Chapter 3 : (Network architecture)

This chapter deals with the principles by which the sensor nodes are formed into wireless sensor network. This chapter focuses on the more concrete wireless scenarios along with the optimization goals. Also the networking protocols in WSNs are derived.

Sensor network scenarios :

Types of sources and sinks: Sources are the entity of network that are used to provide information and sinks are the entity where where information is required.

→ Sink has three options: it may be one sensor node in the wireless network or it can be a sensor node outside of the network. The later case is an example of PDA that used to interact with sensor network. So the last option the sink could be a gateway to another large network (Internet) to which actual information comes from 'far away' and indirectly connected to the network (Internet)

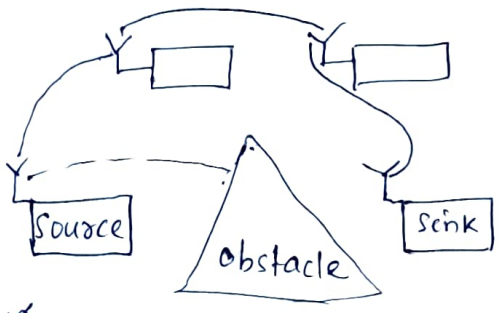
Single-hop versus multihop networks

Due to power limitation of radio comm^n. there is a distance limitation bet^n sender and receiver. Therefore, direct comm^n bet^n source and sink is not always feasible as WSN cover large area with strong attenuation (Buildings).

→ To overcome this data packets take multiple hops from source to sink.

→ Multihopping overcome problems with large distances or obstacles. It also improve energy efficiency of communication.

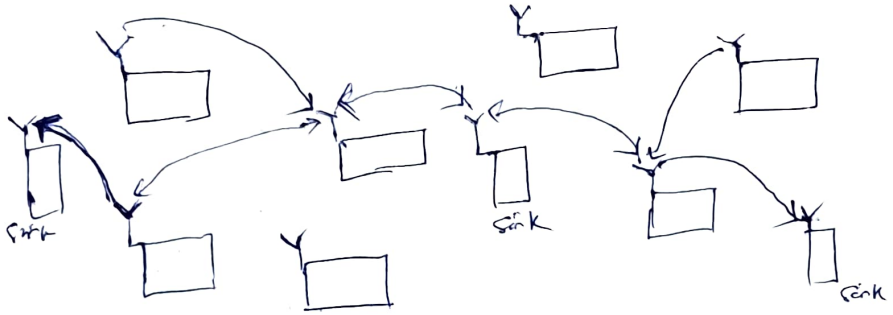
→ The radiated energy for direct communication over distance d is cd^{α} (c is constant, $\alpha \geq 2$ pathloss coefficient), using a relay at distance $d/2$ reduces energy to $2c(d/2)^{\alpha}$. However, by considering the energy consumed at relay node

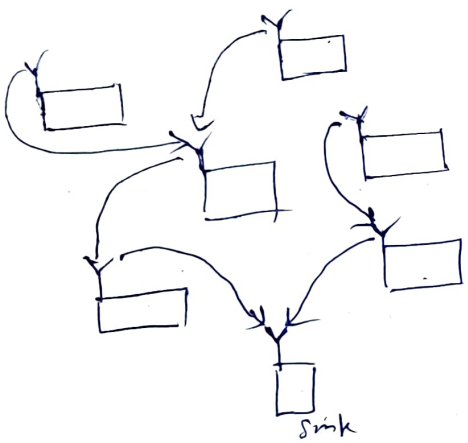


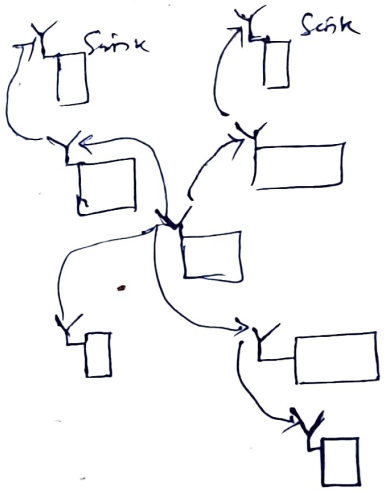
it is observed that energy is wasted if intermediate relays are used for short distance d.

→ The multihop networks are considered to be operated in stored and forward fashion. In this network node have to correctly receive packets before forwarding.

→ Another network type uses (cooperative relaying) where multiple nodes send same packet but all the transmission are not successful, but a node can collectively reconstruct the true packet.







Multiple Sinks and Sources:

Here multiple sources send information to multiple sinks, where either all or some of the information has to reach all or some of the sinks.

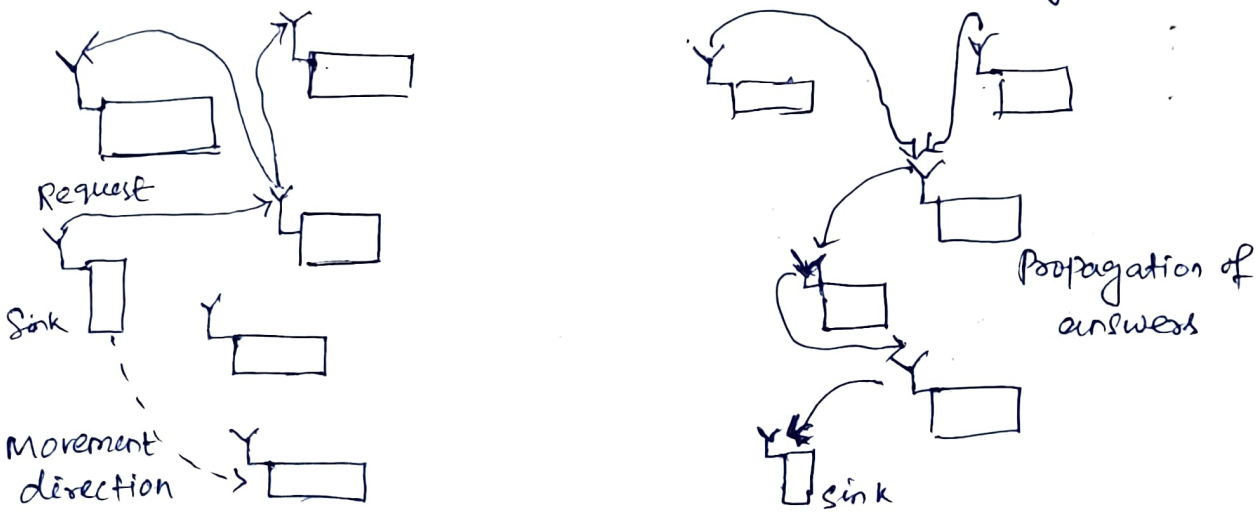
Types of mobility: WSNs has the ability to support mobile nodes. This mobility takes the following forms:

Node mobility: Here the sensor nodes are mobile. Scenarios such as environmental monitoring doesnot require mobile nodes. However, livestock surveillance (sensor nodes attached to cattle) require mobile nodes.

→ During node mobility frequent reorganization of network is required for proper functioning. A trade-off bet?. speed and frequency

node movement cost energy requirement for proper network function is used.

Sink mobility: Here the sinks are mobile. They can be considered as a special case of node mobility. The sink node mobility can be observed without the network part. for example a human user request for information through PDA while walking in an intelligent building.



Sink node before movement interact with WSN at one point and complete the interaction before moving on. Consecutive interactions are dealt as separate and unrelated requests.

→ While moving, a mobile requester may not have data available locally but the data can be retrieved from remote part of n/wk. The mobile sink basically communicate with nodes in its vicinity, therefore during movement the network allows the requested data to follow the mobile requester and reach to it despite of its movement.

Event mobility: In this type of mobility, the event or cause of event is mobile. Such examples are tracking events. In this scenario the target is surrounded by sufficient sensor nodes for all time. The sensors around the object wakeup (when object is in the sensing zone), detect the presence of object and then go to sleep mode.

→ Due to movement of object in the network, it is associated with an area of activity and is referred as frisbee model. For example considering the movement of an elephant. Nodes that are not detecting activity are switched to lower sleep states.

till they convey information from activity zone to remote sink.

→ To address such mobility communication protocols for WSN is designed.

Optimization goals and figures of merit:

Even if networking solutions are available for all the WSN scenarios and applications, the major challange is to optimize a network. The decision need to be made which approach better supports a particular application. The metrics for optimization goals are:

A. Quality of service:

These are low level networking-device-observable parameters such as bandwidth, delay, jitter, packet loss rate and high-level subjective or user observable attribute like quality of voice or video communication. High-level QoS attributes in conventional WSNs are required. However, high-level QoS attributes in WSN highly depend on certeen applications that are:

Event detection / reporting Probability: It indicates the probability of an event occured is not detected or reported to information sink. For example non reporting of fire alarm to a surveillance station.

→ This Probability can be traded off against routing tables (it supports reporting of event) and runtime overhead (sampling frequencies), improper sampling cause improper reconstruction.

Event classification error: If the events are accurately detected and then classified, then the classification error is small.

Event detection delay: It indicates delay between detecting an event and reporting to desired sinks.

Missing reports: If requires Periodic reporting, the probability of undelivered reports need to be small.

Approximation accuracy: For function approximation what is the average, absolute or relative error w.r.t actual function, for edge detection what is edge description accuracy.

Tracking accuracy: The reported tracking position must be close to true position, tracking error should be small.

B. Energy efficiency:

Energy being important resource in WSNs are considered as an optimization goal. It is observed that most of QoS can be enhanced with certain amount of energy except the approximation and tracking task which also depends on network density. Energy efficiency covers many aspects of system which are:

a) Energy per correctly received bit: It provide how much energy is required on average to transmit one bit information from source to destination by considering the energy consumption of all sources at all possible hops. This metric is useful for periodic monitoring applications.

b) Energy per reported (unique) event:
It represents the average energy required to report an event. Only unique events considered in this metric as redundant information about already known event doesnot provide additional info

c) Delay/energy trade-offs:
This trade-off is interesting in WSN because some applications have urgent event which require increased energy investment for quick reporting of such events.

d) Network Lifetime:
It indicate the time for which the network is operational. The probable definitions are:

(i) Time to first node death: when the first node of network is run out of energy or fail or stop operating.

(ii) Network half life: when 50% of nodes run out of energy and stopped operating.

(iii) Time to Partition: when does the first partition of network in two (or more) disconnected parts occur.

Time to loss of coverage: If any area/spot in the network is not covered by or observed by multiple sensor node. For example with K redundant observations in a WSN, the definition for loss of coverage is the first time any spot in deployment region is not covered by atleast K different sensor nodes.

Time to failure of first event notification: of network

It indicates the inability to report an event by a partition in the first place. This can be due to the unnoticed event, which occured due to a dead sensor or due to a partition bet? source & sink.

C. Scalability: It is an important factor in designing efficient routing Protocols. Scalability indicates the WSN's ability to maintain Performance characteristics irrespective of the network size. While making a WSN more scalable often Performance & complexity penalty has occured for small networks. Therefore extreme scalability has direct consequences for protocol design. Hence the architectures and protocols should implement appropriate scalability support rather than trying to be as scalable as possible. In broad sense if some WSN application can provide more efficient solution with less nodes than having more node then scalability need to be made appropriate.

D. Robustness: Apart from the QoS and scalability, the WSN must also exhibit robustness. Their operation should not fail due to less sensor nodes, energy decay, environmental changes, link failure between nodes

Design Principles for WSNs:

From the above discussions the major optimization goals of WSNs are QoS support, energy efficiency, and scalability. However, it is always challanging to design the WSN that achieve all these goals. Therefore, certain techniques and principles have emerged for designing networking protocols. These are:

(a) Distributed organization:

→ Majorly the scalability and robustness optimization goal make the distributed organization critical for WSNs. Here there is abcence of central processor or fusion center (FC) that basically control medium access and make data routing decisions. However, this centralized system is inappropriate when failure of node occurs or WSN operates in limited communication range. Therefore, WSN nodes must self organize themselves to use distributed algorithms & Protocols.

→ In distributed fashion the shortcoming is that it has less accurate solutions compared to the centralized case. with more energy ~~each~~ ~~more~~ one. Therefore, the centralized framework can be used in localized fashion by dynamically electing out a set of nodes that take the responsibilities of centralized agent. These election results in hierarchy and provide dynamic selection. This election process should be repeated contineously to reduce the chances of node overtax, node energy decay, loss robustness.

(b) In-network Processing:-

In distributed architecture nodes of the network not only involved in packet transfer and application programs execution, they are also able to take decisions about network operation. Therefore any data processing which improve the WSN application is an applicable in network processing technique.

(c) Aggregation: This is the simplest in-network processing technique. There are some applications where the sink is interested in getting periodic measurements to check whether there is change in average values of parameter, or difference bet: minimum and maximum value is too large or not, for such cases all sensor nodes data need not be sent to sink rather only the average, minimum & maximum value is required.

→ Aggregation Leads to energy efficiency.

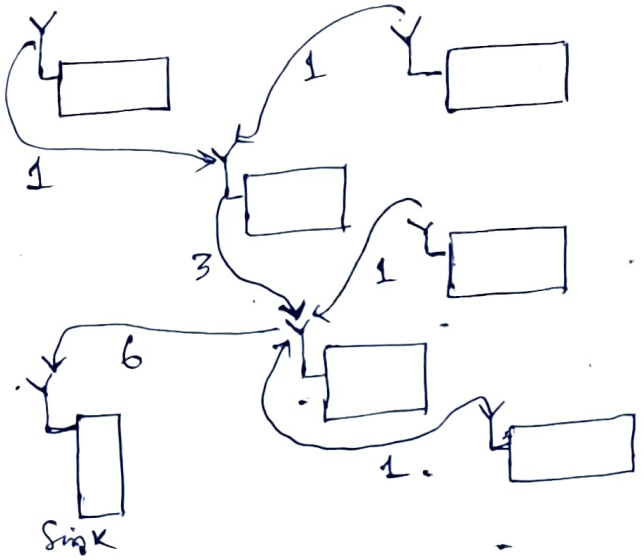
→ Aggregation function is applied to intermediate nodes must be composible to satisfy some conditions to make result meaningful.

→ Mostly used functions for aggregation are average, counting, or minimum. Median function is not open to aggregation at all.

Example of aggregation (computing average values)

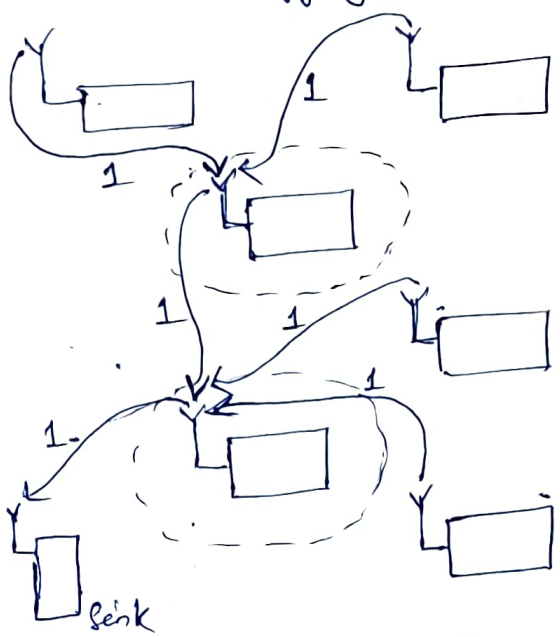
Here a number of sensors used to send their data to sink through multihop communication without and with aggregation.

without aggregation



Total of 13 messages required

with aggregation



The highlighted nodes perform aggregation by computing the average values & require only 6 messages.

* Therefore aggregation reduces the number of transmitted bits/packets by applying an aggregation function in the network.

→ Challanges in aggregation are: where to aggregate i.e, from which nodes, how long to wait for aggregated results, finding the impact due to packet loss.

(ii) Distributed source coding & compression. (Rather than compressing data of each sensor node, the joint information among them used)

In aggregation the objective is not to transmit all bits from sensors to the sink, this reduction in number of transmitted bits can be done through data compression. Their coding and compression requires a lot of effort. Also the information from many sensors need to be encoded & compressed not for a single sensor node. Hence, traditional coding are computationally complex for simple sensor nodes.

→ If the sensors are placed in a physical environment where the readings from adjacent sensors are likely to be similar, i.e they are correlated. Hence, the joint information bet^n. two sensors is required. This type of correlation is called as spatial correlation & it is used not only to transmit the sum of data but also save the overhead. Similarly, temporal correlation can be exploited in sensor network protocols.

(iii) Distributed & collaborative Signal Processing

Apart from trival operations like averaging, fetching maximum the network processing also need to handle complex computations on certain amount of data with energy efficiency.

→ One example of such one is distributed computation of fast fourier Transform (FFT). These distributed computations are mostly applicable to signal processing type algorithms (eg: tracking applications). From the location of input data different algorithms are used to compute FFT in distributed way, with trad-off bet^n. local computational complexity and need for communication.

(iv) Mobile code / Agent-based networking: (used for data routing & data fusion)

It uses the concept of compact presentation of program code that is small enough to be transfered / mobilized from node to node. These code is locally executed, performs measurements collection and then decide where to be sent. In WSNs mobile agents are considered for content of data routing and data fusion. This mobile code is sent out to collect the best planned route by hopping from one travel agent's computer to another & then return to the user from where query is initiated.

(v) Adaptive fidelity: (Trade off bet^n accuracy & energy efficiency)

In WSNs applications such as function approximation the accuracy of an approximation can be improved if more samples of the function is considered. but it eventually requires more

energy investment. This is also the case for event detection and tracking applications.

→ Here an effort is made to adapt the degree of accuracy to the minimum accuracy with energy efficiency.

→ This application is feedback oriented because this application need to adapt to requirements to current status of network such as how many nodes have failed, how much energy to be scavenged from environment, critical event happening etc.

(d) Data centric networking: In typical networks (including adhoc networks), the communication is (addressed) to the specific sender and receiver of data. This data transfer is termed as "address centric" networking paradigm as the address of the intended receiver and sender is important for communication. However, WSN is an event specific network that concern more on the accurate data reception rather than looking for the entities from where it comes. Hence, WSN is data centric network paradigm.

Considering an elephant detection/tracking example, in data-centric application, the nodes in the network that possess "elephant detectors" are informed about the event for "presence of elephant". However, in address/identity-centric network, the requesting node have to find out all nodes that provide this capability and explicitly address them.

→ Data centric networking and addressing improve Performance and energy efficiency of a WSN. This is because data centric solutions scale better by using local information from neighbors and also due to explicit handling of data centric framework with adaptive accuracy.

Implementation options for data-centric networking.

There important ways for data centric networks are:

(i) Overlay networks and distributed hash tables (it is a content-addressable memory)

WSNs have some similarities with peer-to-peer applications like file sharing. In both the cases the user/requester is interested in looking to obtain data from an unknown source.

→ Retrival of data in peer-to-peer network, though unknown source is carried out by overlay (prominent) network by using Distributed Hash Table (DHT). The desired data is identified by a hash key and DHT will provide one or several sources of data associated with the key.

Apart from these similarities, the disparities are. the DHT uses static key and how it correspond to more dynamic, Parameterized request in a WSN.

→ DHT also ignore hop count issues and distance between nodes as DHT comes from IP-networking background (consider nodes as adjacent only on basis of semantic information about their stored keys). This hop count incurs considerable communication overhead in WSNs. Therefore, research is going on by considering the topology of networks & the position of nodes while considering the overlay networks (Networks built over another network)

(ii) Publish/subscribe:

This type of networking has different interaction paradigm, i.e any node interested in given kind of data can subscribe it-, also any node can publish it along with the kind of information in data.

→ After publication, the subscribers are notified about the new data. for example in elephant detection/tracking, the event 'elephant detected' event by any node can be published anytime. The sink nodes then subscribing to the event 'elephant detected'.

→ A subscriber can unsubscribe any kind of event and is not notified further to such events. Evidently, subscription & publication happen at different points in time and identities of subscriber and publisher are unknown to each other.

(iii) Database.

WSNs can also be viewed as dynamic databases. This suits well to the data-centric organization of the networking protocols.

→ Investigating the physical environment by a WSN can be viewed as formulating queries for a database.

→ Casting of sensor networks into relational database framework, the sensors are regarded as virtual table to which relational operators are applied. For example extracting the average temperature reading from all sensors in a given room can be written as:

```
SELECT AVG (temperature)
FROM SENSORS
WHERE Location = "Room 123"
```

In traditional relational database, this query implementation is done by determining an execution plan, the same is followed in WSN. The execution plan has to be distributed by explicitly considering the communication cost.

(e) Some more design principles

(e) Exploit location information: To increase the performance of many applications location information need to be exploited in communication protocols.

→ Mechanism to be used to determine location of sensor nodes, location of observed events. Once the location information is available it simplifies the design and operation of communication protocols and also improve the energy efficiency.

(f) Exploit activity Patterns: The protocols must be designed to handle bursts of traffic due to sudden event occurance. The network must be able to switch between quiescence (silent) modes and of high activity.

(g) Exploit heterogeneity: Related to the activity pattern (how much active) of sensor nodes their construction is made heterogenous. For example some nodes have larger batteries, further reaching comm. devices,

or more processing power. Some nodes have to do more task during network operation, requires more energy depletion, have better opportunities for energy scavenging (nodes under light gets solar energy).

→ Heterogeneity in the network is simultaniously an opportunity and burden. For example nodes with more resources or more capabilities perform more demanding tasks. However, the burden in these task assignment can not be static due to varying nature of WSN and to be reevaluated as time passes and the node/network state evolves.

(h) Component-based protocol stacks and cross-layer optimization:

It promises huge performance gains by going against the standard networking by implementing component based communication protocols as compared to layering based model.

→ They define a ~~def~~ default collection of components that can form a basic "toolbox" of protocols and algorithms to build upon.

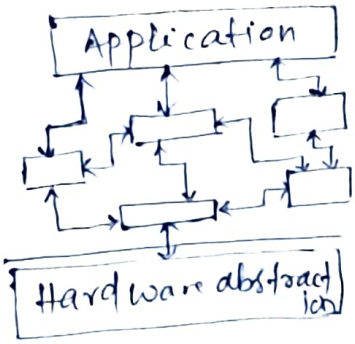
3.4 Service interfaces of WSNs:

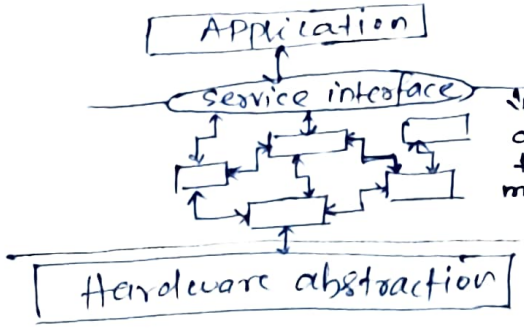
(a) Structuring application / protocol stack interfaces:

Considering the component-based operating system where a component interact with other component using interface (for example the command and event interfaces in TinyOS).

→ Traditional Internet network, where the application Programmer access services of network via interface (sockets). This interface has provisions to handle connections, sending/receiving packets, enquiry about network state. However, this interface is suitable only to adhoc networks not to the WSNs.

→ Therefore, in WSN design choice is to treat the application as another component or as a service interface to access all components in a standardized fashion.





→ The service interface raises the level of abstraction with which an application interact with the WSN.

→ It is also desirable for application to provide possibility to express sensing tasks, in such case a service interface can hide the complexity and is considered as a "middleware" in its own right.

B. Expressibility requirements for WSN service interfaces:

The most important functionalities for service interface includes:

(i) Simple request/response interactions. Support: It stands for retrieving measured value from sensor or parameter setting of a node. The interaction Pattern here is expected immediately. The responses need to be provided Periodically (Periodic measurement applications).

* support periodic measurement applications.

(ii) Support for asynchronous event notifications:

Here a requesting node need to be informed by a network when an event has happened. This is an asynchronous pattern in the sense that there is no prior relationship bet? the time the request is made and the time information is provided.

This asynchronous request have the possibility to cancel the request for information and also have provisions to act accordingly when condition becomes true.

* support periodic reporting of measured values after an event.

→ In both synchronous & asynchronous request, it corresponds to the peer address in a socket comm?. To have data centricity, the peers are implicitely defined by some membership function of an abstract group of nodes. Some of the membership functions are:

(a) Location: All nodes in a given region of space belong to a group.

(iv) observed value : All nodes that have observed values matching to a given prediction belong to a group. Example is to require the measured temp. larger than 20°C.

⇒ Along with these groups, some set theory operations like union, intersection, difference bet° groups is included in service interface.

Note : In general the various ways for identifying addressee of data are by location, by observed value, implicitely by some other form of group membership, some semantic meaningful from (publish/subscribe)

(iii) Easy accessibility of in-network processing functions :

For an operation to access entire groups of nodes, like reading values from group either synchronously or asynchronously, it should specify what type of in-network processing is applied to it. for example (data fusion) process that modify the nature of result must be explicitly allowed by the requesting application.

⇒ In network processing & application specific code may be useful to detect complex events, i.e the events that cannot be detected locally, for which exchange of data between sensors is required.

(iv) Allow specific accuracy and timeliness requirements :

Specification of accuracy of a result can take form such as specifying bounds on the number of members in a group to contribute to a result or the level of compression need to be applied. Similarly, timeliness of data delivery is one aspect. It may be required to provide result quickly with higher cost or slowly with reduced energy. Therefore, tradeoff bet° energy consumption for data packets exchange need to be made explicitly.

(v) Access to node/network status information :

The location, timing, or network status information (available energy reserve, current energy scavenging rate) need to be accessed through the service interface.

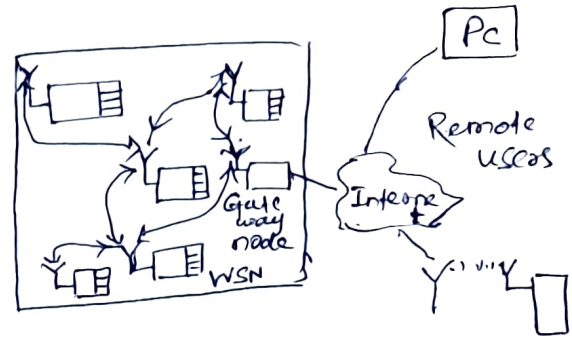
(vi) Support and security management : The seamless connection of

of various nodes or entire network and the access to services in an "unknown" network need to be supported.

→ Security requirements need to be expressed.

Gateway concepts

Need: A WSN concerned to itself is not sufficient and need to interact with remote devices via the internet. Hence, Gateways are necessary for remote access to/from the WSN.



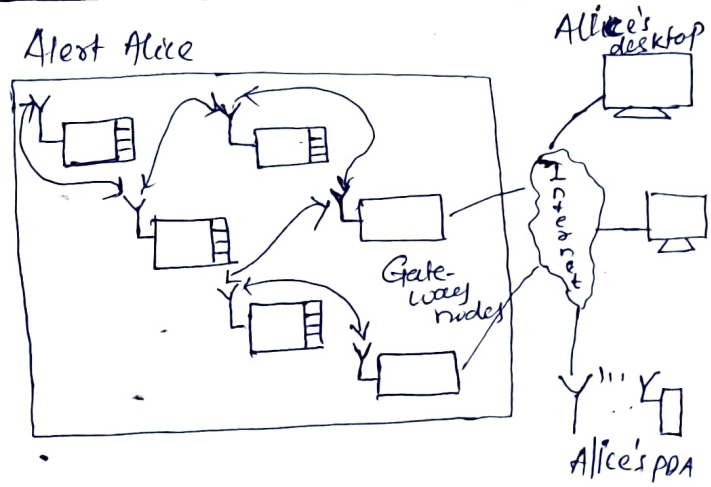
→ The WSN be able to exchange data with mobile device or gateway which establishes physical connection to the internet. This is carried out in the Physical, MAC, and linker easily by the mobile device/gateway which is equipped with radio transceiver in the WSN, or some nodes in WSN to support standard wireless communication technologies such as IEEE 802.11. A simple gateway is a router ~~router~~ between Internet and Sensor network that entail the use of internet protocols within the sensor work However, a gateway can be designed as an actual application level gateway on the basis of application level information.

A. WSN to Internet Communication:

In this case the initiator of WSN-internet communication is present in WSN.

→ Considering alarm message deliver by a sensor node to the internet host. the issues are?

(i) To find gateway from within the network. This is done by solving a routing Problem that offers Specific service, i.e. integrating routing and service discovery.



(ii) choosing the best gateway among several gateways. The option is to build an IP overlay network on top of sensor network.

(iii) How a sensor node knows to which internet host address such alarm message The answer is even if sensor node does not need to be able to process the IP protocol, it includes sufficient information (IP address, port number) in its packets. The gateway extract this information and translate it into IP packets. The main question is which source address used here, the answer is gateway can perform task similar to that of Network Address Translation (NAT) device.

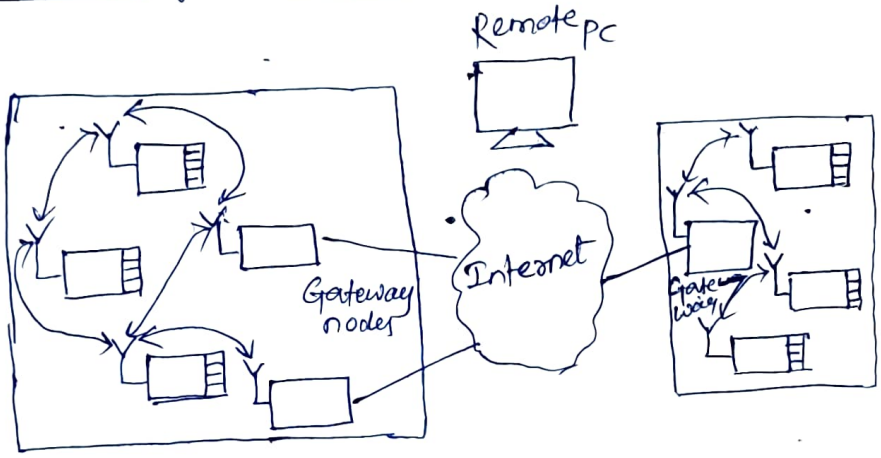
* In this way an intra-WSN event notification message is translated to an internet application message.

B. Internet to WSN communication:

Here, the remote requester trying to access services of WSN.

→ The issues are:

(i) how to find there is an actual sensor network in desired location to answer a need.



(ii) finding out existance of gateway nodes.

(iii) How to translate from IP protocols to WSN protocols.

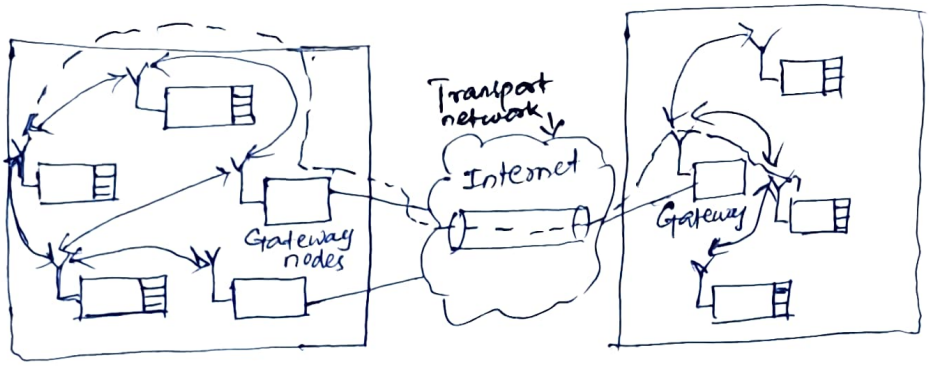
The answer to all lies on the concept that addressing an individual sensor is irrelevant compared to the final data a WSN provides. Therefore, the requesting terminal sends a proper formatted request to this gateway, that acts as application-level gateway or proxy for individual/group of sensor nodes that can answer this request. This gateway translates the request into proper intra-sensor network protocol interactions. These application-level protocol is used by remote requester and gateway is more suitable for communication over the internet and convenient for remote

remote terminal use. The gateway then mask data centric exchange with in network behind address-centric exchange used by internet to select the group of sensor node that need the request from remote requester.

C. WSN tunneling: (Used to form large WSN from small WSNs)

It uses the internet to "tunnel" WSN packets between two remotes WSNs.

→ Here, gateways can act as extensions of one WSN to another WSN.



→ The main idea is to build a larger, "virtual" WSN out of separate parts, transparently "tunneling" all protocol messages between these two networks through internet which act as a transport network.

→ This interface is attractive, but there should not be any confusion bet? virtual link between two gateway nodes with a real link, otherwise protocols (time synchronization or localization protocols) which rely on physical properties of communication link get confused.

→ This tunneling is not applicable only to fixed node architure, it can also be considered a means for interconnection of WSNs with Mobile nodes

Conclusion: WSN use data-centric Paradigm, Hence, the need and possibility to manipulate data as it travels through the network opens new possibilities for protocol design.

The next chapter discusses how these ideas are realized by actual protocols.