OCTOBER 23, 2018

WSN MIDTERM - 1

DR. YUSUF OZTURK

TASNEEM SINGH
822046102

INTRODUCTION

Wireless sensor networks are little sensor nodes that are low-cost, devour low-power and are multifunctional. These nodes can impart untethered in short separations and come stacked with detecting, information handling and conveying segments [1]. Wireless sensor nodes work together to complete their errands through correspondence which can either be ad hoc or methodology based where the gateways goes about as the asset through which sensor data is gathered.

Clustering is an intelligent method to lessen network consumption and increment energy efficiency [2]. Since the nodes in general execute divergent assignments, consequently they are regularly assembled into clusters dependent on particular requirements. This suggests that having cluster heads can fundamentally add to scalability, lifetime and energy efficiency.

Development of a cluster includes relegating the specific job to a node dependent on their parameters. The coordinator of the group which is in charge of the handling, conglomeration and transmission of the information to the base station is known as the cluster head. The hubs that are in charge of detecting and sending the gathered information are the child nodes or Member nodes.

Question 1: Contention and Degraded Channel Utilization

One of the significant advantages of clustering is the spatial reuse of assets to expand a system limit. On the off chance, if the nodes are not neighbors they can utilize similar frequency for wireless communication.

Clusters transmits units of data called packets and they interface with the channel in some way - this is the thing that associates the queue of packets to the shared channel. At the point when the channel is shared, and at least two nodes are transmitting at the same time, the packets will collide and be lost. This is known as collision and regardless of whether just a piece of a packet is engaged with an impact, the whole packet will be lost. This prompts debased channel utilization, except if or until the point that a mechanism is set up that recognizes and avoids collision.

In general, the contention frameworks can be classified as follows [3]

1. Polling
2. Reservation
3. Random Access

Channel Polling is the way toward checking the wireless channel to recognize activity. Amid this period, the nodes don't get data, yet they wake up to recognize activity. In the event that a node detects some activity, it remains wakeful to receive data or else returns to rest. The polling interims can be powerfully changed dependent on network activity conditions. The weaknesses of this protocol are: energy wastage and the requirement for successive conveyance of schedules for avoiding collisions.

Since there are three clusters, 100% utilization can be achieved by always letting one cluster use the channel all the time, assuming it has infinite data to send and if not, we can let the other cluster jump in until the first cluster has data.

The above scheme is not fair at all. To introduce fairness ideally, we divide the capacity equally among the nodes requesting to use it, but this is also marred by the following problems:

- Not all nodes may need to send data all the time
- How to determine who has the data to send?
- Some data might be more important than the other – how to prioritize?

One of the existing protocols such as ALOHA allows all client nodes to communicate with the hub on the same frequency. The protocol is simple and straightforward – newly arrived packets are transmitted immediately. The packet suffers collision if the previous attempt is not yet finished or the next packet was sent to soon by the other cluster. Understandably, the protocol witnesses great amount of collisions and hence pure ALOHA gives the lowest throughput of just about 18%.

The other possibility of transmitting packets originating from three different clusters is to use slots where a cluster sends a packet in the next time slot with probability p. The main question is: what should be the value of p? If p is very large, there will be increased number of collisions

and if p is very small, the channel will be under-utilized. This is very basic idea behind Slotted ALOHA that gives a throughput of 37%.

One plausible solution to the above stated problem could be a varying value of 'p'. If the clusters can dynamically adjust the values of p from a feedback mechanism, a reasonable amount of utilization can be obtained. For instance, if there is a collision, we can decrease the value of p and it can be kept the same or be the increased depending on the fact that the transmission is working successfully. Some of the concerns associated with this protocol are:

- After a series of successive failures, p diminishes to a really small value which leads to lower transmission attempts. This can starve out the cluster nodes.
- Some nodes can capture the channel for a period, starving out the others. This occurs when one node maintains a particularly high p leading to short term unfairness and problems with the queues [4].

Another contention-based solution is using RTS/CTS where before sending a packet, anode sends a "Request to Send" packet. If the receiver receives the RTS correctly, it replies with a "Clear to Send". Any node hearing the CTS, even if it didn't hear RTS, knows that the channel is busy and for how long. This information is stored in a Network Allocation Vector. This solution is marred by the hidden/exposed terminal problems.

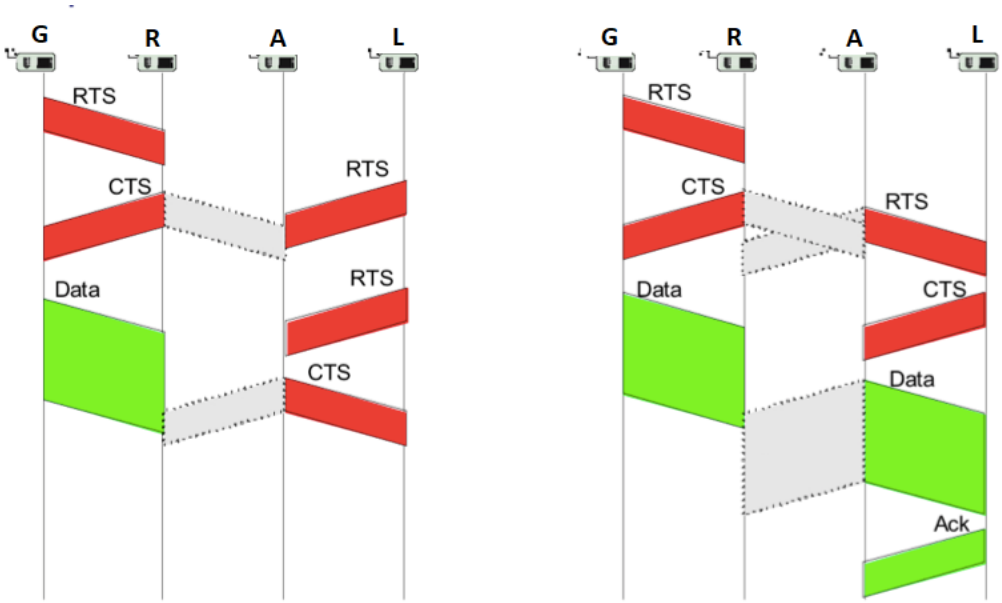


Fig 1: Example problem cases for the given scenario (Source: Sami Rollins, "MAC Layer", University of San Francisco, PowerPoint Slides 12-13)

With respect to the given data collection tree, following are the problems that are encountered.

Problem #1: A cannot decode the CTS from R to determine the duration of the exchange because of the collision with RTS from L. A does not know how long transmission will be and replies to second RTS from L causing a collision.

Problem #2: A cannot decode R's CTS and upon receiving CTS from L, sends data causing collision at B.

Possible solution: Make sure that CTS packets are longer than RTS packets. A is guaranteed to sense R's CTS after it finishes its RTS and will not send data. Also, in such cases, A must defer transmission for time to send 1 max-length data packet.

Question 2: Multichannel MAC implementation and its benefits

Since it is a clustered network, each of the three cluster has a unique ID and IDs of all clusters form a sequential list. Each cluster head is aware of its own as well as its neighbor Cluster Head's ID. Each node maintains a Neighbor list which includes information such as the neighbor's ID and assigned channels.

The assignment of channels is based on cluster ID, lower IDs selects channel with priority. After, when cluster head refers a Neighbor list before it selects a channel, it is ensured that it does not select the channels until all other CHs with lower IDs have selected their channels. The cluster head broadcasts its channel information after the selection through public channel, so that its neighbor CHs could update their NL. Meanwhile, the nodes in the same cluster tune the channel of their transceivers to the same channel selected by the CH.

After channel assignment, each CH tunes its transceiver to the assigned channel. To meet communication requirements of the cluster with most nodes, the Base Station broadcasts the maximum time slots of intra cluster communication within each channel. In this stage, Cluster Head collects data from its member nodes. To avoid collision caused by multi nodes sending data to CH at the same time, TDMA is adopted within cluster for channel access. The time frame structure of each cluster in the intra-cluster communication stage are composed of Synchronous beacon, Request phase, Schedule phase, Data transmission, and Sleep phase.

Firstly, CH broadcasts synchronous beacon within its cluster channel, and synchronize with its member nodes. Nodes with data to be sent send requests to CH with CSMA method, which includes node ID and the size of the data to be sent. On receiving requests from its member nodes, CH schedules the time for node members based on priority and broadcasts time slots to them. Each member node then switches on its transceiver and sends data within its assigned time slot and switches off the transceiver and sleep in other time slots [5].

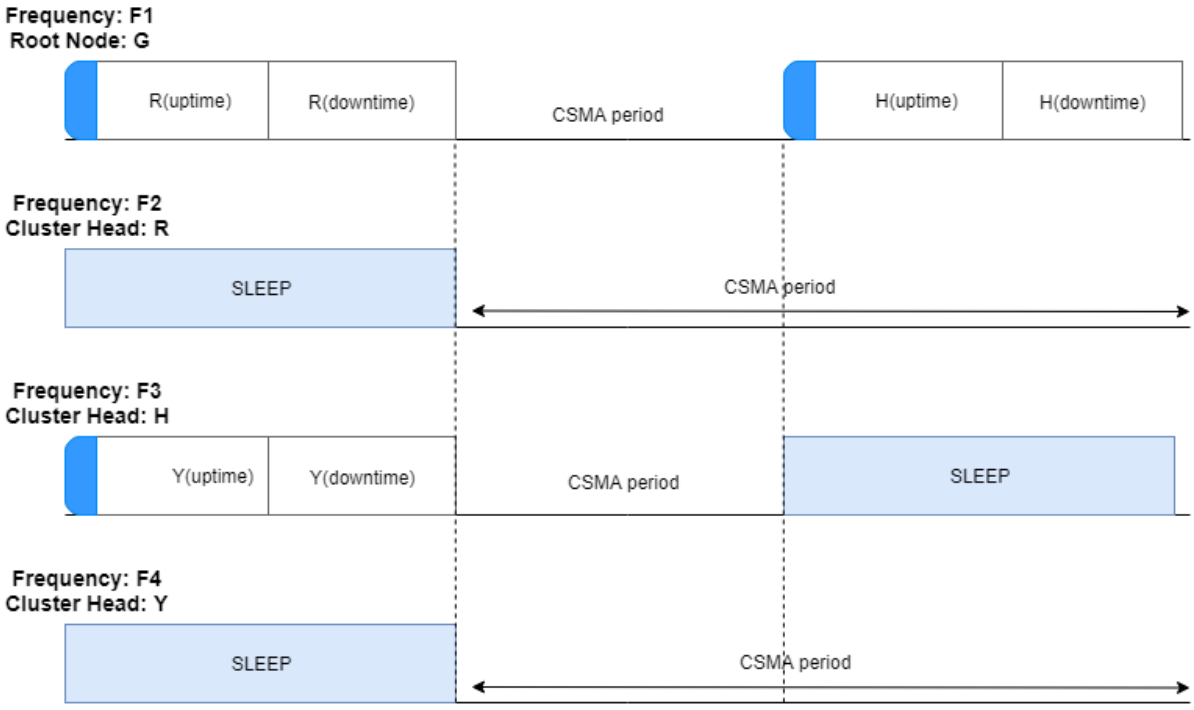


Fig 2: Multi-channel implementation for root and network nodes

After aggregating the sensor readings in its cluster, a CH forwards the data to the BS over an inter-CH path. After intra-cluster communication, each CH is still on its own frequency channel. However, they have to switch to the same channel in order to communicate between clusters along the path. Therefore, different channels need to be assigned to different paths to ensure non-interference communication between them. Hence, each path uses the channel used by the cluster nearest to the Base Station. The Cluster Heads on the same path then use TDMA to access the channel and the time slots are scheduled in depth-first ordering, of which the deepest CH has the highest priority.

Some of the advantages that can be derived from multichannel MAC implementation is:

- Increase in total network throughput by reducing packet collision rate
- Reduced average delay
- Increased security

From fig.2 we infer that when a node visits the other node, the child nodes enter a sleep period where no communication takes place between them and the parent node.

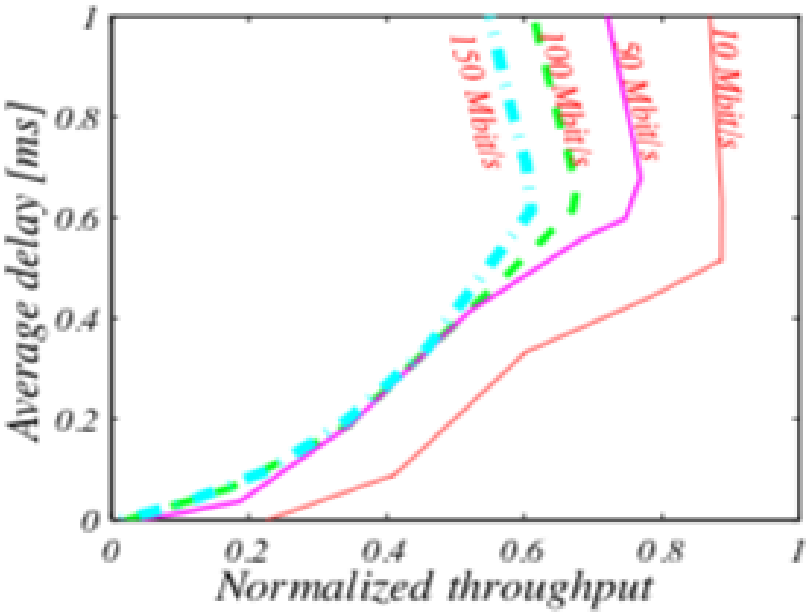


Fig 3: Average delay versus the normalized throughput for single channel simulations (Source: A comparison between single and multi-channel CSMA/CD protocols of equivalent capacity by Rodellar et al.)

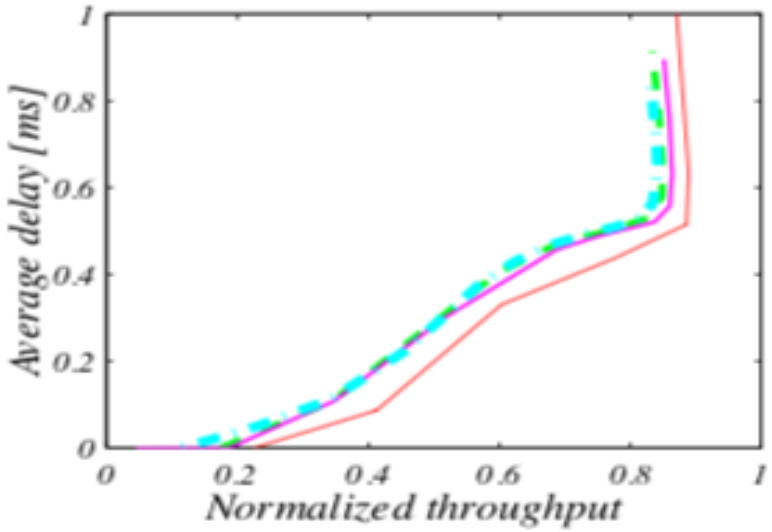


Fig 4: Average delay versus the normalized throughput for multi-channel simulations (Source: A comparison between single and multi-channel CSMA/CD protocols of equivalent capacity by Rodellar et al.)

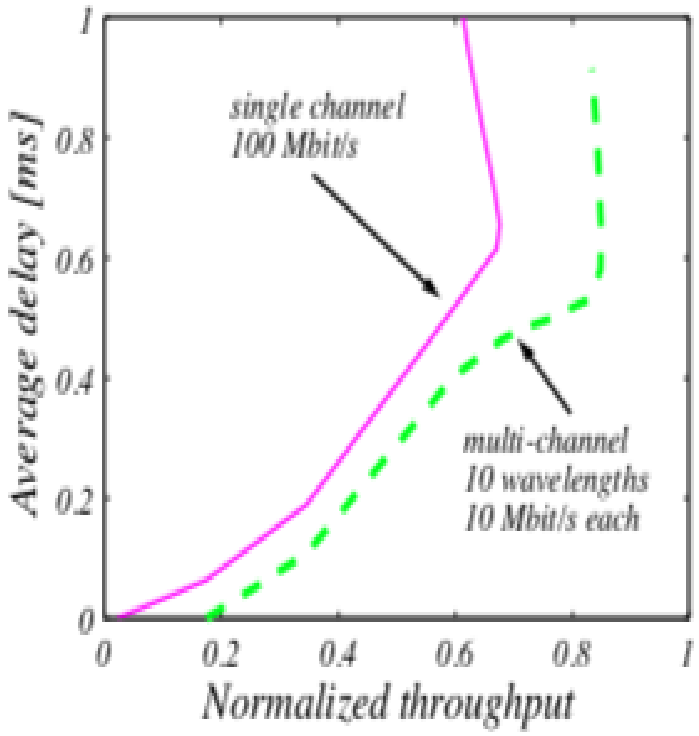


Fig 5: Average delay versus normalized throughput for single channel and multi-channel protocol of 100 Mbit/s total capacity (Source: A comparison between single and multi-channel CSMA/CD protocols of equivalent capacity by Rodellar et al.)

[6] discusses the single channel and multi-channel MAC performances where four capacities are considered 10,50,100 and 150 Mbit/s. When increasing the capacity, the major goals are to obtain an adequate delay and throughput results in order to support real-time services. From the simulation results observed by the author, it is clear that the multi-channel protocols have a better performance measures than single channel ones.

Question 3: Frequency allocation mapping to GTS OF 802.15.4

In beacon enabled mode, the coordinator generates periodic beacon frames with a provision of a superframe structure, which provides applications with a time frame property. It also makes it possible to obtain real-time guaranteed service by allocating the Guaranteed Time Slot (GTS) on a first-come-first-served or on a priority basis, where a GTS is a period that can be reserved by a sensor node for its data transmissions. The superframe structure is characterized by a Beacon Interval (BI) and a superframe duration (SD). Here, BI is the time between two consecutive beacons and includes an active and inactive period [7].

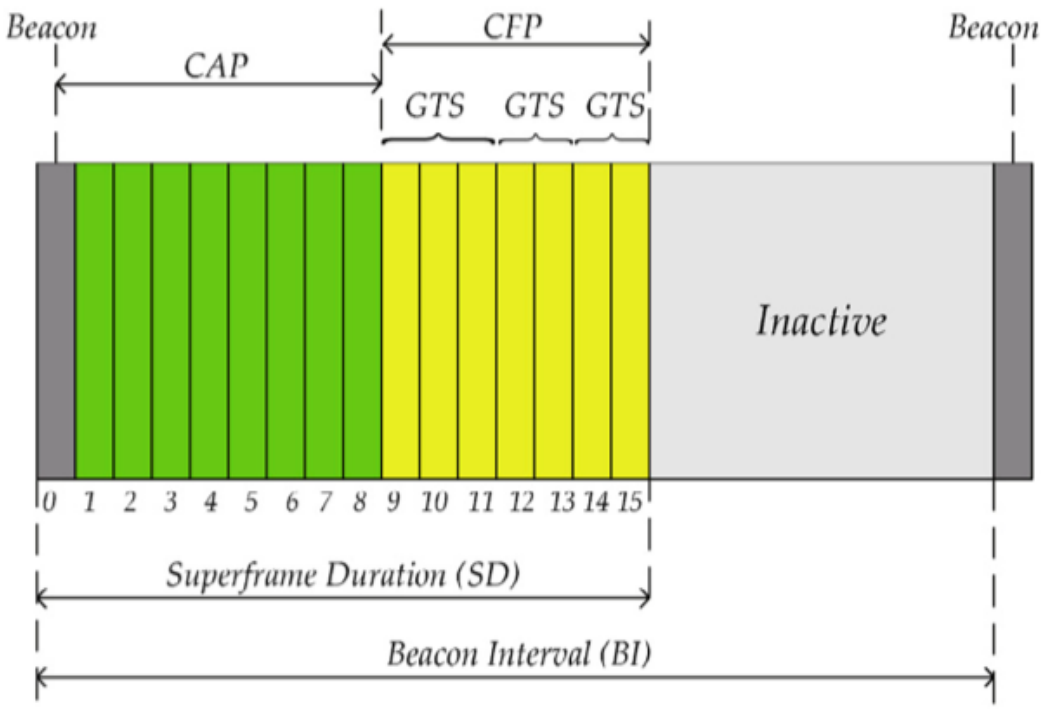
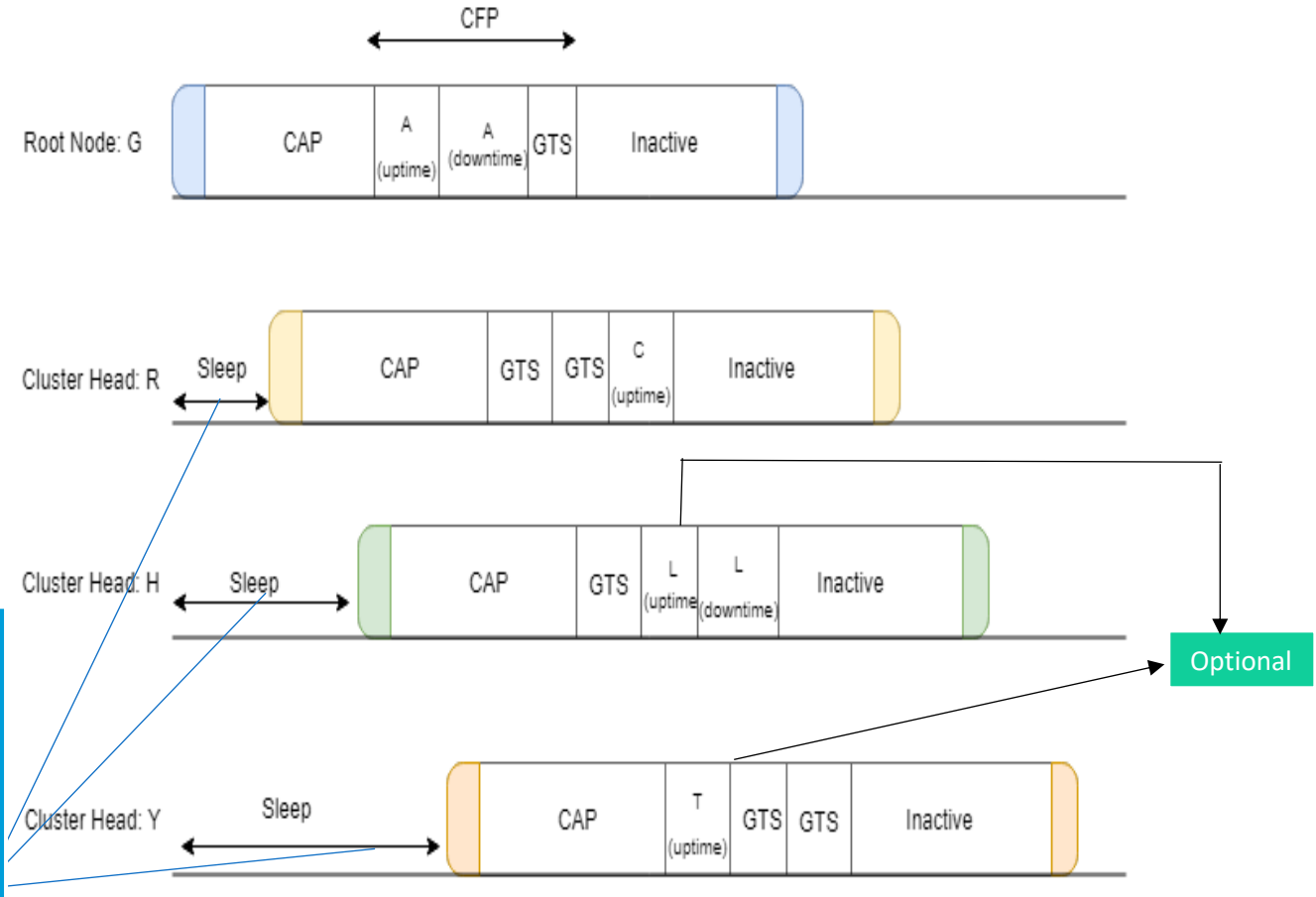


Fig 6: IEEE 802.15.4 Superframe structure

The superframe duration is divided into – Contention Access Period (CAP), Contention Free Period (CFP) and the inactivity period. During CAP, a slotted CSMA/CA algorithm is used for channel access. MAC commands such as association requests and GTS requests are transmitted in the CAP. CFP is for the use of nodes requiring dedicated bandwidth, the

communication occurs in a TDMA style by using number of GTSs, pre-assigned to the individual sensor nodes.



Sleep times can vary. For nodes, this can be incorporate between the CSMA and GTS periods.

Fig 7: Map of frequency for root and network nodes

Whenever a device requires a certain guaranteed bandwidth for transmission, the device sends GTS request command using CSMA/CA during CAP. The CAP starts immediately following the beacon and goes until the start of CFP. Upon receiving the request, the coordinator first checks the availability of GTS slots in the current superframe, based on the remaining length of the CAP and the desired length of the requested GTS. Provided there is sufficient capacity in the current superframe, the coordinator determines, based on FCFS fashion, a node list for GTS allocation in the next superframe, and informs the nodes about the allocation of slots in the GTS descriptor in the following beacon frame.

The CFP starts at the boundary that immediately follows where the CAP ends and it should complete before the end of the active portion of the superframe. All the GTSes are located in

the CFP and they occupy continuous slots. The length of CFP can increase or shrink depending on the length of available GTS slots.

For the above figure, root node G will have a CSMA period where it can interact with its child node A. This period is incorporated in CAP. Similarly, for cluster heads H, R and Y, each of them will have CSMA periods for their child nodes L, C and T.

The GST period for R and Y can be optional since both the Cluster Heads have just a single child and so having a GST is in turn a waste of memory. Or these can be reserved for future new nodes that might join in. As a new node joins the network, they can request GTS slots depending on the cluster they become a part of. The cluster head can then assign them a slot in the CFP accordingly.

Question 4: Synchronization

Synchronization in wireless nodes allows the use of a TDMA algorithm in a multi-hop wireless network. Wireless synchronization is used for various purposes including location, proximity, energy efficiency and mobility. To find out about the exact location of nodes in a wireless sensor network, time synchronization is used. Time stamped messages will also be transmitted between the nodes to determine their relative proximity. Time synchronization is used to save energy; it will allow nodes to sleep for a certain time and then awaken to receive a beacon signal periodically. Since a majority of wireless nodes are battery-powered energy efficient protocols are a necessity. Finally, a common timing between nodes will enable the speed of a moving node to be determined. There are three basic types of synchronization methods for wireless networks. The first is the relative timing and the most simple. It depends on ordering messages and events. The basic idea is to determine whether Event 1 occurred prior to Event 2. This can be achieved by comparing local clocks. The next method is the relative timing in which network clocks are independent of each other and nodes track drift and offset. Normally, the matching neighbor nodes holds these information on drift and offset of a particular nodes. The nodes can synchronize their local time at any time with other local nodes.

The last method is global synchronization, where the global timescale is constant throughout the network. This is obviously the most complex and hardest to implement. Very few synchronizing algorithms use this method especially since this type of synchronization is usually not needed[8].

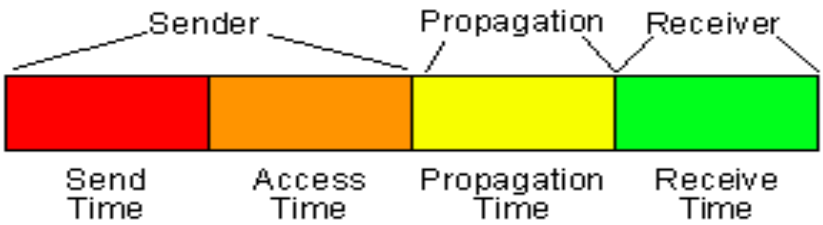


Fig 8. Breakdown of packet delay components (Source: Time synchronization in wireless networks by Michael Roche)

As shown in fig. 8 the sender's time builds the time message to transmit on the network. Access time is the MAC layer delay in network access. Propagation time is the time to transmit bits on the medium. The receive time is time to process the message and to transfer it to the host.

Protocol to synchronize all nodes with their respective cluster centers

The cluster head h transmits its time information to its cluster members including the hardware clock value $\tau_h(t)$, skew compensation parameter α^h(t), and offset compensation parameter β^h(t) to its cluster members. When the cluster member *i* receives its cluster head *h*'s information, the current reading of its hardware clock $\tau_i(t)$, clock compensation parameters α^i(t) and β^i(t) will be recorded, and the information will be returned to its cluster head h. Now, one time information exchange is completed.

The local node l will update its clock compensation parameters based on the latest data received from the node j when it has a historical record of the same node. Meanwhile, node j is a cluster member when the cluster head is node l. If node l is a cluster member, node j is a cluster head.

Cluster head A would adjust its logical clock parameters to synchronize with node 1 in the logical time after two sync message exchanges, since the relative clock skew between two nodes is based on at least two sets of adjacent hardware clock. Likewise, after another sync message exchange between cluster head A and its entire cluster member, nodes 1, 2, 3 and 4 can synchronize with cluster head A; namely, cluster A requires a total of three message exchanges to realize time sync.

When cluster A has achieved synchronization, it serves as the time source of its adjacent cluster F and B. So, after another at most three times sync message exchange for cluster B and F, respectively, cluster A, B and F can synchronize as shown in Figure 9b, and the sync process for cluster C, D and E is shown in Figure 9c [9].

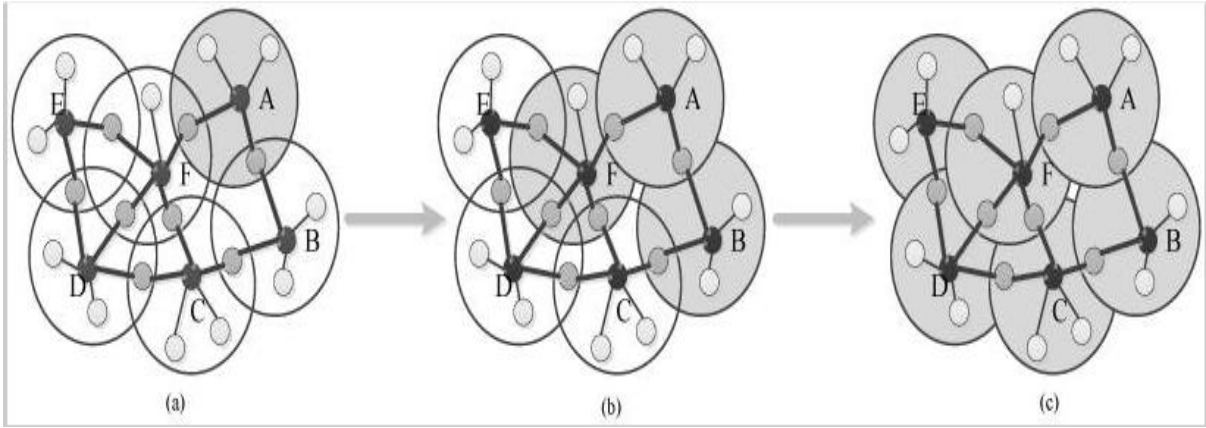


Fig. 9: Illustration of inter-cluster time synchronization. (a) Time synchronization in cluster A (b) Cluster B and F synchronize with cluster A (c) Cluster C, D and E synchronize with their adjacent clusters respectively.

Clock Drift

Since all hardware clocks are imperfect, local node clocks may drift away in time, so observed time or time intervals may vary for each node in the network. For some node i in the network, the local clock can be approximated as: Ci(t)= a+ bi, where ai(t) is the clock drift and bi(t) is the offset of the node i's clock.

Drift denotes the clock rate (frequency) and the value difference from real - time t is offset. Using equation (1), we can compare the local clocks of two nodes in a network, say node 1 and node 2 as: C1(t)= a12· C2(t) + b12 (2) We call a12 the relative drift, and b12 the relative offset between the clocks of node 1 and node 2. If two clocks are perfectly synchronized, their relative drift is 1-meaning that the clocks have the same rate and their relative offset is zero-meaning that they have the same value in that moment [10].

Ratio based time synchronization protocol

By implementing RSP, which is a tree based synchronization scheme in our network, root G is elected by the leader election algorithm and it will the first to start the time synchronization procedure.

Root G sends out messages which contain the time stamp information and sender ID. The nodes H, R and A, all in the G's radius will receive the synchronization packets. After this reception, they will correct their local clocks and become synchronized nodes.

Similarly, the newly synchronized nodes will broadcast new synchronization messages just like root G did and all the nodes now in the radius of synchronized nodes H, R and A will receive these synchronization messages.

The above procedure will continue until each node in the given network forms a virtual synchronization tree structure. If a node receives a duplicate synchronization message from its neighbor – it is discarded.

Fig. 10: Synchronization message packet

The first field of the packet, Send_ID is the ID that belongs to the sender/node of the synchronization message.

Seq_num is a variable field which increases by one as soon as a new synchronization message is transmitted.

Time_stamp is the global time that is estimated by the sender and is used by the receiver to estimate global time.

Sync_root is the root ID reserved by the synchronized nodes and New_root is used to announce that a new node is elected. If a new root is elected, the existing nodes will use the new root's global time-stamps to calculate and adjust their clock drift ratios and offsets.

Question 5: Cluster Tree Mesh network

In a cluster tree mesh network, the cluster heads maintain 3 tables each which are: Members, List of unassigned addresses and Child Networks. Here, the concept of nodes maintaining a table is introduced, which is called the neighbor table. Since all nodes in a cluster are operating on the same frequency, every node can hear every other node in its cluster as well as its peripheral.

The member list contains all the nodes in the given network whereas the networks table only contains the cluster heads working under the single umbrella of the root node. List of unassigned addresses come in handy when we a new node joins the network and it needs to be assigned an address. The list size decreases with every new joining node.

The following scheme is utilized for the protocol used:

1. Every node transmits a 'hello' message every x seconds.
2. Every node publishes their neighbors every y seconds, where y>x.

For instance, if node 3:2 wants to talk to node 2:2. The green and the blue circles represent the radius in which node 3:2 and cluster head 1:1 can hear respectively. The red direction lines indicate the path node 3:2 would take conventionally to reach node 2:2.

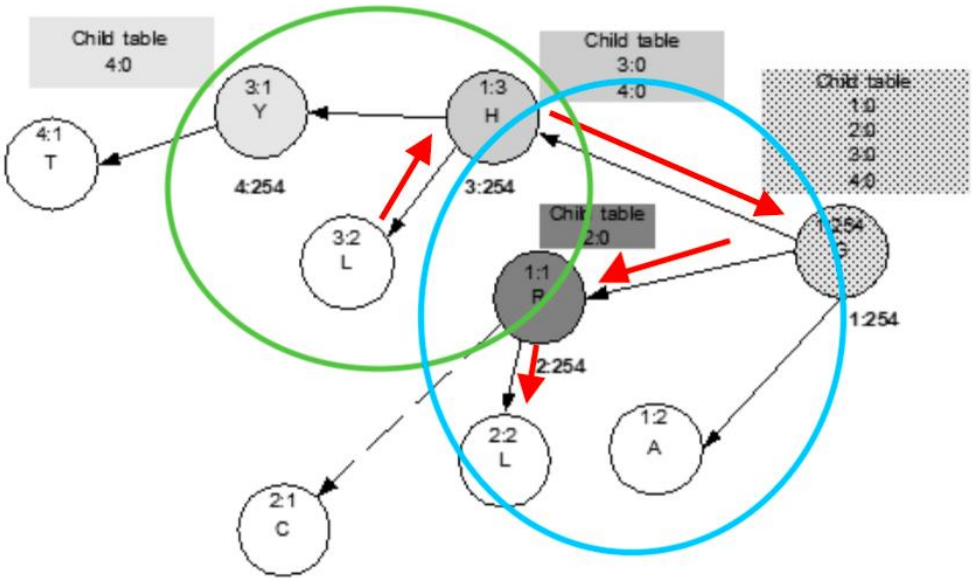


Fig. 11: Representation of actual network scenario in terms of hearing radius for nodes 3:2 and 2:2

According to the above information, the following table is the depiction of node 3:2 neighbor table.

NODE	NEXT HOP
1:3	3:2
3:1	3:2
1:1	3:2
1:254	1:1
1:2	1:1
2:2	1:1

Using this table, we can see that node 2:2 is 2 hops away from node 3:2. This was possible due to the fact that node 2:2 is in the hearing radius of cluster head 1:1 which in turn is in the hearing radius of node 1:3. Using this table the node 3:2 L would not have to go through four hops to reach node 2:2 L, instead now that it is aware that it can reach it within two hops, it will try to establish the shortest and the most cost-effective path.

Some of the disadvantages that the proposed protocol has:

- Memory and Bandwidth consumed
- Slow convergence because of update rates

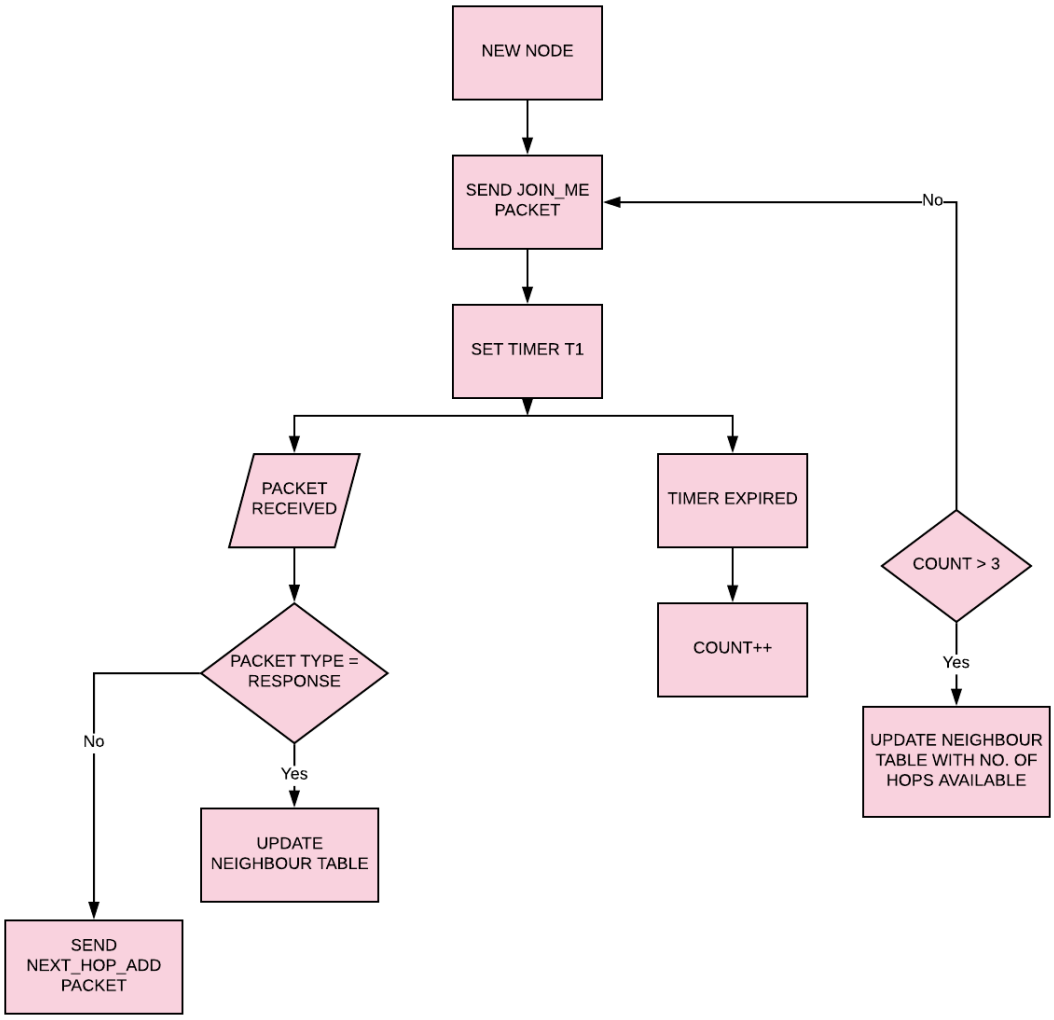


Fig. 12: Chart to determine the new node for next hop discovery

Rules that will be executed by the nodes to achieve mesh routing:

1. If destination is in your neighbor table, NextHop = NextHop[dest]
2. If the destination networks parent is in your neighbor table, NextHop = Parent[dest]
3. If a node in the neighbor table belongs to the same network, NextHop = A node
4. Deliver the packet to the parent.

Neighbor tables and cost

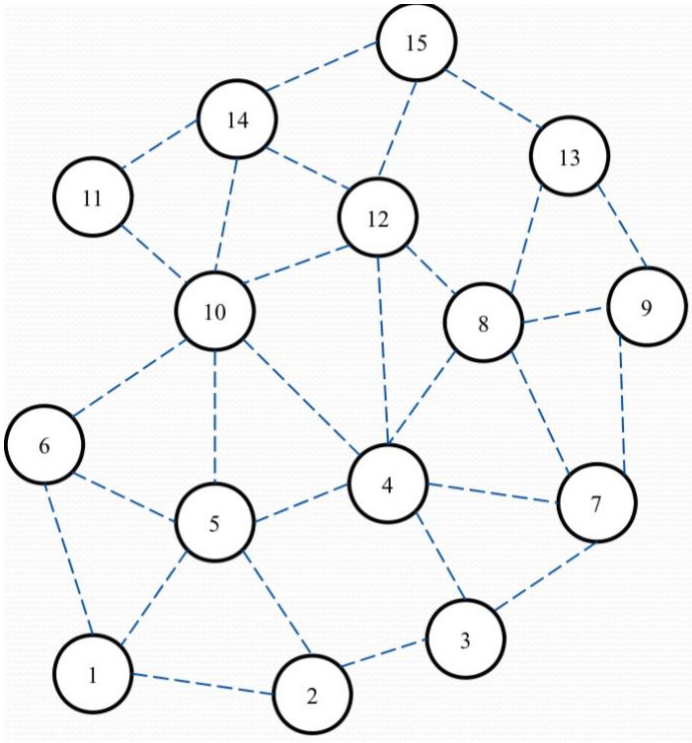
Source Based Routing can be of two types:

Strict – Packets will have to follow the path defined. If any of the path is not available, the packet will be dropped.

Loose – If the path is not available, revert to IP's best effort services

To construct the routing table, the sending node first sends a 'hello' message to all its neighboring nodes. Every node on the network maintains a neighbor table which comes in handy when routing packets from the source to the destination. Similar, to the above case, the sender node by the virtue of its own and its neighbors neighbor tables, has the ability to cover all the nodes in the given network. Deciphering the hops that it will take the source to get to the destination following different paths, a cost factor is determined which is crucial in selecting the path. In general, we choose the path with the least cost associated with it for routing a packet [12]. There can be numerous routing metrics which are namely:

- Battery Life
- Bandwidth of the link
- Reliability of the link
- Mobility of nodes
- Other resources



Dest	NextNode	Dist
2	2	1
3	2	2
4	5	2
5	5	1
6	6	1
7	2	3
8	5	3
9	2	4
10	6	2
11	6	3
12	5	3
13	5	4
14	6	3
15	5	4

Fig. 13: Example network with table constructed in reference to node 1

Each node upon receiving an update, quickly disseminates it to its neighbors to propagate the broken-link information to the whole network. Thus, a single link break leads to the propagation of table update to the whole network.

Neighbor discovery and routing

Since we need a protocol to resolve the globally unique addresses to the dynamically assigned addresses we use Ad Hoc Demand Distance Vector (AODV) Routing.

Summary Points for AODV:

- Reactive or on-demand
- Establishes best path
- Converges quickly to shortest path
- Lifetime can be kept short for routing tables since not all discovery packets have to go through all nodes
- Makes use of the routing tables to store routing information which stores the following data.

destination addr	next-hop addr	destination sequence	hop count	life time

- If a nodes wishes to send a packet to a particular destination - It checks its routing table to determine if it has a current route to the destination
    - If Yes, forwards the packet to next hop node
    - If No, it initiates a route discovery process
- Route discovery process starts with the creation of a Route Request (RREQ) packet. The source node is responsible for it.
- The packet contains source node's IP address, its current sequence number, destination IP address, destination sequence number
- Packet also contains broadcast ID number
    - The ID gets incremented each time a source node uses RREQ
    - Broadcast ID and source IP address form a unique identifier for the RREQ

Neighbour and route discovery

- A reverse route entry for the source node in its route table is created once an intermediate node receives an RREQ
    1. Reverse route entry consists of <Source IP address, Source seq. number, number of hops to source node, IP address of node from which RREQ was received>
    2. Using the reverse route a node can send a RREP (Route Reply packet) to the source
        - Reverse route entry also contains – life time field
- RREQ reaches destination - In order to RREQ a node should have in its route table:
    1. Unexpired entry for the destination
    2. Seq. number of destination at least as great as in RREQ (for loop prevention)
- The node responds to RREQ only when both the conditions are met and the IP address of the destination matches with that in the RREQ. The response involves sending an RREP back using unicasting and not flooding to the source using reverse path
- If conditions are not satisfied, then node increments the hop count in RREQ and broadcasts to its neighbors
- Ultimately, the RREQ will make to the destination [13].

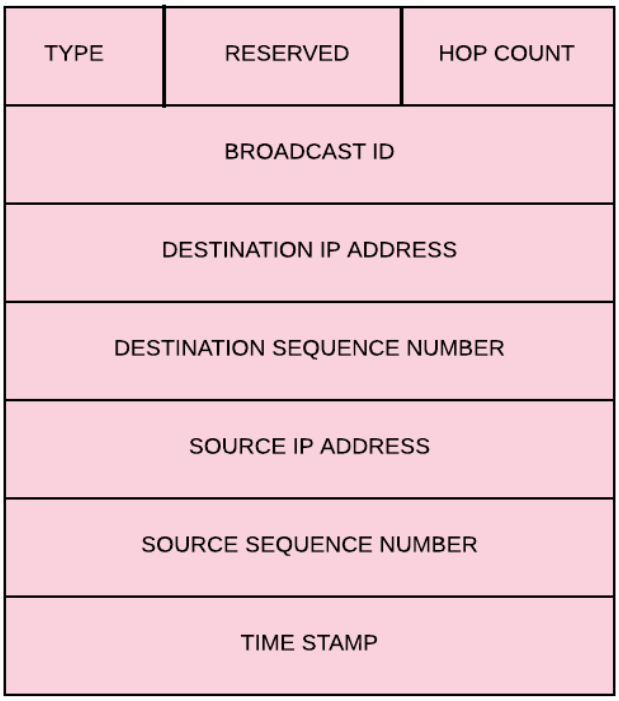


Fig. 14: RREP – Route Reply packet

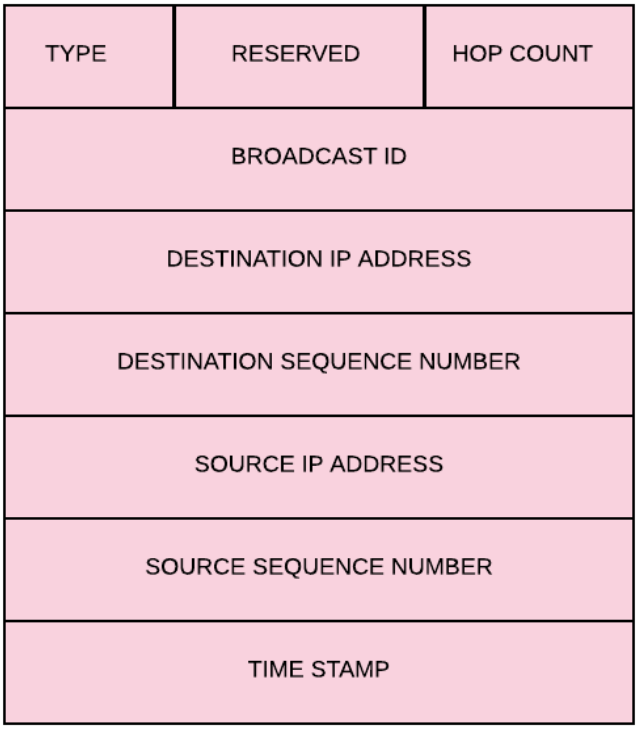


Fig. 15: RREQ – Route Request Packet

Question 6: Resolving GUID addresses

Because of the dynamic addressing contrived in this system, there can be at most number of 254 clusters and a maximum of 254 nodes for every cluster. Along these lines, on the off chance that one of the nodes needs to send information to a destination then it might be conceivable that the packet should experience multiple hops to arrive at the destination. There can be numerous ways that can be pursued to reach to a destination. Along these lines, with the aim to find the most proficient way to the destination, a routing protocol is necessary. The routing of data packets through various hops is possible with the help of routing tables. Since every node keeps a neighbor table, this allows one hop communication amongst all the nodes. This neighbor table likewise comprises of the GUID of every node.

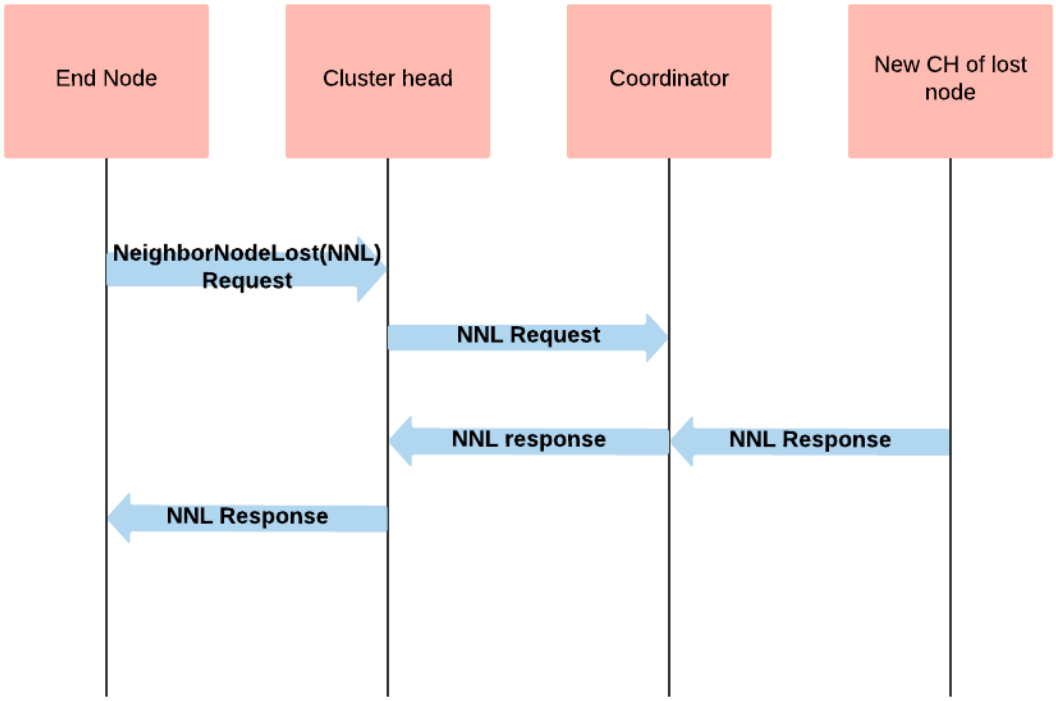


Fig. 16: Sequence diagram for neighbor node discovery failure and recovery

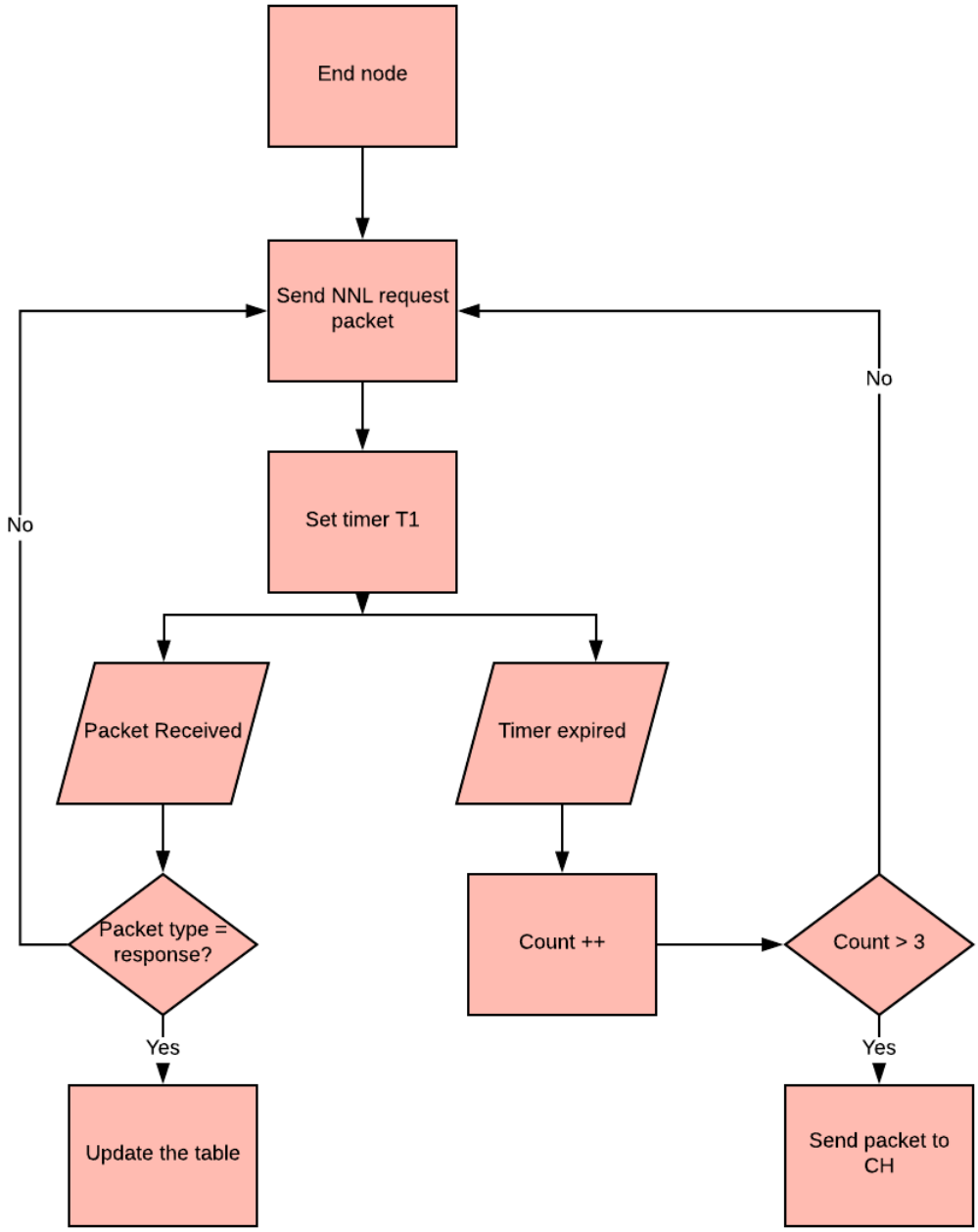


Fig. 17: Activity diagram for end node to declare lost neighbor

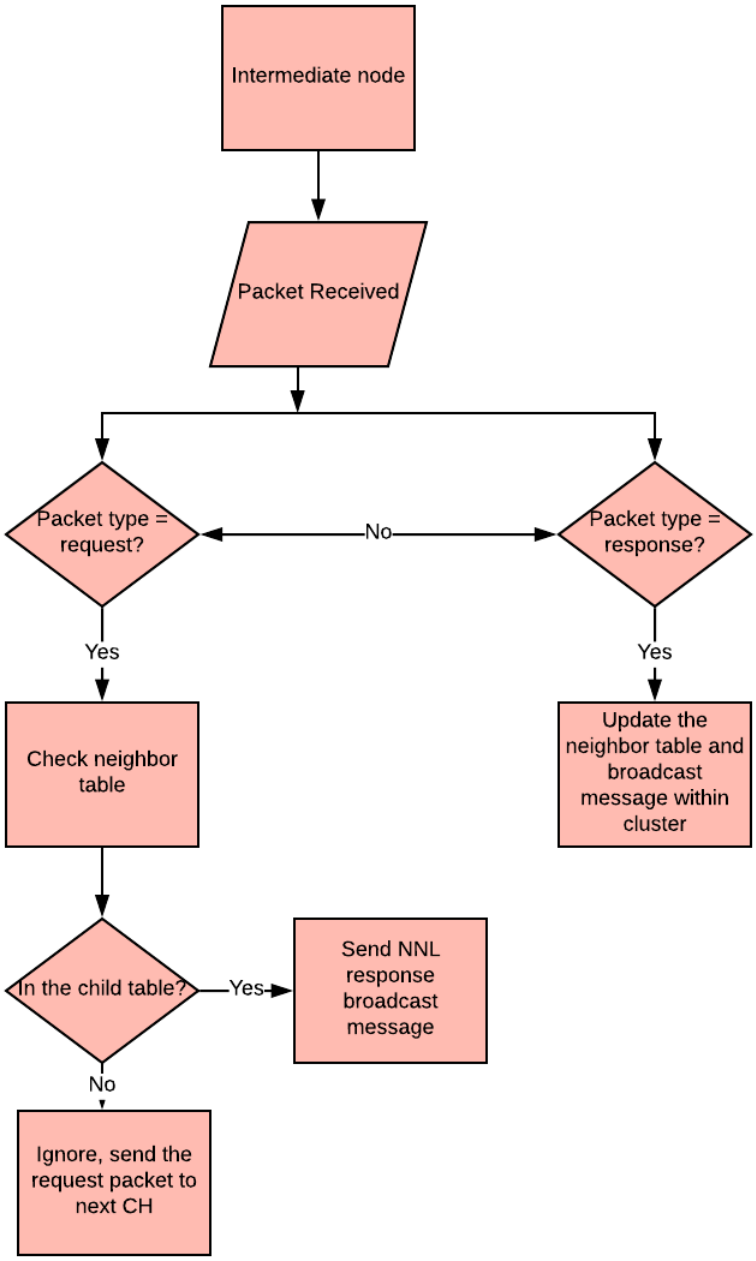


Fig. 18: Activity diagram for intermediate node that receives NNL packet

The cluster head is informed by the node that lost its child node that it is unable to reach it. In response to this, the cluster-head then informs the PAN Coordinator that the node with a given Node Id is lost. This packet is known as NNL request packet. The PAN coordinator then broadcasts a packet containing the lost packet's GUID and asks the cluster heads if they have the given node as their child. All the cluster heads that receive this message begins looking into their cluster. If any cluster head has the lost node, it will update its neighbour table and

send a broadcast message saying that the given node is now a part of its cluster. This broadcast message is received by all the nodes and the nodes then consequently update their Neighbour tables and routing tables. This is how the network resolves the issue using GUID.

Ques 7: advantages and disadvantages of 1) data collection tree, 2) mesh with single hope neighbor table, 2) mesh with two hop neighbor table

Category	Single-Hop	Two-Hop	Data Tree Clustering
Reliability: • **Frequency Agility** • **Message Loss** • **Adaptability** • **Security**	Moderate Reliablility	Moderate Reliablility. More secure since clients are not directly connected to the gateway, so a flooding attack cannot bring the entire network down.	Highly reliable. As a node leaves the given network the other nodes re-route their signals to ensure a more reliable communication path.
Power Management: • **Sleeping Routers** • **Sleep Strategy** • **End Nodes**	Moderate power consumption	More power consumption. Every hop reduces the bandwidth, outer nodes will take longer to reach the network and will experience performance degradation which will take up more energy to recover.	Moderate-low power consumption. Power consumption is uneven across network nodes. Nodes closer to base station consume more power than leaf nodes.
Scalability: • **Network Size** • **Traffic Volume**	Highly scalable but lags in network planning	Moderately scalable and poor network planning. Achievable throughput lower than single hop due	Less scalable but good network planning. Setting up and maintaining a large network can

		to transmission overlap of consecutive forwarding nodes.	take a considerable amount of time.
Data Movement: • **Data Rate** • **Latency** • **Range**	Moderate latency	Moderate-high latency	Less Latency
Cost	Moderate	Moderate	Formation of tree is time consuming and costly
Management Overhead	Low	Moderate	Moderate

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “Wireless sensor networks: a survey **“,** Computer Networks, vol. 38, pp 393-442, 2002

[2] Anup Pawar et al, “Secure and Efficient Data Transmission in Cluster based Wireless Sensor Network”, IJCSMC, vol. 4, pp 132-142, 2015

[3] Eytan Modiano. “Packet Multiple Access: The Aloha Protocol”, Massachusetts Institute of Technology, PowerPoint file

[4] Katrina LaCurts, “MAC Protocols”, Massachusetts Institute of Technology, Lecture file

[5] Lu Wei, Zhang Longmei, “A Novel Multi-Channel MAC Protocol for Cluster Based Wireless Multimedia Sensor Networks”,Physics Procedia,vol 25,2012, pp. 2203-2210,

[6] Rodellar, Daniel & Bungarzeanu, Cristian & Garcia, Hector & Brisson, Caroline & Küng, Alain & Robert, Philippe. (1998). A comparison between single and multi-channel CSMA/CD protocols of equivalent capacity. 1-8.

[7] Feng Xia, Ruonan Hao, Jie Li, Naixue Xiong, Laurence T. Yang, Yan Zhang,Adaptive, “GTS allocation in IEEE 802.15.4 for real-time wireless sensor networks” ,Journal of Systems Architecture, vol 59, Issue 10, Part D,2013,pp. 1231-1242

[8] Time synchronization in wireless networks by Michael Roche

[9] Wang, Zhaowei et al. “Cluster-Based Maximum Consensus Time Synchronization for Industrial Wireless Sensor Networks”, Ed. Muhammad Imran et al. *Sensors (Basel, Switzerland)* 17.1 (2017): 141. *PMC*. Web. 21 Oct. 2018.

[10] Fikret Sivrikaya, Bulent Yener, “Time Synchronization in Sensor Networks: A Survey”

[11] Sheu, Jang-Ping & Hu, Wei-Kai & Lin, Jen-Chiao. (2008). Ratio-based time synchronization protocol in wireless sensor networks. Telecommunication Systems. 39. 25-35. 10.1007/s11235-008-9081-5.

[12] http://www.srmuniv.ac.in/sites/default/files/files/wnr.pdf

[13] Dr. Baruch Awerbuch, Dr. Amitabh Mishra. “Ad hoc On Demand Distance Vector (AODV) Routing Protocol”, John Hopkins University