :

- **Scalability:** Works efficiently in **large-scale, multi-hop** sensor networks.
- **Fault Tolerance:** No single point of failure; synchronization is **distributed** across multiple nodes.
- **Energy Efficiency:** Reduces unnecessary message exchanges, conserving **battery life**.
- **Clock Drift Compensation:** Uses an **averaging mechanism** to minimize errors caused by clock drift.
- **Robust to Network Delays:** Handles **variable network latencies** better than single-source synchronization protocols.
- **Self-Organizing:** Nodes dynamically adjust synchronization based on local observations.

Challenges of TDP

Challenges of TDP in WSN:

- **Computational Overhead:** Nodes must calculate **average time values**, which increases processing time.
- **Synchronization Accuracy:** While it minimizes errors, it may not achieve **millisecond-level precision** like some other protocols (e.g., RBS).
- **Multi-Hop Delays:** Time information diffusion can be **delayed** in large networks.
- **Initial Synchronization Complexity:** Requires proper **selection of master nodes** for accurate time distribution.
- **Environmental Factors:** Wireless interference and **network congestion** can affect time synchronization accuracy.

Rate Based Diffusion Protocol

- Rate-Based Diffusion Protocol is a **time synchronization technique** used in WSNs where nodes synchronize their clocks by adjusting their local time based on the rate of clock drift observed over multiple communication rounds. It ensures **gradual synchronization** rather than abrupt corrections, improving stability and energy efficiency.

Algorithm:

1. Repeat following steps with some frequency:
2. For each sensor n_i in the network do
3. Exchange clock times with the neighbours
4. For each neighbour n_j do
5. Let the time difference between n_i and n_j be t_i-t_j
6. Change time of n_i to t_i- $r_{ij}(t_i$-$t_j)$

RBD

Sensor nodes exchange their time information with neighbours

Sensors adjust their time by $r_{ij}(t_i-t_j)$

Where

r_{ij} is the diffusion rate

(t_i-t_j) is the time difference between nodes n_i and n_j

Advantages & Challenges of Rate-Based Diffusion Protocol

- **Advantages:**
  - **Energy Efficient** – Avoids frequent resynchronization messages.
  - **Stable Synchronization** – Prevents sudden large clock adjustments.
  - **Scalability** – Works well in large WSNs with many nodes.
- **Challenges:**
  - **Slow Convergence** – Synchronization takes time to stabilize.
  - **Drift Variability** – Different sensor nodes may have unpredictable clock drifts.
  - **Communication Overhead** – Nodes must exchange drift data periodically.

Problem1 : Calculating Adjusted Clock Rate using RBD protocol

- A sensor node has a **local clock drift rate of +2 ms per second** compared to a reference clock. The node applies Rate-Based Diffusion (RBD) by adjusting its clock rate gradually. If the adjustment step is **0.5 ms per second per synchronization cycle**, how many cycles will it take to synchronize the node with the reference clock?
- **Solution:**
  1. **Initial drift rate** = +2 ms/sec
  2. **Adjustment per cycle** = 0.5 ms/sec
  3. **Target drift rate** = 0 ms/sec (perfect synchronization)
  4. Number of cycles needed:

$$\text{Cycles} = \frac{\text{Initial drift rate}}{\text{Adjustment per cycle}}$$

$$= \frac{2}{0.5} = 4 \text{ cycles}$$

It will take **4 synchronization cycles** to fully correct the drift

Problem 2: Synchronization Time Estimation using RBD protocol

- In a WSN using RBD, Node A and Node B exchange their clock rates. Node A has a drift of **+3 ms/sec**, and Node B has a drift of **-1 ms/sec**. If the nodes adjust their drift at a rate of **0.4 ms/sec per cycle**, how long will it take for them to synchronize?
- Solution:
  1. **Relative drift between A and B:**

$$\text{Total drift} = (3 - (-1)) = 4 \text{ ms/sec}$$

  2. **Adjustment per cycle** = 0.4 ms/sec

$$\text{Cycles required} = \frac{4}{0.4} = 10 \text{ cycles}$$

Node A and Node B will synchronize in **10 cycles**.

innovate achieve lead

Thank you

BITS Pilani, Pilani Campus

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

IMP Note to Self

Start Recording

IMP Note to Students

- It is important to know that just login to the session does not guarantee the attendance.
- Once you join the session, continue till the end to consider you as present in the class.
- IMPORTANTLY, you need to make the class more interactive by responding to Professors queries in the session.
- **Whenever Professor calls your number / name ,you need to respond, otherwise it will be considered as ABSENT**

Setting up a wireless network to monitor environment

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Wireless Sensor Network - Localization

WSN Localisation

Why Localisation?

- Dynamic environment
- Hundreds of sensors are placed randomly – over a large area
- Initial location of the nodes may been unknown
- Estimation of a nodes position used
  - Measurement without position is useless
  - Allows energy efficient geographic routing
  - Self-organization and Self- healing is easier
  - Obstacles can be found and by-passed
  - Tracking – Measurement itself

innovate achieve lead

What is localisation?

Actual placement of sensors is tedious and cost intensive

Sensors are distributed randomly

Locating sensors is important to make sense of the data being recorded

Importance of localisation

Sensor measurements without location information are useless.

Knowledge of positions allows energy-efficient geographic routing methods without route discovery.

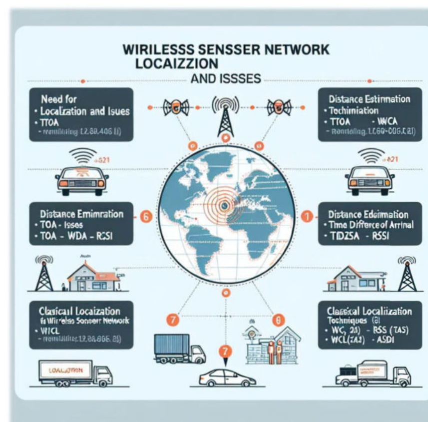
Basic processes in sensor networks like self-organization or self-healing can be applied very efficiently with position information.

Obstacles can be found and consequently bypassed easily.

Often, the position of a sensor node itself is the demanded parameter, e.g., in applications with target tracking, where sensor nodes are attached to a moving phenomenon.

Asset Tracking

Litum
Transform. Reach beyond.

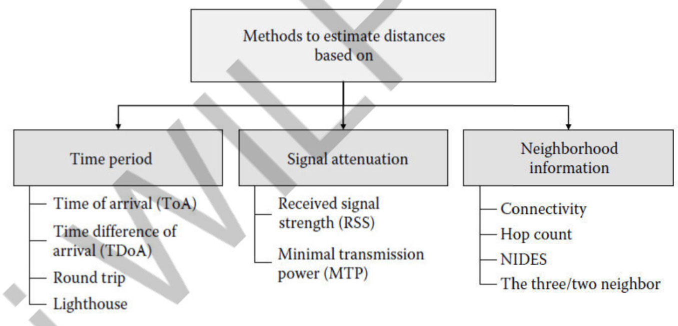


Locating a sensor

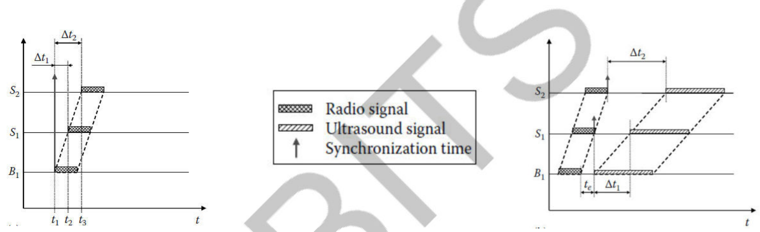
GPS/GLONASS based system
- Costly and not energy efficient
- Size a constraint

Use sensors themselves to perform location detection
- Use beacons

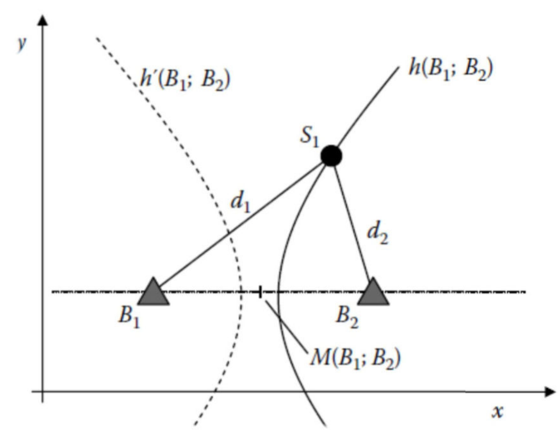
Distance estimation



TOA



TDoA



BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

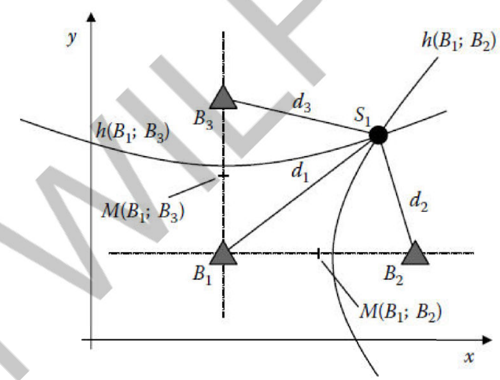
TDoA

$$\Delta d = \Delta t \cdot c = d_1 - d_2$$

Sensor located on Hyperbola

$$1 = \frac{\sqrt{(x - x_1)^2 + (y - y_1)^2}}{\Delta d} - \frac{\sqrt{(x - x_2)^2 + (y - y_2)^2}}{\Delta d}$$

TDoA



TDoA

Benefits of TDoA

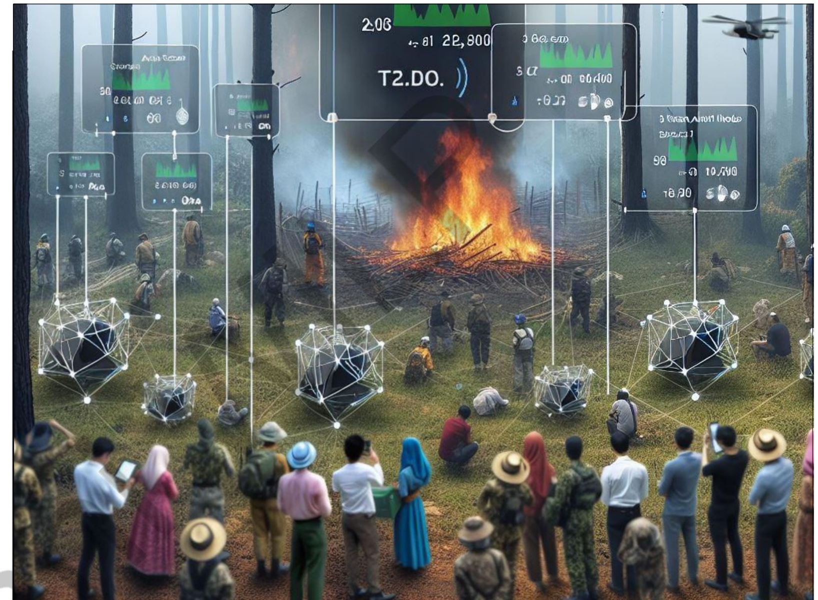
- Localised nodes act as beacons
- No additional beacons required

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

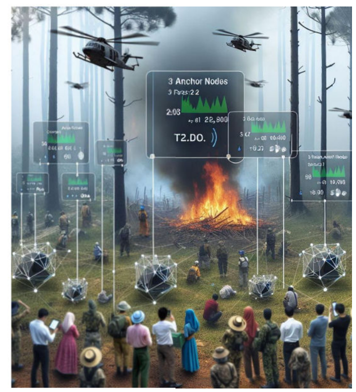
CASE STUDY - 1



innovate achieve lead

TDoA-Based Localization

- **Scenario Context:**
- A forest fire detection system uses WSN to locate a sensor triggered by smoke. Three synchronized anchor nodes record the Time Difference of Arrival (TDoA) of the sensor's signal.



innovate achieve lead