**Simulating Network Clustering and Routing Mechanism for IoT Island**

**Fangqi Liu 96121528 Email : fangqil2@uci.edu UciNetIDs: fangqil2**

**Te-Chih Chen 76285429 Email: techihc@uci.edu UciNetIDs: techihc**

**Jia-Wei Chiang 43688514 Email: jiawec5@uci.edu UciNetIDs: jiawec5**

1. **Overview**

In this project, we are exploring data collection and transmission in IoT island, where IoT devices are isolated with outside network (cannot connect to network access point directly) due to lack of network infrastructures. To achieve the connectivity between the IoT devices and the outside network, there are three tasks to solve. First, a clustering method can partition IoT devices into multiple clusters depending on their spatial location and choose the cluster head (CH) which will act as a local sink to gather data from surrounding neighboring nodes. Then, it would elect the rendezvous point (RP) according to the predefined trajectory of mobile sinks (MS) which enables the pre-stored data to be transmitted to MS in a single hop way. Second, we enable buses that act as MSs running on predefined route to collect data from the RPs. The last issue of our project is to establish connection between CH and MS. After accomplishing these tasks, we could fulfill IoT island scenario.

1. **Related Research**

A number of MS-based data gathering approaches in Wireless Sensor Network (WSNs) have been proposed in recent years. The adaptive behavior of MS can minimize the wastage of energy in data forwarding but these advantages are MS path trajectory dependent because sensed buffered data should be gathered within a given time limit especially in delay-sensitive applications. Some MS-based data collection related works are listed as below.

[1] proposed a routing protocol which is based on adaptive and biased sink mobility along with need-based clustering. Firstly, they use mobile sink to traverse from dense region to sparse region according to the density of nodes for ensuring efficient big data collection. Then they use need-based clustering to minimize the data collection time or power consumption of the nodes.

[2] tried to reduce the MS route length in WSNs by considering rendezvous points (RPs) approach. RPs are used to store the data from normal nodes and communicate buffered data to MS. The MS must arrive at stop points to accumulate data from the WSN. This algorithm alleviates the data latency problem by optimizing the density of stop points of MS while ensuring whole network coverage.

[3] utilized Closest Rendezvous Points (CRPs) technique instead of random RPs selection. This approach has shorter data collection path and it is further decreased while multiple mobile elements (MEs) are used.

[4] made MS visit only selected points (which is also termed as RPs) in appropriate time instead of normal nodes or all the nodes. The normal nodes just need to transfer data via multi-hopping to their nearest RPs. Weighted Rendezvous Planning (WRPs) heuristic is used to compute the MS tour that stops at these RPs and weights each node according to its hop distance from the MS path trajectory and the amount of data it forward to its closest RPs.

[5] used a clustering method to minimize the data propagation delay. In this approach, each node maintains a one hop distance from their closest Cluster Head (CH). CHs firstly perform data fusion of cluster nodes data as a collection point (CP) and then the MS reaches all selected CPs for capturing the whole data.

[6] introduced an adaptive route adjustment schema for grid formation for periodic data collection from WSNs. The algorithm finds optimal routes to the latest updated location of the MS and decreases the network overhead. The algorithm also makes use of virtual backbone network (VBN) to timely update the minimal cost route by tracking the MS current location position and its movement.

[7] adopted tree topology along with clustering approach with the help of MS. The algorithm consists of three parts: tree construction (calculating weight of each node according to its residual energy, distance to BS, number of two hop neighbor and average residual energy of one hop neighbor), RP and SRPs selection (choosing highest weight nodes to be RPs and SRPs in respective cluster) and data collection (using MS to gather data from RPs and SRPs).

[8] collected the big data load from a great number of servers using RPs method. MobiCluster protocol here uses the urban buses to carry MS for retrieving sensed data from the remote area of WSNs. Multi-sized clustering mitigates the hot-spot problem and improves network throughput while rendezvous node (RNs) selection guarantees connectivity of sensor domain with MSs and enables timely data collection. The selection process is done based on remaining node energy and Euclidean distance among SNs and the MS.

Our method takes [8] as reference and implements it in the environment of ns-3 simulator. We randomly deploy some clusters of sensor nodes (Stations (STA) in our ns-3 simulation) in a bounded area and select a cluster head (access points (AP) in our ns-3 simulation) for each cluster. Then, we deploy some MSs which move around the area through some predefined routes for retrieving sensed buffered data from APs. Finally, MSs transmit the data gathered from APs to upload points (UPs) once they arrive at any intersection of routes.

1. **Simulation Platform**

We build our simulation on ns-3 simulator which is a discrete-event network simulator. ns-3 has been developed to provide an open, extensible network simulation platform, for networking research and education. It provides models of how packet data networks work and performs and provides a simulation engine for users to conduct simulation experiments.

Our project mainly includes Wi-Fi model, UDP model and mobility model in ns-3 simulator. The WifiNetDevice models a wireless network interface controller based on the IEEE 802.11 standard. The set of 802.11 models provided in ns-3 attempts to provide an accurate MAC-level implementation of the 802.11 specification and to provide a packet-level abstraction of the PHY-level for different PHYs, corresponding to 802.11 specifications.

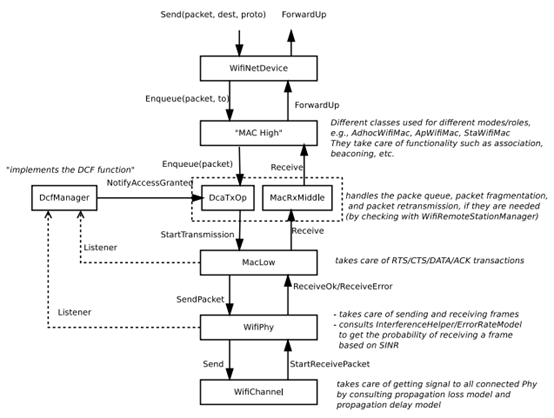


Fig 1: WifiNetDevice architecture

As shown in Fig 1, WifiNetDevice model is consisted of MAC high models, MAC low layer, and PHY layer models. In MAC high models, there are Wi-Fi topological elements - Access Point (AP), non-AP Station (STA), and STA in an Independent Basic Service Set which is also commonly referred to as an ad hoc network. The MAC low layer is split into three main components, ns3::MacLow which takes care of RTS/CTS/DATA/ACK transactions, ns3::DcfManager and ns3::DcfState which implements the DCF and EDCAF functions, ns3::DcaTxop and ns3::EdcaTxopN which handle the packet queue, packet fragmentation, and packet retransmissions if they are needed. The physical layer models are mainly responsible for modeling the reception of packets and for tracking energy consumption.

The mobility models are used to track and maintain the current cartesian position and speed of an object. A number of helper classes which are used to place nodes and setup mobility models (including parsers for some mobility definition formats).

The UDP model is an implementation of the User Datagram Protocol described in RFC 768. It implements a connectionless, unreliable datagram packet service. Packets may be reordered or duplicated before they arrive. UDP generates and checks checksums to catch transmission errors.

1. **Plan of simulation**

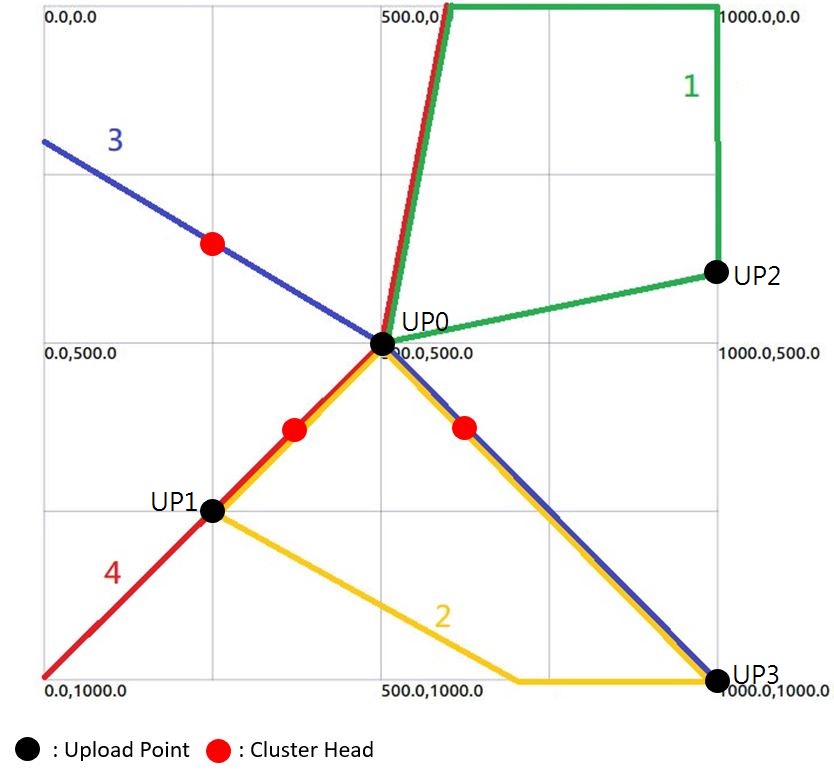


Fig 2: Bus Route Map

Before start to explain our design for ns-3 simulation, let us define some abbreviations to stand for different nodes in our bus route map.

* STA: clustering sensor nodes
* AP: cluster head
* MS: mobile sink
* UP: upload point

We simulate IoT island in a 1000\*1000 area with 4 MSs act as buses. Four MSs have constant routes just like real world (1, 2, 3, 4 shown in Fig 2). Two of them travel in cycle and others turn around while reaching the end point. Each MS would travel in constant speed to exchange packets with APs and transmit data to UPs.

There are three clusters of STA in our ns-3 simulation. Each of them belongs to different local area networks (LANs) and MSs and UPs belong to another LAN. We use UDP protocol for transmitting sensed data and the process of data transmission is “ STA → AP → MS → UP ”. Hence, STA, AP and MS need to send packets while AP, MS and UP work for receiving packets.

Each cluster contains 20 STAs and these STAs work as echo clients in the UDP protocol. For STAs in the same cluster, each of them sends 20-byte packets in a time interval of 5 seconds through an UDP connection to its dedicated AP.

Each MS calls the function “showPosition” every second for determining whether it is close enough to any AP or UP. If it discovers that it has entered the range of an AP, it has AP call “sendpacket” function. Then, AP starts to send buffered data which it received from STAs to the MS. On the other hand, if MS discovers that it has entered the range of an UP, it calls “sendpacket” function by itself and start to send data received from APs to the UP.

Each AP, MS, or UP creates a socket and binds the socket to its IP address before starts to receive packets from any other node in the simulation. Once completed all the pre-works for receiving packets, AP call “ReceivePacket” function, MS call “msReceivePacket” function and UP call “Upload” function to receive data from STA, AP and MS, respectively. The function listed are all callback functions and their works include copying data from received packet to their buffer whose capacity is unlimited, updating the size of data in the buffer and recording number of received packets.

1. **Results**

In the experiment, sensors send packets through mobile sinks to upload points. In order to observe packet loss rate and the delay time with different upload point setting, we change the number and location of upload points. Fig 3 is the screenshot while mobile sink is transmitting packets to upload point. Table 1 shows the simulation result.

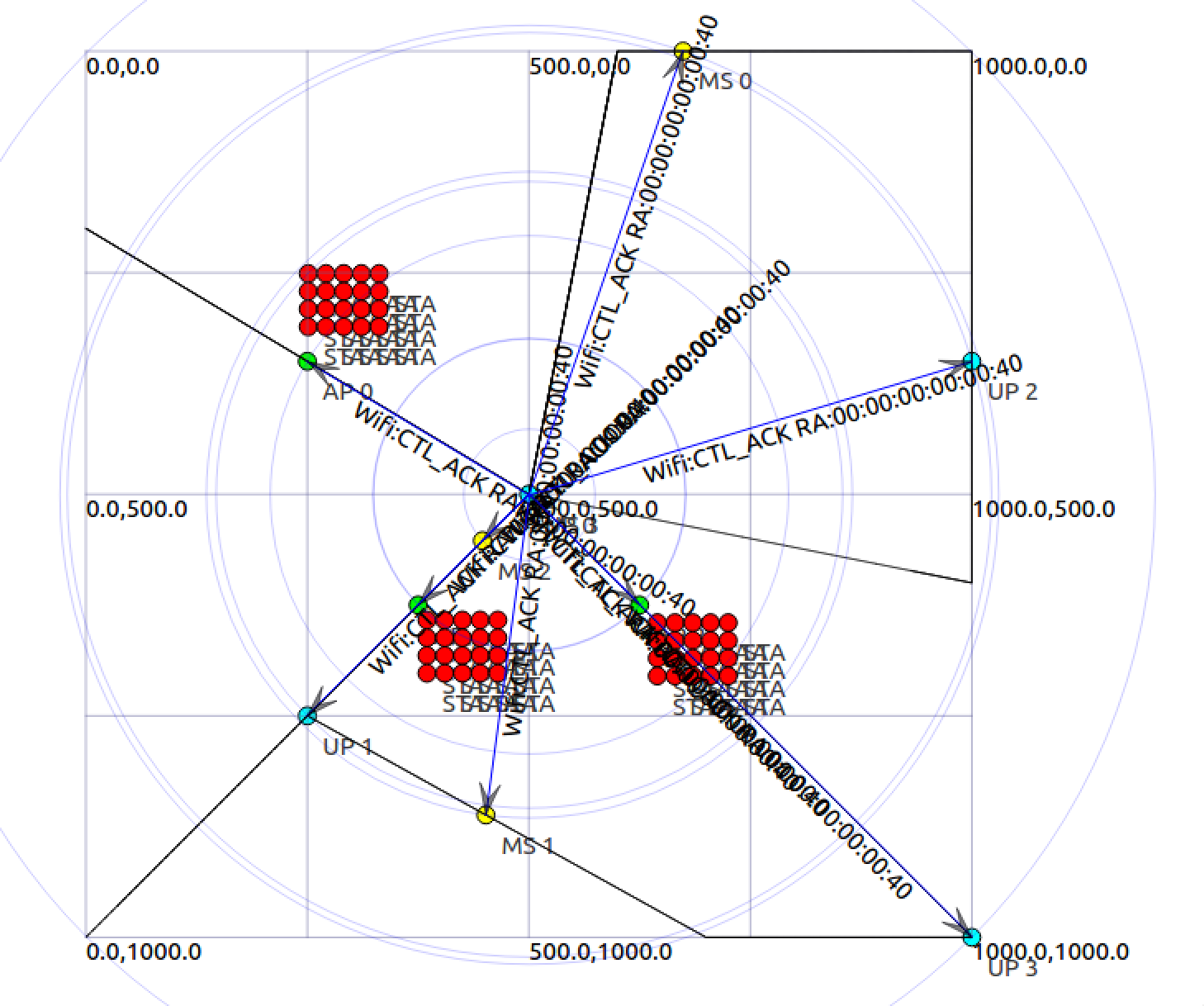


Fig 3: mobile sink sends packets to upload point

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UP numbers | Packets received at UP0 | Packets received at UP1 | Packets received at UP2 | Packets received at UP3 | Average delay time | Packet loss rate |
| 0 | 2280 | 0 | 0 | 0 | 347.481 | 0% |
| 1, 2 | 0 | 800 | 0 | 0 | 148.954 | 65% |
| 2, 3 | 0 | 0 | 0 | 1840 | 273.422 | 20% |
| 0, 1, 2 | 1480 | 800 | 0 | 0 | 232.867 | 0% |
| 1, 2, 3 | 0 | 800 | 0 | 1480 | 230.131 | 0% |
| 0, 1, 2, 3 | 840 | 800 | 0 | 640 | 169.709 | 0% |

Table 1 : simulation result

In general, we could tell that while we add more upload points, the delay time and packet loss rate would decrease. Furthermore, the locations of the upload points are crucial. UP0 is more essential than other upload points because it locates at the central in our map and every bus routes would go through it. When we remove UP0, it causes packet loss and increment in delay time. On the contrary, UP2 does not receive any packets at any time. It seems to be useless due to the lack of sensor surround it. Since the UPs cover every route, as the number of UP increases, delay time would decrease.

1. **Future Works**

In our ns-3 simulation, we make MSs only move around within a bounded area we demonstrated in section 4. However, this deployment of MSs’ routes is actually too ideal to happen in the real world. As a matter of fact, buses carrying MSs are possible to move away from the bounded area and the data it collected from APs may not be uploaded to any UP before it leaves. In order to address the problem of failing to collect all sensed data, we need to put UP at every exit that buses leave from the bounded area. Knowing that it costs more with more UPs, we have to come up with a solution that can minimize the number of UPs on the bus route map while meet the requirement of transmitting sensed data in an acceptable delay time in the future.

Moreover, mobile sinks have unlimited buffer in our simulation which won’t happen in real world. Additionally, data is heterogeneous and be prioritized according to its importance or temporal effectiveness in the real world. The priority of a piece of data may change in its lifetime and usually the data gets higher priority if it has less time to live. When the buffer of a MS is not unlimited, which is more closed to the condition in the real world, the impact of priority of data can be tremendous. MS tend to choose and carry data with higher priority and drop data with lower priority whenever its buffer is full. Therefore, considering the influence of prioritized data and limited buffer capacity, we also need to design methods of transmitting sensed data more carefully in our future works.

1. **References**

[1] A.H. Khan et al, “Dynamic joint sink mobility with need-based clustering in wsns,” International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.320-325, March 2015.

[2] J. Xu, L. He, Z. Chen, G. Huang, and T. Yuan, “Reducing the path length of a mobile bs in wsns,” IEEE International Seminar on Future BioMedical Information Engineering, pp. 241-274, December 2008.

[3] A. Alomari, N. Aslamy, W. Phillips, and F. Comeau, “A scheme for using closest rendezvous points and mobile elements for data gathering in wireless sensor networks,” IFIP Wireless Days, pp. 1-6, November 2014.

[4] H. Salarian, K. Chin, and F. Naghdy, “An energy-efficient mobile-sink path selection strategy for wireless sensor networks,” IEEE Trans. on Vehicular Technology, vol. 63, pp. 2407-2419, June 2014.

[5] Prashanth and S. V. Nandury, “Cluster-based rendezvous points selection for reducing tour length of mobile element in WSN,” IEEE International Advance Computing Conference IACC, pp. 1230-1235, June 2015.

[6] A. W. Khan, A. H. Abdullah, M. A. Razzaque, and J. I. Bangash, “A virtual grid-based dynamic routes adjustment scheme for mobile sink- based wireless sensor networks,” IEEE Sensor Journal, vol. 15, pp. 526- 534, January 2015.

[7] C. Zhu, S. Wu, G. J. Han, L. Shu, and H. Wu, “A tree-cluster-based data gathering algorithm for industrial wireless sensor networks with mobile sink,” IEEE Access, vol. 3, pp. 381-396, April 2015.

[8] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, “Rendezvous-based approach enabling energy-efficient sensory data collection with mobile sinks,” IEEE Trans. On Parallel and Distributed Systems, vol. 23, pp. 809-817, May 2012.