1. What are the typical components of a sensor node?
   1. **Controller**   
      Processing of relevant data, executing arbitrary code)
   2. **Watchdog**:   
      Sensor controlled/time controlled/communication-controlled wakeup
   3. **Wireless radio**

Short Range, low bit rate

* 1. **Sensor**Thermometer, cameras light, sensors, etc
  2. **Power Supply**  
     batter, solar, vibration

1. **Which component of a sensor node have a major influence on its autonomous lifetime?**Wireless Radio: Data Exchange requires a lot of energy

Power supply: Bigger supply -> Longer operation time

1. **What are the key differences between a wireless ad-hoc network and a conventional wired network?**

Wireless ad-hoc networks do not rely on a physical connection while wired networks do.

Wireless ad-hoc networks are decentralized where devices act both as router and end node.

Wireless ad-hoc network nodes can move freely within the network range

Wireless ad-hoc networks are less reliable because of the moving nodes

**Wireless Ad-hoc Network**

Physical Infrastructure: Wireless  
Topology: Decentralized/mesh  
Connectivity: Mobile  
Reliability: Less reliable

**Conventional Wired Network**

Physical Infrastructure: Cables or wires  
Topology: Centralized  
Connectivity: Stationary  
Reliability: More Reliable

1. **How does the power of a radio signal at the location of a transmitter relate to the power of the radio signal the location of the receiver?**The power mist be equal to the number of hope times the distance to each device consideration of the signal attenuation
2. **Outline why multi-hop wireless communication is more energy- efficient than single -hop wireless communication**The Single Hop communication only uses two device and thus need more energy the further away the receiver is. The multi-hop always required less energy because the signal is transmitted by a chain of transmissions. The advantage can be calculated by Nα−1  
     
     
     
   **Personal**

The more intermediate nodes there are in a network, the less energy is required to forward a packet to its destination. In other words, as the network becomes denser, the energy needed for packet transmission decreases. Therefore, multi-hop communication, which involves multiple intermediate nodes, requires less overall energy compared to single-hop communication when transferring data.

1. **What is meant by the term “Data centricity” in relation to sensor networks**Its not about the identity of the node but what data it can provide is the concept of data centricity. Neither sender nor receiver needs to know about their partner. The answer is not necessarily triggered directly by a question.
2. **What is the typical networking interaction paradigm in wireless sensor networks? Describe this concept!  
   Personal**   
   The typical networking interaction paradigm in wireless sensor networks is the publish/subscribe paradigm. In this model, publishers send data to a central software bus, which stores subscriptions from various subscribers. When the data changes, the software bus notifies the relevant subscribers based on their subscriptions, enabling efficient and decoupled communication between publishers and subscribers.
3. **What is meant by the term “Partial State Record”**A partial state record refers to the intermediate result of an aggregation process, where essential statistics such as the sum and count of previously aggregated values are maintained. This record is crucial for computing aggregates like averages, as it tracks the ongoing summation and the number of values included.   
   **The behaviour of partial state records varies**: they can be distributive, maintaining the same size as the final aggregate   
   *(e.g., MIN, MAX, SUM);*   
   algebraic, having constant size but not being full aggregates themselves   
   *(e.g., AVERAGE);*  
   holistic, requiring the retention of all measured values   
   *(e.g., MEDIAN);*   
   or content-sensitive, proportional to some data properties   
   *(e.g., HISTOGRAM).*   
   Aggregation functions with distributive and algebraic partial states are suitable for in-network aggregation, whereas content-sensitive and holistic functions present challenges for practical aggregation.
4. **What are typical properties of aggregation function in relation to the unreliable character of wireless communication?**Aggregation functions  
   Given two partial state records <x> and <y> and aggregation function f computes a new state record <z> = f(<x>,<y>).

1. Duplicate sensitive aggregates are affected if the measured values of a node is used in computation more than once  
2. Exemplary aggregates return one or more representative values from the set of all values  
3. Summary aggregates compute some properties over all values

1. **Why is the sequential programming model unsuitable for programming sensor node?**In the case of sequential programming model, the sensor is being polled for available data**.** Also, the transceiver is being polled for any packets arrived or packets that must be transmitted. There are couple of risks with this programming model.  
   1. We can miss some data from the sensors while a packet is received or processed  
   2. We can miss a packet when the sensor information is being processed  
     
   Draw this in the exam   
   A diagram of a process flow

   Description automatically generated
2. **Why is the process-based programming model unsuitable for programming sensor nodes?**In case of process-based programming model, we have two separate processes or threads;  
   One for handling sensor.  
   Second is for the packets  
   This produces too much overhead because the following reasons:  
   1. A lot of memory i required, since each process requires its own stack space in the memory  
   2. The context switch between the processes has a high rate.  
   Draw this in the exam  
   A diagram of a process

   Description automatically generated
3. **Which programming model is suitable for sensor? Describe this programming paradigm!**The programming model suitable for sensor nodes is the event-based programming, the reason, in this case, a sensor node waits for any event to happen. This can be, for example, a timeout event, available data from a sensor, or an arrival of a packet. Once the event has occurred, it is then handled by a short sequence of instruction.  
     
   **Some Deep explanation**:  
   How it works  
    1. Types of Events:  
    Timeout Events: Triggered after a specified time interval  
    Sensor data availability: New data from a sensor becomes available  
    Packet arrival: A data packet arrives from another node  
    2. Event Handling:   
   i.When an event occurs, the sensor node transitions from idle or regular   
    processing to an event-specific handler  
    ii. Sensor Event Handler: Activated when a sensor-related event occurs,   
    such as new data availability.  
    iii. Radio Event Handler: Activated when a communication-related event   
    occurs, like the arrival of a data packet.  
    3. Instruction Execution:   
    i. Upon activation by an event, the node executes a short sequence of  
    instructions tailored to handle that specific event
4. **TinyOS applications are written in nesC. nesC uses two types of components. What functions do “ modules have and what functions do “Configurations” have in nesC application?**The nesC implementation consists of two parts:  
    A PowerupC modul and a PowerupApp C configuration  
    The module PowerupC contains the executable logic of the application  
   The module resides in the file Powerup.nc  
      
     
    **Modul Powerup C**{  
    uses interface Boot;  
    uses interface Leds;  
    } **implementation**{  
    event void Boot.booted(){  
    call Leds.led0On();  
    }  
    }  
     
    **Configuration** PowerupAppC {  
      
    }  
    **implementation**{  
    components MainC, LedsC, PowerupC;  
    PowerupC.Boot -> MainC.Boot;  
    PowerupC.Leds -> LedsC.Leds;  
    }  
   Writing the code in the exam would give us points or?
5. **What does the term “wiring” mean in relation to nesC application?**The term “wiring” means binding an interface user to an interface provider. In other words, configuration consists of a signature and an implementation block. The implementation block can bind an interface user to a provider using  
    the -> or <- operator  
    This process is called wiring  
     
   Example is given in question 13
6. **The functional relationship between different nesC components is defined by interfaces. Which functions are implemented by interface providers and which functions are implemented by interface users?**  
   Personal  
   In nesC programming, the functional relationship between components is defined by interfaces, where interface users and providers have distinct roles. Interface users can call commands provided by the interface and must implement the events signaled by the providers. On the other hand, interface providers are responsible for implementing the commands that users can call and can signal events that the users must handle.   
     
   This clear division of responsibilities ensures structured and modular interaction between different components in nesC, enhancing the overall functionality and maintainability of the system.
7. **Many TinyOS interface are “Split phase”. What does this term mean?**This means that the user of the interface sends a command  
   The provider of the interface performs the associated operation  
   The provider signals an event that indicates the completion of the operations  
   Draw this in the exam  
   A diagram of a user interface

   Description automatically generated
8. **The execution model of TinyOS uses “tasks”. Which special functions do these tasks have?** Tasks are very particular functions; they are an important building block for the execution model. Tasks look like C functions, but:  
    1. They can’t have a return value  
    2. They don’t take any parameters  
    3. They can be executed only within the naming scope of a component  
   4. Tasks schedule a function to be called later  
    5. Taks do not interrupt one another  
    6. Tasks can be seen as time-flexible background jobs
9. **Describe the execution model implemented by TinyOS! How does Tiny OS’s schedular work?** 1. Split-Phase Interface  
    Operations like sending packets or sampling sensor are handled in two   
    phases:  
    1. Command initiation: A non-blocking command starts the   
    operations and immediately returns.  
    2. Interrupt Handling: The hardware component issues an   
    interrupt when the operations is complete  
    3. An event handler processes the completion and can post a   
    new tasks  
    2. Execution Model:  
    1. Energy saving: The mote stays asleep most of the time to save   
    energy  
    2. Computation Start: Computation is initiated by the hardware   
    interrupts. (The post operations adds tasks to an internal task   
    queue)  
    3. Schedular Operations:  
    1. Tasks Queue Processing: The scheduler processes tasks in   
    First In, First Out (FIFO)  
    Interrupts can interrupt tasks and post additional tasks

**Chapter 3  
  
Personal**

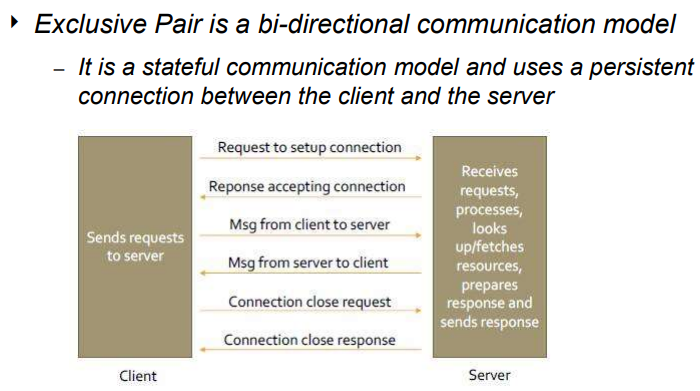
1. Describe the 5-Layer model of the IOT  
   There is no single consensus on architecture for IoT, but we can have a starting point.
   1. Client/External Communication Layer
      1. This layer handles the interaction between the IoT system and the external entities. Provides web protocols, dashboards, and user interfaces / web interface to communicate with the other IoT systems
   2. Event Processing and analytics layer
      1. As each IoT devices generate a lot of that, so this layer manages and processes the vast amounts of data generated by IoT devices. Functions of this layer are that it uses techniques to analyse data in real-time, employing pattern matching algorithms to detect significant events (e.g., identifying dangerous individuals at an airport)
   3. Aggregation/Bus layer
      1. Facilitates communication between IoT devices and protocols. Functions of this layer is to translate between different protocols, provides security processes like authorization and verification to ensure data stream integrity for a specific data is accessed from specific data source.
   4. Communication Layer
      1. Deals with the protocols and methods for data transmission between devices. Function of this layer is to support various communication method. For example: Radio modules, Bluetooth, long-range modules) and includes protocols like MQTT and CoAP to enable efficient data exchange
   5. Device layer
      1. Comprises the actual IoT devices, which can include various sensors and actuators. Function of this layer is to act as an embedded web server, accessible through HTTP polling or more efficient method like WebSockets for real-time, bidirectional communication
   6. Additional Management blocks: Device, Identity, and access management
      1. **Device Management:** Deploys software, manages version control, automates updates, monitor device state and availability, and can block or wipe compromised devices
      2. **Identity and Access Management**: Implements authentication and authorization to ensure secure data transfer and function access between devices
2. **What are the characteristics of the “request-response” communication model?**Its a stateless communication model. Each request-response pair is independent of each other   
   A diagram of a server

   Description automatically generated  
   Clients sends a request to a sever, and the server responds to the request, now when the server receive the request, it decided how to respond fetches the required data from the database, prepares the response in an appropriate represents like in JSON and sends the response to the client. As this a stateless mode of communication so, so all the requests are independent to each other  
     
   Example: HTTP
3. **What are the characteristics of the “publish-subscribe” communication model**A diagram of a subscriber

   Description automatically generated

The Publish-Subscribe communication model involves three main instances: **publishers, consumers, and brokers**, it is suitable in event driven applications  
Publishers act as sources of data, sending messages to specific sever called the broker, the broker handles the routing distributions of the message, to the various subscribers/consumers.   
Consumers subscribe to these topics to receive relevant data.   
Brokers manage the topics and facilitate the forwarding of messages from publishers to the appropriate consumers.   
This model decouples the producers and consumers of data, enhancing scalability and flexibility. For example, a publisher can send messages to topics such as Topic 1 and Topic 2, and brokers ensure these messages are delivered to all consumers subscribed to those topics, allowing Consumer 1, Consumer 2, and Consumer 3 to receive only the data they are interested in.  
  
Example MQTT

1. **What are the characteristics of the “push-pull” communication model?**A diagram of a company

   Description automatically generatedThe "Push-Pull" communication model uses queues to decouple messaging between producers and consumers. Producers send messages to queues, which act as buffers to handle inconsistent data rates between producers and consumers. This decoupling allows producers and consumers to operate independently, with queues managing the flow of messages to ensure consumers can process them at their own pace. This model enhances flexibility and scalability, as it allows the system to handle varying speeds of message production and consumption effectively.  
     
   This is many used to have consistency with data rate between Publisher and Consumers
2. **What are the characteristics of the “exclusive pair” communication model?**The "Exclusive Pair" communication model is a bi-directional and stateful communication method that maintains a persistent connection between a client and a server. This model allows both the client and server to continuously exchange messages over the same connection until it is explicitly closed. It begins with the client requesting to set up the connection, followed by the server's response accepting it. Messages are then sent back and forth, and the connection is closed upon request and confirmation from both sides. This approach ensures efficient and continuous communication without the need to repeatedly establish new connections.
3. **What is the purpose of the 6LoWPAN network layer protocol?**The purpose of the 6LoWPAN network layer protocol is to enable IPv6 to be used with wireless embedded devices. Since Zigbee does not natively support Internet protocols, 6LoWPAN allows these devices to communicate using the standard Internet Protocol, thereby facilitating the integration of wireless embedded devices into IP-based networks. ADD MORE INFO FROM YOUR OWN NOTES
4. **How does the network protocol COAP work in principle?**COAP is a stateless request/response protocol. However, a separate protocol extension is used to allow observation of resources. Lile in the case of MQTT, a client can subscribe to a resource. COAP is usually used while we use 6LoWPAN, uses UDP and doesn’t maintain an open TCP connection. Like http, COAP also supports GET, POST, PUT and DELETE methods
5. **How does the network protocol MQTT work in principle**MQTT works on the publish-subscribe communication model where a client connects to a broker, which acknowledges the connection and keeps it alive as long as possible. The client subscribes to specific topics on the broker, and the broker manages message distribution to all subscribed clients. The MQTT protocol allows clients to send regular ping requests to maintain the connection and uses "last will and testament" messages to notify others when a client disconnects. Implemented via TLS for security, MQTT brokers can request authentication details like username and password. This model negates the need for constant polling, enhancing efficiency.
6. **What role do "topics" have in the MQTT protocol?**A topic is a UTF-8 string, which is used by the broker to filter messages. Subscribers are often interested in a great number of topics. Individual subscribing to each topic would be cumbersome (Wild-cards)
7. **What role do the symbols "\*" and "#" have in relation to MQTT topics?**In MQTT topics, the symbols "+" and "#" serve as wildcards to simplify subscription to multiple topic levels. The "+" symbol is a single-level wildcard that matches one topic level. For example, "building/+/sensors" can match "building/floor-1/sensors" and "building/floor-2/sensors". The "#" symbol is a multi-level wildcard that matches multiple topic levels, such as "building/floor-1/#", which can match all sub-topics under "building/floor-1", like "building/floor-1/sensors" and "building/floor-1/blinds". These wildcards make it easier to subscribe to a broad range of topics without specifying each one individually.

Questions for the Prof.

1. For some question, with theory if we create a diagram, can we cover points from some missing things that were not written in our answer (Like missing keywords)

2. How much should we prepare from the Lab ex.