# Revision Mid 1 IIOT Content

## M2M vs IoT

- M2M focuses on interactions between machines and devices
- M2M is part of the IOT
- M2M standards have a prominent place in IoT standards landscape
- M2M requires no Human Intervention
- M2M focus on machine interactions via one or more tele com / communication networks like 4G
- IoT has broader scope as it focuses on communication between devices/things, things and people, things with applications, and people with applications.
- IoT includes notion of internet connectivity but does not focus on telecom networks only.

## CPS vs IoT

- CPS made up of Closed Feedback Loop: Sensing, Processing, actuation.
- CPS helps maintain state of environment
- CPS includes concept of digital twin, which is a digital model of the physical system. It is used to predict the behavior of actual system, and thus can help in working out correct commands for the physical system.
- CPS is a subdomain of IoT as IoT focuses also on networking and big data processing in edge and cloud along with Sensing and Actuation.
- IoT paradigm does not necessarily require a feedback **or digital twin system.**

## WoT vs IoT

- IoT is about creating a network of objects, things, people, systems and applications, WoT tries to integrate them to the Web
- Technically speaking, WoT can be thought as a flavour/option of an application layer added over the IoT's network layer
- The scope of IoT applications is broader and includes systems that are not accessible through the web (e.g., conventional WSN and RFID systems).

* Precision if given 0.7, do +- 0.35. If Resolution/Sensitivity given 0.2 do +- 0.2.
* In QoS-2, if you see from receiver side, message is received Exactly once. But if you see from sending side, messages are sent until they are received and receiving side ensures duplicates are discarded. In Short, From **Receiving Side Perspective**: **Exactly once message** is received, From **Sending Side Perspective:** Message is transmitted **At least once.**

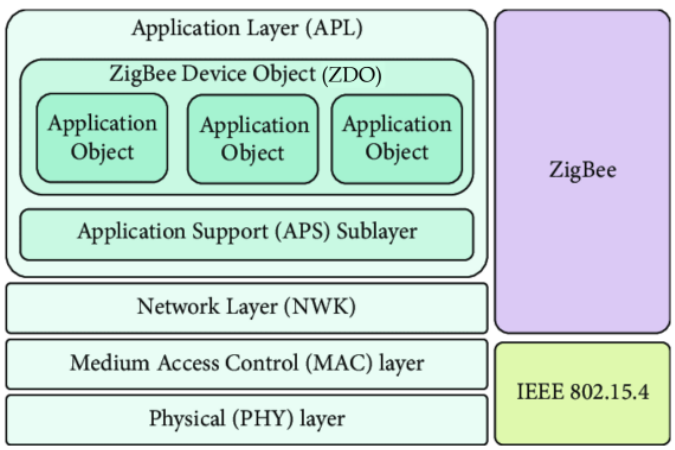
# IIOT MID 2 – THINGS TO MEMORIZE

IEEE 802.15.4

- Operates in unlicensed frequency bands 868MHZ – Europe, 915MHZ – Australia, North America, 2.4GHZ – Worldwide
- Infrequent, Short Packet Transmissions with Low duty cycle (<1%)
- Standard Transmission Range: 10m to 75m & up to 1000m outdoors.
- Data rate peaks at 250kbps
- CSMA/CA – Clear Channel Assessment, Back-off, increase back-off interval, ack, max retries
- Device Types – FFD, RFD

ZIGBEE

- Controller/Coordinator (ZC), Router (ZR), End Device (ZED)
- Routing: AODV
- Offers non-ip communication



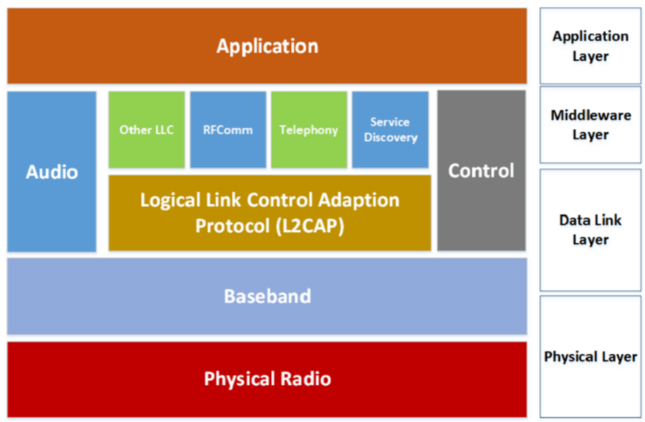

Z-wave

- RF for signaling and control.
- Operating frequency is 908.42 MHz in the US & 868.42 MHz in Europe.
- Data rates can vary depending on frequency, but max out at 100 Kbps.
- Controller (can also have in addition, secondary controller), slave nodes
- Mesh Network Topology – can Support 232 nodes in a network
- Addressing: Home ID (32 bits), Node ID (8 bits)
- Routing: Source Routing (Precomputed Route)
- Routing hops limited to 4 repeaters
- Healing messages

Bluetooth

- operates in the unlicensed, ISM band at 2.4 to 2.4835 GHz.
- 1 Mbps data rate for version 1.2 and 3 Mbps data rate
  for version 2.0 combined with 'Enhanced Data Rate' mode

- Class 3 - 1 meter or 3 feet, Class 2 – 10 meters or 30 feet, Class 1 100 meters or 300 feet
- Topology: Piconets
- Classic Bluetooth: 3 bits address, up to 7 non-parked slaves, 1 master node
- BLE Mode: 24-bit Address (each master-slave relationship is itself a piconet)
- Scatternets (Classic Bluetooth)
- Connection Establishment (Inquiry, paging/pairing, connected)
- Operating modes:  Active, Sniff, Hold, Park
- Frequency Hopping – Physical Layer, 79 Channels, 1600 hops/sec
- Each piconet has its own hop sequence (Chosen by Master Node)
- Any interference will last 625 microseconds – one hop period
- Adaptive Frequency hopping – Bluetooth 1.2 – Blacklisting



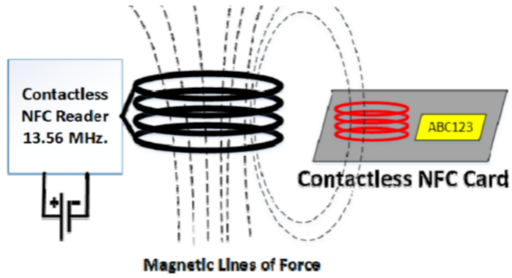

- Baseband (wireless modulation, frequency hopping, connection procedure (inquiry, paging), data whitening, logical links establishment, Error Detection/correction, Data whitening, C-res)
- L2CaP – in data link layer, multiplex multiple logical connections between two devices, segmentation, reassembling, Monitoring QOS (delay, loss, jitter), Group Management
- RFCOMM (binary data transport,60 simultaneous connections between 2 BT Devices)
- SDP (req-resp model)

RFID

- Derived from AIDC
- RFID system (RFID tag or smart label, an RFID reader, and an antenna)

NFC

- **Data-transmission** frequency is 13.56MHz (unlicensed globally).
- data at a rate of either 106, 212 or 424 kbps.
- Can store 96 and 512 bytes of data.
- Communication range is less than 20 cm
- Modes of Operation: Peer to Peer, Read/Write, Card Emulation




LoRa (Long Range)

- low-power wide-area networking technology
- Wirelessly connects battery operated devices to internet
- Sub-GHz unlicensed bands, e.g. 915 MHz in US, 868 MHz in Europe, 433 MHz in Asia
- Range: 5km in urban and up to 15km in rural areas (less without direct line of sight)
- Data rates: 0.3 to 50 kbps
- LoRa is the physical radio layer (modulates the data into radio waves)
- LoRaWAN is the communication protocol and system architecture (defines how devices use the LoRa hardware)
- LoRa is proprietary technique, while LoRaWAN protocols are openly published by a large non profit group, the LoRa Alliance.
- Topology: Stars of Stars (Gateway Nodes, End Devices, Central Network Server)
- Gateways - network server through a non-LoRaWAN network (IP over cellular or Ethernet)
- End Devices – Gateways : Single Wireless Hop (Gateways act as transparent bridge)
- LoRaWAN MAC Layer (ALOHA, device transmits as soon as it has data, may ask for ack)
- LoRaWAN (Facilitates Uplink [Device to Server])
- Three Classes Devices (for uplink all follow ALOHA, Downlink: A - short/fixed time wind to recv, B Can schedule additional recv wind, C listen always except when transmitting

6LoWPAN

- IP based communication
- All devices 64-bit hardware address (sometimes also called MAC address).
- IEE 802.15.4 Frame size 127 bytes, IPv6 requires 1280 Bytes
- Adaptation Layer in Data link layer introduced (Packet fragmentation and reassembly, Header compression, Mesh Networking)
- Edge Router (communication to the 6LoWPAN devices, relays data to internet, compression of IPv6 headers. 40-byte IPv6 header can compress to 2 to 20 bytes, Initiates the 6LoWPAN network, Exchanges data between devices on the 6LoWPAN network)
- Mesh Routing (Route-over (net lay using ipv6 add), Mesh- under(adp layer using link layer addr)
- Very first byte in each stacked header is 'dispatch byte' which identifies the type of header for receiving node (01 – iv6 addr, 10-Mesh,11- Frag)

None of the technologies (zigbee, Bluetooth, lorawan, 6lowpan) is capable of delivering large amounts of data like HD video. wired Ethernet or fiber optic connection would likely be preferred for transmitting high-definition video feeds reliably and with minimal latency

ROLL (Routing Over Low Power and Lossy Networks)

- Working group formed by IETF
- Specify Routing solutions for LLNs
- LLNS Routing Protocol Requirements (Support for unicast-multicast-anycast traffic, ability to dynamically and automatically recompute new paths due to change in conditions in the network, routes optimized for different metrics)

RPL (Routing Protocol for Low-power and lossy networks)

- MP2P, P2MP, P2P
- A DODAG is a DAG rooted in one destination Up-towards root or down – towards leaves
- Parent: where arrow is pointing towards (can be multiple)
- Child: where the arrow comes from (can be multiple)
- Three parents maximum are maintained by each node that provides a path to the root.
- Control Messages (DIO, DIS, DAO DAO-Ack)
- Routing Metrics (Hop count, latency, node energy consumption, Throughput, security (encryption), most common ETX – expected transmission count - average number of packet transmissions required to successfully transmit a packet)
- Objective function in routing metrics – Constraint
- Objective function combines constraints and routing metrics
- OF is used to compute rank during DODAG formation.
- Storing detailed children information requires more memory so RPL routing works in two modes: storing or non-storing.
- Storing mode: each node keeps a routing table containing mappings between all destinations reachable via its sub-DODAG and their respective next-hop nodes (learned via DAO)
- Non storing mode: root only maintaining routing information, exploits this global view for source routing, i.e., by including routing information directly into the packet itself.
- Scalable, Adaptive, Security (integrity protection, authentication, confidentiality, encryption, and proactive)

LOADng (Lightweight On-demand Ad-hoc Distance vector routing protocol –next generation)

- Reactive routing protocol: Routes are only discovered when there is data to forward.
- RREQs, RREPs - unicast hop-by-hop forwarding towards the originator.
- Receiving multiple RREQs allows intermediate nodes to learn the best path towards originator.
- Intermediate nodes also learn the path towards destination.
- RERR – Route Broken then sent to originator that will initiate route discovery (RREQ/RREP cycle) again.
- Optimized flooding of RREQs is supported (messages are rebroadcasted to a subset of neighbors), reducing the overhead incurred by generation and flooding.
- Only the destination is permitted to respond to an RREQ
- RREQ/RREP messages generated by a given router share a single unique, monotonically increasing sequence number.

IPV6

- 16 bytes of IPv6 address are represented as eight fields (groups) of hexadecimal digits, separated by colons. 1 hex=4 bits. Global, unique local, link local
- Payload in each fragment length must be multiple of 8, except the last one.
- In an IPv6 Packet, the payload includes everything after the base IPV6 header, Unfragmentable Header and Fragmentation Header 8byte - FH.

General Points

- In CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), the random wait time, often referred to as the "backoff time," is not predefined but is determined dynamically within a certain range. This process helps to reduce the chances of collisions when multiple devices attempt to transmit data over the same communication channel. each node picks its own random timer.
- Piconet is one master connected to up to seven slaves. Master controls the piconet. Scatternet is an interconnection of multiple piconets. A node joins two or more piconets and acts as a bridge for communication b/w piconets. That node can’t be master in both. In scatternet, each piconet continues to operate independently.
- LoRaWAN> Zigbee When long distance connectivity for sensor nodes is required (5km in urban-15 km in rural instead of outdoors 1000m best case or 10m-75m standard), data rates requirements are quite low (0.3 to 50 kps instead of 250Kps), and prolonged battery life is important.
- 6LoWPAN >> LoRaWAN When nodes are only required to communicate over **short distances**, within meters. When **IP connectivity of individual nodes is desired**. When slightly **higher data rates** are required, up to **250kbps.**
- In DODAG formation, order of DIO sent by nodes does not impact the final answer.
- Cluster-tree is less expensive to set up, while still providing the advantage of coverage over a larger area.
- Zigbee's mesh networking capabilities make it suitable for deployments with a high density of devices in close proximity.
- Zigbee topology could be star, cluster tree or mesh. There is a coordinator for whole PAN and several routers. End-nodes only connect to their parent. Zwave is always a mesh. There are one or more controllers, and other nodes are slaves. Slaves forward the data in mesh but do not make routing decisions.
- Class A is to be used when a longer battery life is desired, but downlink latency is tolerable. Class C devices are always listening, so these used when a real time action is expected (in response to a command from the server).
- Topologies:
    - Zigbee: Mesh, Cluster Tree, Star
    - Z-Wave: Mesh
    - 6LoWPAN: Mesh
    - Bluetooth: Piconets, Scatter Nets
    - LoRaWAN: Stars of Stars (End Devices, Gateway, Network Server)

- Routing:
  **- Zigbee:**
    - Mesh: Utilizes the **Ad hoc On-Demand Distance Vector (AODV)** routing protocol.
    - Cluster Tree: **Uses hierarchical routing**, where communication is often centralized through parent-child relationships.
    - Star: Typically involves **direct communication** between end devices and the central coordinator (no specific routing protocol).

  **- Z-Wave:**
    - Mesh: Uses a **source routing algorithm**, where the route is determined by the source device and each device knows the entire network topology.

  **- 6LoWPAN:**
    - Mesh: Typically uses the **Routing Protocol for Low-Power and Lossy Networks (RPL),** which is designed for IPv6 networking in low-power environments.

  **- Bluetooth:**
    - Piconets: No specific routing protocol, as communication is **directly managed by the master device.**
    - Scatternets: Routing is more complex and not standardized within the Bluetooth specification. It relies on custom implementations to manage data forwarding between piconets.

  **- LoRaWAN:**
    - Star-of-Stars: Communication is generally **direct between end devices and gateways (star topology).** There is **no multi-hop routing, as gateways forward messages to a central network server using IP-based backhaul.**

In the context of DODAG formation in RPL routing, DIO is normally broadcasted. Is there any situation when it can be sent as a unicast? Can a node be part of 2 DODAGs in the same RPL Instance?
Answer: DIO is sent as unicast in response to DIS message from a node who wants to join DODAG.
No, it cannot be part of 2 DODAGs in the same RPL instance.

# HELP SHEET IIOT POST MIDS

**Raspberry PI**

- RPi - much more powerful & faster CPU (**multi-core 1.4 GHz** on Model 3B vs **single core 16 MHz** on Arduino Uno)
- Processers are **64 bit** (earlier 32 bit). Arduino models are generally **8 or 16 bits.**
- RPi has a larger memory – starting at **256 MB**, Arduino typically **less than 10KB.**
- Runs an **OS** (UI, Multiple Processes, Low Level & Hardware Features Abstraction)
- No permanent storage (HDD / SSD / flash) is included by default. Instead, the OS is stored on an **SD card.**
- needs to **plugged in to a monitor** for display and **a keyboard/mouse** for input.
- Used in **IOT Applications** as **small & relatively cheap**, **on-board processing ability**, **can directly interface with sensors & actuators** & **allows network connectivity.**
- **Applications:** home automation, security camera systems, smart TV, AI assistant, media server, robotics, FM radio station etc.
- Pin Numbering Modes: Ordered **on-board** or **Broadcom SoC** pin numbers.
  ```
  mode = GPIO.BOARD or GPIO.BCM
  GPIO.setmode(mode)
  ```
- **Set pin I/O direction**: **GPIO.setup(13, GPIO.IN) or GPIO.setup(13, GPIO.OUT)**
- **read/write operation**: GPIO.output(13, GPIO.HIGH) or GPIO.input(17)
- Only a single pin **(board #12)** on RPi is capable of creating **PWM output.**
  ```
  pwm = GPIO.PWM(12,1000)    # create an object of PWM class, freq = 1KHz
  pwm.start(50)              # start generating PWM signal of duty cycle 50%
  pwm.ChangeDutyCycle(75)    # To adjust the duty cycle
  pwm.stop()                 # To turn off pwm output
  ```
- At end of program, **reset all pins back to input mode** using **GPIO.cleanup()** call to avoid accidental damage to pins.
  ```
  import RPi.GPIO as GPIO
  def main(): #program goes here
  try: main()
  except: print('an exception occurred')
  finally: GPIO.cleanup()
  ```
- Execute program **as root**. **OS** considers **all GPIO pin access** as **sensitive action**, requires **superuser privileges**.

PI HEADLESS MODE

- Headless means without a screen connected
- Install **remote login** or **remote desktop server apps**, Connect to PI using **client apps (e.g TigerVNC – Graphical or SSH – text based)** on another machine
- ssh <username>@<server ip> [To connect via SSH]
- All **GPIO pins** on RPi operate at **3.3 Volts** (not 5V, its only to supply power)

- Pi can receive digital data that uses **5V** as logic high using **Voltage Divider Concept**
- **Vout = (R2/(R1+R2)) * Vin**
- when R1 = 1kΩ, R2 = 2kΩ, Vout = 3.33V
- Ultrasonic Distance Sensor (Speed of sound in air is **343 m/s**)

  from **gpiozero** import **DistanceSensor**
  ultrasonic = DistanceSensor(echo=17, trigger=4) # BCM pin numbers
  while True: print(ultrasonic.distance)
- GPIO pins on Pi **cannot 'drive' a motor**, need **external 'motor driver' module**
- **DC motor** can spin **forwards or backwards** depending on **direction of current** flowing through it. We use **H bridge** to reverse (Motor drivers include circuit looks like H).

  **Vss pin -** needs **3.3-5V for internal circuit** of L293D.
  **Vs pin - motor power supply 4.5V to 36V**.
  **Enable pins** - set high for enabling the 1st and 2nd H bridge
  **Input pins** - connect to Pi, **output pins –** motor

  **Motor speed is controlled via a PWM signal.**

  from gpiozero import Motor
  motor1 = Motor(forward=24, backward=23, enable=25) # instantiate a Motor object, specifying **BCM** pin numbers
  **motor1.forward()** # start motor spinning at full speed or **motor1.forward(0.5)** # or set a slower speed (float 0 to 1) or **motor1.backward(0.7**) # spin backwards at 70% speed
  motor1.reverse() # change spinning direction
  motor1.stop() # stop spinning

PI CAMERA MODULES

- Any USB camera can be used but also dedicated port for camera connections, CSI (Camera Serial Interface) - direct and faster interface to camera modules

  from picamera import PiCamera
  camera = PiCamera()

  ***TO CAPTURE AN IMAGE***

  camera.hflip = True | camera.vflip = True # Adjust camera settings if needed
  camera.brighness = 50 | camera.resolution = (1024, 768)
  camera.capture('hello.jpg') # Take a picture

  ***TO MAKE A 10 SECOND RECORDING***

  camera.start_recording('myvideo.mp4', format='h264')
  time.sleep(10) # make a 10-second video
  camera.stop_recording()

  ***Storing captures directly to a server use standard sockets API***

  import socket
  mysocket = socket.socket() # Open a client socket
  mysocket.connect(('<SERVER_ADDRESS>', 8000))
  conn = mysocket.makefile('wb') # Get a file-like object for this socket
  camera.capture(conn, format='jpeg') # Pass this object as capture destination

## TCP FLOW AND CONGESTION CONTROL

- The **receiver window size (rwnd) is a flow control** mechanism that ensures the sender does not overflow the receiver's buffer. This information is communicated from the receiver to the sender using the TCP header.
- The **congestion window size (cwnd) is a congestion control** mechanism that ensures the network is not overwhelmed by too much traffic. It is managed by the sender and is not explicitly communicated in the TCP headers. TCP sender limits transmission: **lastByteSent – lastByteAcked <= CWND**. cwnd is **dynamically adjusted** in response to observed network congestion (implementing TCP congestion control). AIMD

## TCP in IoT –challenges & adaptations

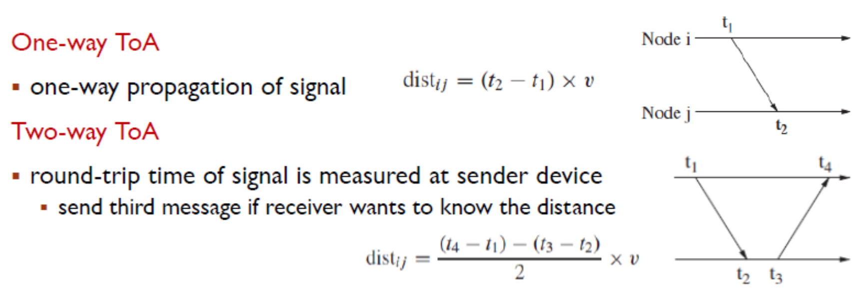
1. **memory-constrained** TCPs cannot implement the sliding window mechanism so Keep sliding window of size 1
2. TCP delayed ACKs (widely applied by TCP) **reduce the throughput** so turn it off
3. three-way handshake for connection setup can become a **significant overhead** for IoT nodes so Opt for **long-lived TCP connections** but **Firewalls** and **middleboxes** might not allow such long-life connections. In such a case, utilize **TCP Fast Open (TFO)** feature.

## WIRELESS SENSOR NETWORKS

- Wireless network consisting of **spatially distributed autonomous devices** using **sensors** to **cooperatively monitor physical or environmental** conditions at different locations.
- Wireless sensors are limited in memory, computation power, bandwidth, and energy.
- Deployment is usually ad-hoc so lack of maintenance and battery replacement
- **Acquired information is wirelessly communicated to a base station (BS), which propagates** the information to **remote devices for storage, analysis, and processing**
- Data Dissemination **(flat, hierarchical [Clustering. Cluster Head]**
- Types of **location:**
    1. **Global Position:** Position within general global reference frames i.e. latitudes and longitudes
    2. **Relative Position**: based on arbitrary coordinate systems and reference frames.
- **Range Based** Localization (**Distance or Angle to Anchor nodes**), **Range-Free** Localization (**Connectivity to anchor nodes**)
- **Ranging Techniques:**

**1)** ToA (Time of Arrival)

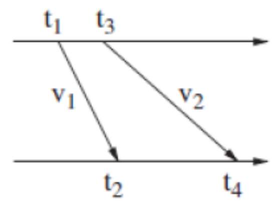
sound waves: 343m/s, radio signals: 300km/s. highly accurate synchronization



**2)**

## Time Difference of Arrival (TDoA)

- two signals with different velocities
  - e.g. radio signal (sent at $t_1$ and received at $t_2$), followed by acoustic signal (sent at $t_3 = t_1 + twai_t$ and received at $t_4$)
- no clock synchronization required
- distance measurements can be very accurate
- need for additional hardware




$$\text{dist} = (v_1 - v_2) \times (t_4 - t_2 - t_{\text{wait}})$$

**3)**

## Received Signal Strength (RSS)

- radio signal decays with distance
- In free space, RSS degrades with square of distance, expressed by Friis transmission equation ($G_t$ and $G_r$ are transmitter and receiver antenna gains respectively)

$$\frac{P_r}{P_t} = G_t G_r \frac{\lambda^2}{(4\pi)^2 R^2}$$

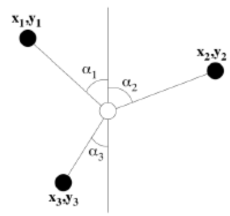
- In practice, the actual attenuation depends on multipath propagation effects, reflections, noise, etc.
- realistic models replace $R^2$ with $R^n$ ($n = 3 \ldots 5$)

## Angle of Arrival (AoA)

- finds out direction of signal propagation
- measures angle between signal and some reference orientation
- can be achieved using:
  - a continuously rotating receiving element. The direction in which highest signal is received is the desired angle.
  - an array of antennas or microphones.
    - spatial separation of antennas or microphones leads to differences in arrival times, amplitudes, and phases.
    - accuracy can be high (within a few degrees)
- adds significant hardware cost

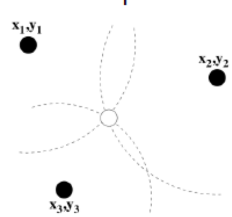
# Triangulation

- Example of range-based localization
- Uses the geometric properties of triangles to estimate location
- Relies on angle (bearing) measurements
- Minimum of two bearing lines (and the locations of anchor node) are needed for two-dimensional space



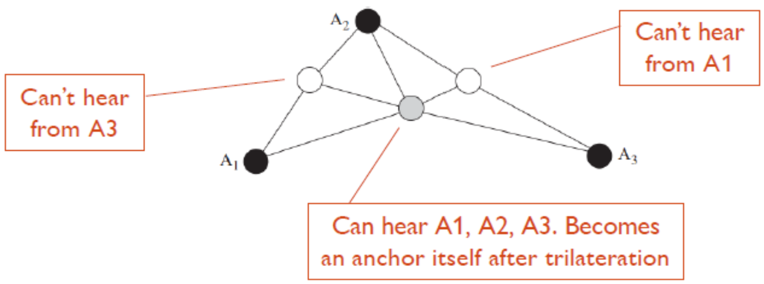

# Trilateration

- Another example of range-based localization
- Calculate a node's position based on measured distances between itself and a number of anchor points with known locations.
  - It is known that the sensor must be positioned somewhere along the circumference of a circle centered at the anchor's position with a radius equal to the sensor–anchor distance.
- In two-dimensional space, distance measurements from at least three noncollinear anchors are required to obtain a unique location (i.e., the intersection of three circles)



# Iterative Multilateration

- Problem: what if node does not have at least three neighboring anchors?
- Solution: once a node has determined its position, it becomes an anchor
- Iterative multilateration repeats until all nodes have been localized.
- Challenge: error accumulates with each iteration




**Range-Free Localization Approach**

- **1) Ad-hoc Positioning System (APS):**
  In an ad-hoc positioning system, when you have two hop counts between the same nodes (min and max), you typically use the minimum hop count. This is because the minimum hop count represents the shortest path or the least number of intermediate nodes required to reach the destination, which usually translates to the most efficient route in terms of latency and resource usage. Using the minimum hop count helps in maintaining efficient and effective routing within the network.
- **2) Approximate Point in Triangulation (PIT Test): Area Based.** a node M is outside a triangle formed by anchors A, B, and C if there exists a direction such that a point adjacent to M is either further or closer to all three points **simultaneously**; otherwise M is inside. signal strength measurements can be used to determine if a node is closer to an anchor or further away. In dense networks, node movement can be emulated using neighbor information (exchanged via beacons)

General Points:

- By default, TCP sends out N segments immediately, and then starts waiting. N is defined by congestion-window or receive-window, whichever is smaller. Once receives gets a cumulative ACK of m segments (m < N), it sends out another m segments. So effectively, N packets are always in-flight (sent, waiting for acknowledgement). Purpose of this mechanism is to increase the throughput as high as possible.

- IoT devices do not have enough memory to maintain a window of size N. Also, throughput is not a concern in many IoT applications. Therefore, the sender uses a window size of 1, which makes it a stop-and-wait methodology.
- WSN different from regular networks:

**Energy Efficiency and Power Management**

Battery-Powered Nodes: Nodes are often battery-powered, necessitating energy efficiency.

Low Power Consumption: Designed to minimize power consumption through efficient protocols and duty cycling.

Energy Harvesting: Some nodes use energy harvesting techniques to extend operational life.

**Self-Organization and Scalability**

Ad Hoc Deployment: Nodes self-organize without pre-existing infrastructure.

Dynamic Topology: Network topology can change due to node mobility or environmental factors.

Scalability: Capable of handling large numbers of nodes efficiently.

- How and for how long a **memory is allocated** for a piece of program determines **the speed of task execution.**
- When a command function call is made, the callee creates a task (that will be scheduled by OS sometime later) and returns immediately. Later when the task is complete, the callee signals an event for the caller.
- A machine operating without a display screen is called headless. Pi can be setup can headless by installing remote login (ssh) or remote desktop (vnc) server software.
- Pi Pico is not a single-board computer means "Pico contains a much **weaker microcontroller** with **limited computation and memory –** not enough to run an operating system."
- Settings a small MSS can help in IoT because smaller segments imply **less memory required at both sender and receiver.** Some IoT applications have a limited MTU at the link layer. (e.g. IEEE 802.15.4). For such cases, small MSS means **packet fragmentation won't be needed by layer 3.**
- Correction Factor of Closest Anchor IS USED.
- A new three-way handshake for connection setup can become a significant overhead for IoT nodes, e.g. a sensor sending periodic data might need to go through connection setup before every transmission. IoT devices can opt for long-lived TCP connections. If that is not possible, they may utilize TCP Fast Open feature in which the sensor node sends the data within the SYN packet.
- Sensor networks often monitor critical infrastructure or carry sensitive information, making them desirable targets for attacks. Also, Conventional security techniques are often not feasible due to their **computational, communication, and storage requirements**.