P2P Based Data Management in WSNs: Experiences and Lessons Learnt from a Real-World Deployment

SYEDA NIDA SAHAR*, FAISAL KARIM SHAIKH**, AND SANA HOOR JOKHIO**

RECEIVED ON 05.09.2011 ACCEPTED ON 01.10.2011

ABSTRACT

WSN (Wireless Sensor Network) has attracted wide attention from researchers over the recent years, with a major focus on the issues of data transport and management. Many traditional data management protocols are being developed for WSN and are evaluated using simulations only. Unfortunately there are very less protocols which are evaluated on real testbeds. Recently, P2P (Peer to Peer) protocols are gaining much attention to be utilized by WSNs since both have many similarities. However, a WSN faces resource constraints in terms of limited power supply, small storage capacity, low computing power and connectivity issue. Contrary, DTM (Data Transport and Management) protocols developed for P2P systems cause inefficient use of resources in a WSN. Lately, a VCP (Virtual Cord Protocol) has been proposed that uses P2P techniques for information management in a WSN and provides the efficient resource utilization. In this paper, we present the implementation and deployment of VCP to show its performance on a real deployment scenario using Mica2 motes.

Key Words: **Wireless Sensor Networks, Data Transport and Management, Peer to Peer.**

1. INTRODUCTION

A WSN consists of sensor nodes having scarce resources, which makes the DTM a challenging task [1]. Since, WSNs are naturally data centric with limited resources, the data management in a WSN is different from the other communication networks. Accordingly, well organized data management and data transport are important in a WSN. A lot of research has been done to increase the efficiency of data transport [2,3] and data management [4,5] in WSNs. These protocols are majorly evaluated through simulations only. It is well established in WSN community that simulation environments do not reflect the real deployment conditions [6]. Thus, recently the focus is on evaluating the protocols on testbed and in real deployment scenarios.

The client-server approaches and end to end communication systems are unsuitable for the characteristics of WSNs. Currently, P2P systems are gaining much attention due to their good data management characteristics. There are many similarities between P2P

* Post-Graduate Student, Department of Communication Engineering, Mehran University of Engineering & Technology, Jamshoro.
** Assistant Professor, Department of Communication Engineering, Mehran University of Engineering & Technology, Jamshoro.
*** Assistant Professor, Department of Computer Systems Engineering, Mehran University of Engineering & Technology, Jamshoro.

systems and WSNs, e.g. each node in the network shares other node's resources, does not have permanent connectivity, has equal responsibilities and is fully independent regarding its own resources, has separate ID and behaves as a router when forwarding the data. Structured P2P systems that rely on DHTs (Distributed Hash Tables) [7] seems to overcome these problems as it uses decentralized approach and provide O(1) complexity for storing and retrieving data. The protocols that use DHTs approaches are commonly used in WSNs [8,9]. A structured P2P system provides a link between content and the address of the node. These links are stored in the DHTs. The DHTs manage data on the network in such a way that it could be easily stored and retrieved. Each node in the network has only a partial view of the whole network, and it utilizes this partial view when it has to forward the data towards the destination. But structured P2P systems use underlying routing schemes which cause inefficient use of resources in WSNs. To resolve this problem, DHTs and underlying routing techniques are combined to form virtual coordinate systems. Most of these algorithms are quite complicated and only few are implemented on real platform. Chord [10] and VRR (Virtual Ring Routing) [11] are well known examples that use DHTs services. GHTs (Geographic Hash Tables) [12,13] are also used in structured P2P that provides optimal scalability using GPSR (Greedy Perimeter Stateless Routing) for data transport. It has drawbacks that it produces a data refresh message even if there is no change in topology and it requires physical locations of nodes too. GRWLI (Geographic Routing Without Location Information) [14] overcomes the drawback of GPSR by using the virtual position of nodes instead of physical location, but it has an unavoidable communication overhead for creating a virtual position. All these protocols have different drawbacks, but one common drawback is an inefficient use of the resources when implemented in WSNs [13] as these protocols are basically designed for P2P systems having plenty of recourses.

To overcome the drawbacks mentioned above, one of the prominent approaches for DTM in WSN using the P2P approach is the VCP (Virtual Cord Protocol) [4]. The Chord and VRR have drawbacks that one hop in the virtual networks usually implies several hops in the real networks and they have complex and large routing table. These drawbacks have been covered in the VCP by providing same hops in the real and virtual networks and simple routing table. The VCP overcomes the drawback of GRWLI by providing an avoidable communication overhead for assigning virtual positions. In the VCP, each node is responsible to forward a packet/message to its single hop neighbors only hence number of transmissions in the network is distributed among the all nodes in the network. In this way the resource of power is efficiently utilized. The simulation results in [13] suggest that VCP has a low communication overhead, uses adequate routing paths and performs better than the other routing protocols.

Generally, in a WSN the field trials or real world/testbed results have been more reliable than the simulation studies. As the sensor nodes need to interact with the environment having numerous parameters that are difficult to simulate. This motivates us to deploy VCP on a real scenario and to verify the simulation results. In this paper, we highlight the implementation issues and the deployment strategies for VCP on the widely accepted Mica2 sensor node platform.

The rest of the paper is organized as follows. Section 2 details, the VCP and its different processes. In Section 3 VCP implementation, real world deployment and the implementation issues are discussed. The experimental results are highlighted in Section 4. We conclude our findings in Section 5.

2. VIRTUAL CORD PROTOCOL

In this section, we present a brief overview of the VCP protocol [4]. Next, we describe how the network is formed by VCP and how the routing tables are updated for DTM.

2.1 V CP Overview

VCP is inspired by structured P2P technology and provides efficient data management using the routing techniques with DHT services [4, 14]. It uses virtual positions for data transport, where each node maintains a successor, a predecessor and a routing table to store information of its neighbors. Each node in the network can have a maximum of two virtual neighbors and can broadcast a message to update the other nodes about its current status and to assign virtual positions to the new nodes. All the nodes are responsible to hold information of their single hop neighbors only, so the functionality of each node would not be affected by the size of the network.

2.2 Building a Network

Each node in the network is assigned a virtual position by a node that is already part of the network from a predefined range between S (Start Point) and E (End Point). During the network setup phase, it is necessary that a node is preprogrammed as an initial node and assigned a virtual position S or E to start the network. Each node that becomes a part of the network, starts broadcasting a HELLO message to update the other nodes about its status and to assign the virtual positions to the new nodes joining the network. If a new node added in the network receives a HELLO message, it sends a request to the old node to set its virtual position in the network. In VCP implementation, we have used S equal to 0 and E equal to 1. The preprogrammed initial node is assigned a virtual position 0.

2.3 VCP Data Management

VCP data transport is carried out using the single hop neighbors. If the source and the destination nodes cannot directly communicate with each other, then the data will be sent to a node having the closest virtual position to the destination as shown in Fig. 1. Considering the Fig. 1 the data storage process is as follows:

- Let us consider a node 0.5, which senses a data P.
- P will be hashed and let us suppose hashed value of P is 0.9.
- Node 0.5 searches in its routing table for a closest neighbor to the hashed value 0.9 and finds the node 0.75.
- Node 0.5 sends a request to node 0.75 for storing P (shown with green arrow in Fig. 1).
- Node 0.75 receives the request and searches in its routing table for a closest neighbor to the hashed value 0.9. Node 0.75 finds the node 1.0.
- Node 0.75 sends a request to node 1.0 for storing P (shown with green arrow in Fig. 1).
- Node 1.0 receives the request and searches in its routing table for a closest neighbor to the hashed value 0.9. But at node 1.0, no more routing is possible because its own virtual position is the one closest to the hashed value 0.9. Hence, node 1.0 stores P.

Similarly, considering the Fig. 1 the data retrieval process works as follows:

- Let us consider now a node 0.0, which requires a data D.

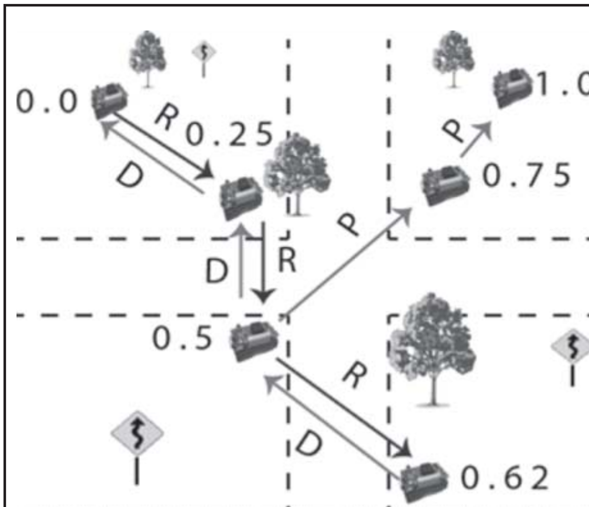
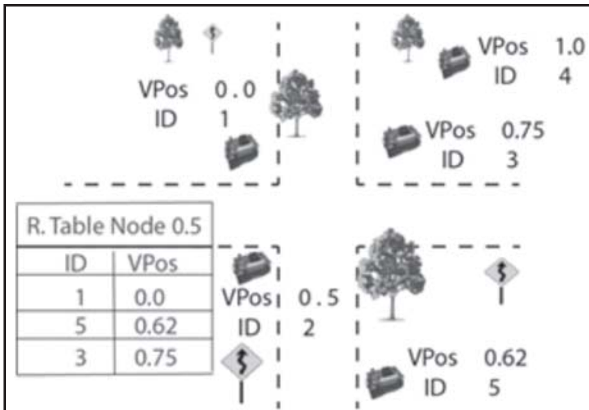


FIG. 1: VCP DATA MANAGEMENT

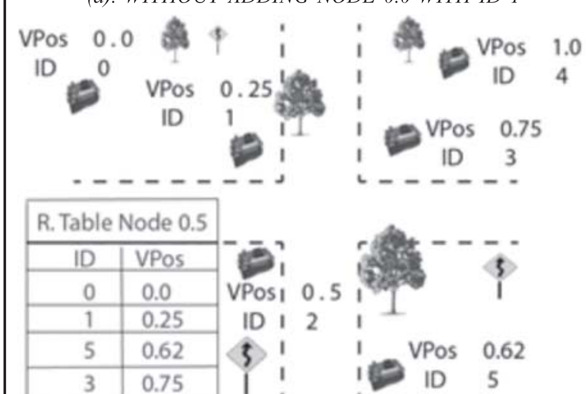
- D will be hashed and suppose the hashed value is 0.66.
- Node 0.0 searches in its routing table for a closest neighbor for the hashed value and finds the node 0.25.
- Node 0.0 sends a request to node 0.25 for retrieving D (shown with the blue arrow in Fig. 1).
- Node 0.25 receives the request and searches in its routing table for a closest neighbor to the hashed value 0.66. Node 0.25 finds the node 0.5.
- Node 0.25 sends a request to the node 0.5 for retrieving D (shown with the blue arrow in Fig. 1).
- Node 0.5 receives the request and searches in its routing table for a closest neighbor to the hashed value 0.66. Node 0.5 finds the node 0.62.
- Node 0.5 sends a request to the node 0.62 for retrieving D (shown with the blue arrow in Fig. 1).
- Node 0.62 receives the request and searches in its routing table for a closest neighbor to the hashed value 1.0. But, no more routing is possible because its own virtual position is the closest one to the hashed value 0.66. Therefore, node 0.66 searches for the D in its memory.
- Node 0.62 responses to node 0.0 following the route 0.5-0.25-0.0 (shown with the pink arrow in Fig. 1).

2.4 VCP Routing Table Update

To route the message accurately from source node to destination node, each node must update its routing table. As the virtual position is being changed by a node independently, it is vital for each node to update its routing table. Each node broadcasts a HELLO message to update the other nodes about its current status. Fig. 2 demonstrates the routing table updates process. The Fig. 2(a) shows routing table of Node 0.5 without adding a Node 0 in the network. The Fig. 2(b) shows routing table of Node 0.5 after adding a Node 0 in the network. Let us consider a Node 0.5 which receives a HELLO message from Node 0.25 with ID 1. Node 0.5 searches in its routing table that a node with ID 1 is present or not (Fig. 2(a)). Since, ID 1 is present in the routing table the Node 0.5 checks either it has same virtual position 0.25 or not. Node 0.25 with ID 1 has virtual position 0.0 in routing table of node 0.5 (Fig. 2(a)) hence node 0.5 updates its current virtual position to 0.25 in its routing table (Fig. 2(b)). For adding a new node in the routing table, suppose a Node 0.5 receives a HELLO message from Node 0.0 with ID 0. Since Node 0.0 with ID 0 is not present in the routing table of Node 0.5 (Fig. 2(a)), Node 0.5 adds Node 0.0 with ID 0 in its routing table (Fig. 2(b)).



(a). WITHOUT ADDING NODE 0.0 WITH ID 1



(b). WITH ADDING NODE 0.0 WITH ID 1

FIG. 2. VCP ROUTING TABLE UPDATE

3. REAL WORLD SCENARIO DEPLOYMENT

In this section, we first discuss the hardware and software details regarding implementation of VCP. Next, the deployment scenario is described to evaluate VCP. Later, the implementation and execution details of VCP are presented. In last we discuss the various issues encountered during the implementation and deployment of VCP.

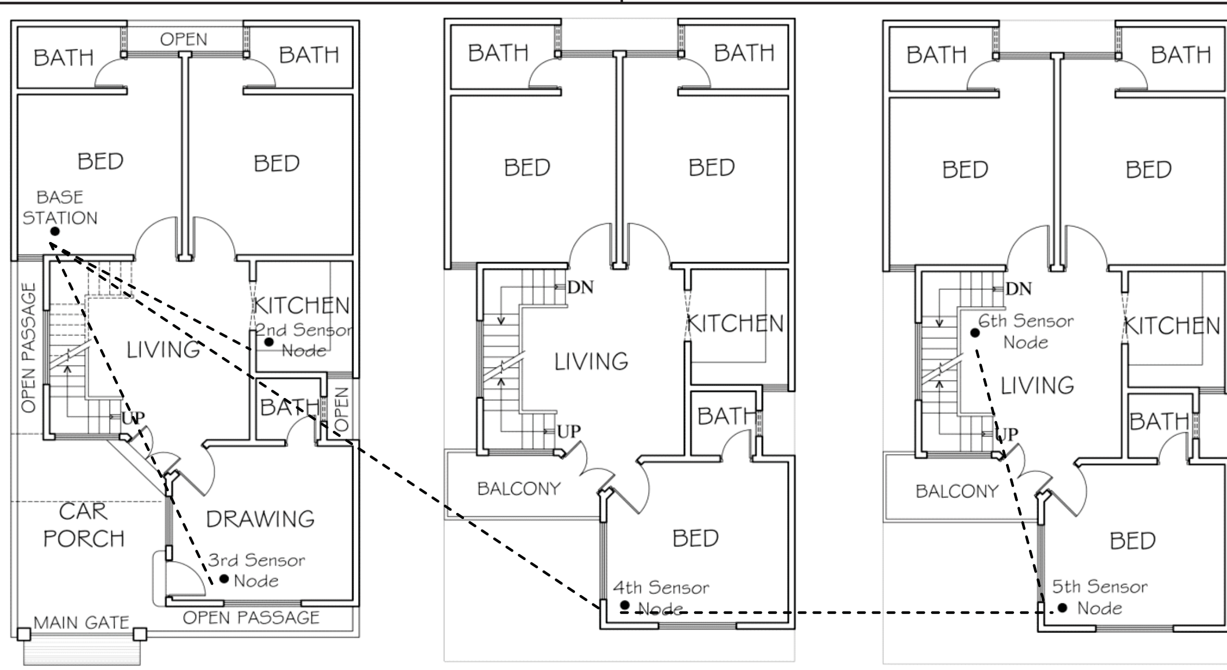
3.1 Hardware and Software

Six Mica2 sensor nodes have been used for the implementation of VCP. One of the sensor nodes is attached with the MIB510 programming board, considered as a base station node. Whereas, the remaining five sensor nodes are randomly distributed to accomplish the scenario. TinyOS-1.1.15 is used to program the sensor nodes.

3.2 Deployment Scenario

Fig. 3 depicts a scenario where sensor nodes are deployed inside a house to measure the light readings. During the sunlight/bright conditions the sensor nodes sense light and accordingly switch off the lights. When the sensor nodes read light values below certain threshold, i.e., cloudy weather or in night, the lights are switched on. In case when some lights are not working properly, sensor nodes detect darkness and inform the base station for quick action. Since, VCP is based on structured P2P and the data is replicated on several nodes, the base station can easily retrieve the required light reading by sending query to different nodes.

Among six Mica2 motes, three motes are deployed on the ground floor including the base station. One mote is deployed at the first floor and remaining two motes are deployed at the second floor as shown in the Fig. 4. Connectivity of sensor nodes with the base station is shown with the help of dotted lines in Fig. 4. The base station is placed in the bedroom on ground floor (Fig. 3 (a)), the second node is in the kitchen at approximately 20 feet away from the base station and the third node is in the drawing room at approximately 45 feet away. Both the second and the third sensor nodes are located at a single



(a) GROUND FLOOR (b) FIRST FLOOR (c) SECOND FLOOR

FIG. 3. VCP DEPLOYMENT SCENARIO

hop. The fourth sensor node is placed in the bedroom at approximately 48 feet away from base station (Fig. 3 (b)). In Fig. 3(c) (Second Floor), the fifth sensor node is placed in the bedroom at approximately 16 feet away the fourth node. The hop counts for this node is 2, as it is in the communication range of the fourth node towards the base station. The sixth node is placed in another bedroom (on the same floor, i.e. 2nd floor) at approximately 30 feet away from the fifth node. For this node, the hop count is 3 and it is in the communication range of the fifth node only.

3.3 Prototype Implementation

The unique source address to each node is assigned and the nodes that are not part of the network are assigned a virtual position -1.0. Each node that becomes part of the network starts broadcasting a HELLO message after every second along with the other network functions including sensing light from the environment and maintaining the routing table. A variable SET_POS is used to check the status of the node to determine if the node is busy in setting the virtual position or not. This variable prevents the assignment of same virtual positions to multiple nodes and it also prevents assignment of two different virtual positions to a single node. The virtual positions are allocated between value S and E as mentioned in the Section 2.2. Once the virtual positions are assigned, the data transmission is carried out using the single hop neighbors, as discussed in the Section 2.3. Table 1 presents a pseudo code for data storage and retrieval processes.

Although, VCP shows the good performance in the simulation results, the hardware implementation of the protocol highlighted a number of issues. They are briefly discussed in the next section.

3.4 Encountered Problems during Implementation

During the VCP implementation certain problems were encountered which are inherent in the original proposed protocol. These issues are discussed below:

TABLE 1. DATA STORAGE AND DATA RETRIEVE PROCESSES.

Data Storage	Data Retrieve

float m,o;int n;
 	
if(host<= Neighbor_Table[1[3])
if(Neighbor_Table[1][3] != Pos)
Rout_Data at Neighbor_Table[1][1]
else
Write_Data in memory
 	
if(host>= Neighbor_Table[Index][3])
if(Neighbor_Table[Index][3] != Pos)
Rout_Data at Neighbor_Table[Index][1]}
else
Write_Data in memory
 	
if((host > Neighbor_Table[1][3]) && (host < Neighbor_Table[Index][3])){
for(n=2 ; n<Index ; n++)
if(host> Neighbor_Table[n][3]){
m =(Neighbor_Table[n][3]-host);
o = (host-Neighbor_Table[n-1][3]);
if(m > o)
Rout_Data at Neighbor_Table[n-1][1]
else
Rout_Data at Neighbor_Table[n][1]
	float m1,o1;int n1;
	
	if(host<= Neighbor_Table[1[3]){
	if(Neighbor_Table[1][3] != Pos){
	Rout_Data at Neighbor_Table[1][1] }
	else
	Search data in memory;
	
	if(host>=Neighbor_Table[Index][3])
	if(Neighbor_Table[Index][3] != Pos)
	Rout_Data at Neighbor_Table[Index][1]
	else
	Search data in memory
	
	if((host>Neighbor_Table[1][3]) && (host< Neighbor_Table[Index][3]))
	for(n=2 ; n<Index ; n++)
	if(host> Neighbor_Table[n][3]){
	m = (Neighbor_Table[n][3]-host);
	o = host-Neighbor_Table[n-1][3]);
	if(m > o)
	Rout_Data at Neighbor_Table[n-1][1]
	else
	Rout_Data at Neighbor_Table[n][1]

- Once, the preprogrammed initial node leaves its virtual position and assigns S to another node, and if this node reboots itself then its virtual position again becomes S and it starts assigning positions from the beginning. As a result, all the previous virtual positions assigned by this node and by other nodes (that were depending on the virtual position of this node) become inaccurate and the unique virtual position assignment operation fails.
- When a new node receives a HELLO message from a node, which is not S or E, and it already has two virtual neighbors (maximum limit of virtual neighbors) then new node's virtual position remains unset and its resources cannot be utilized.
- The VCP joining process works efficiently when the new nodes are added one by one in the network. If more than one new node is added, after receiving a HELLO message, the chances of assigning the same virtual positions to the new nodes are approximately 70% (obtained by simultaneously adding more than one new node many times). This results in inaccurate virtual position assignment.
- During deployment of the sensor nodes, it was observed that since the base station was supplied with a direct power supply, and all the other nodes were operating on batteries, the base station had higher transmit power than all the other nodes. This creates a major problem for the data transport (routing), since after hearing a HELLO message sensor node update their neighbor tables. The HELLO message of the base station was heard by the majority of the sensor nodes in the network. These sensor nodes, on receiving a HELLO message, try to communicate with the base station by directly sending the data packets to the base station, but due to low power their transmission range is shorter than that of the base station leading to a packet loss.
- It is also observed that the MAC layer collisions also result in packet losses.

Certain issues of VCP can be adjusted during deployment phase e.g., nodes can be added one by one in the network to assign unique virtual position to each node and nodes must be placed in such a way that it can receive a HELLO message of at least two nodes so that its position can be set.

4. RESULTS

We evaluate the performance of VCP for reliability and overhead for joining, data storage and data retrieval processes. The results are obtained by repeating experiments many times and then taking the average values.

4.1 Reliability

The reliability of the VCP is analyzed by sending a packet from each node in the network to the base station at the interval of one second. The graph (Fig. 4) shows the number of packet received versus time in seconds at the base station. The PRR can be defined as the rate at which the packets are received from different hops to the base station, or the ratio of the number of packets received at the base station to the number of packets sent from the nodes.

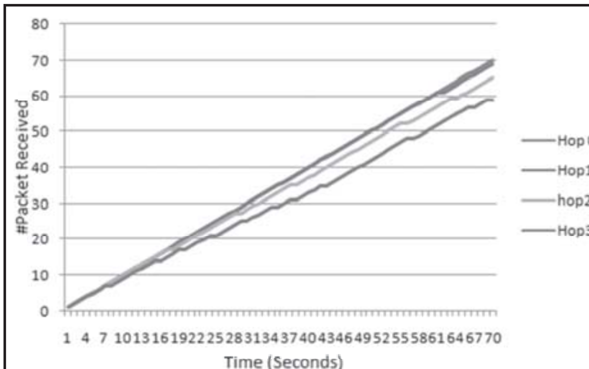


FIG. 4. PACKET RECEIVED RATE VS TIME

From the Fig. 4, it is clear that at hop 0 the PRR is 100%, since every second a packet is generated so at time interval 70, 70 packets are received by the sink which provides 100% PRR. Similarly, at hop 1 the PRR is 99%, at the Hop 2 it is 93% and at hop 3 the PRR is 84%. This shows that as the number of hops is increased the reliability of VCP is decreased this is mainly due to MAC collisions.

4.2 Overhead for Joining Process

Fig. 5 shows the number of transmissions required to set a new node's virtual position in the network. It is observed that Node ID 0 requires no transmission for setting its virtual position as it is a preprogrammed node and its virtual position is preset. Furthermore, if the new node is having only one neighbor than it requires 3 transmissions whereas if two neighbors are present it requires 6 transmissions to acquire virtual position. Thus, for the joining process the overhead remains constant.

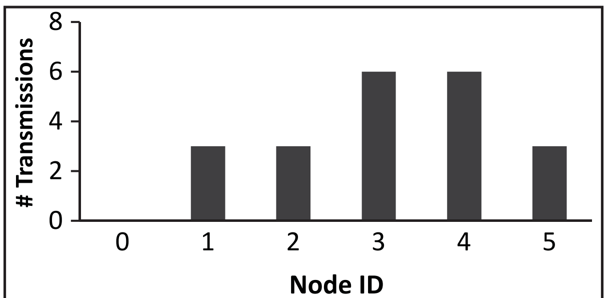


FIG. 5. VCP JOINING OVERHEAD

4.3 Overhead for Data Storage Process

The number of transmissions required to store the sensed data depends on the number of nodes present in the network and the hashed value of the sensed data. Accordingly, different nodes are responsible for storing a range of the hashed values. Fig. 6 depict the relationship between numbers of transmissions required to store the different hash values. It is observed that if a node with lower virtual position sense a value whose hash number is higher than it requires more transmissions to store the data. Alternatively, if the sensed hash value is smaller it requires fewer transmissions.

4.4 Overhead for Data Retrieval Process

In majority of cases the base station requires to retrieve the sensed data. Accordingly, Table 2 depicts the number of transmissions required for different ranges of the hashed values after adding 6 nodes in the network.

TABLE 2. TRANSMISSIONS REQUIRED FOR RETRIEVING DATA AT BASE STATION

Hashed Value	No. of Transmissions
0.000-0.250	0
0.251-0.900	1
0.901-0.965	2
0.966-1.000	3

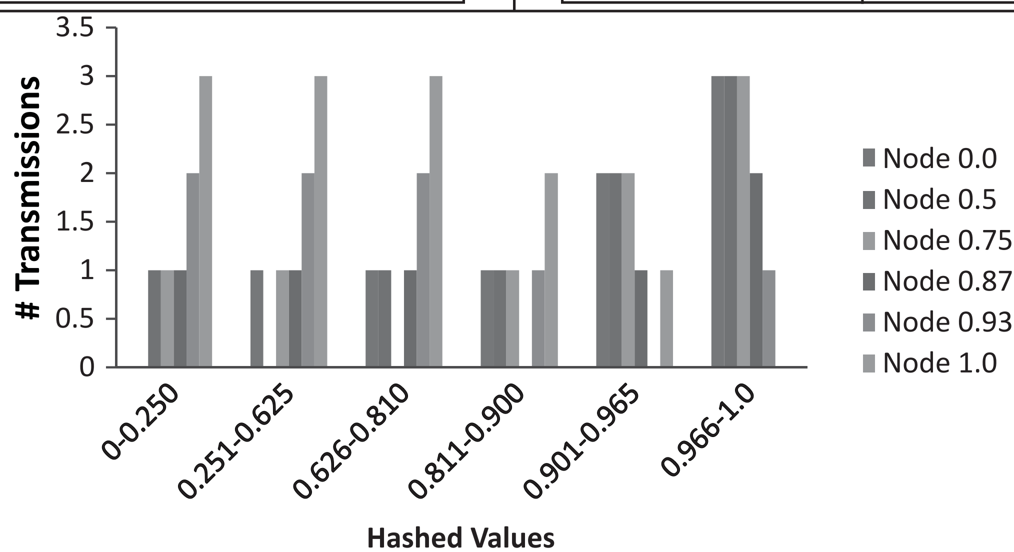


FIG. 6. NUMBER OF TRANSMISSIONS VS HASHED (SIX NODES IN N/W)

As a base station is at the single hop to nodes 0.5, 0.75 and 0.875. Therefore, it requires only one transmission for retrieving data from these nodes as a range from 0.251 to 0.900 is assigned to these three nodes. For retrieving data from node 0.937 base station requires two transmissions as it is at hop 2 and from node 1.0 it requires three transmissions as it is at hop 3.

4.5 Observations

Generally, it is observed that the performance of VCP on real deployment scenario is not as accurate as it was mentioned in the simulation study [3]. In simulation, number of packets received at the base station from different hops were 100% while in real scenario the number of packets received at the base station are affected by the number of hops as mentioned in the Section 4.1.This is mainly because of the collisions at the MAC layer and due to the unavailability of reliability mechanisms which are not considered in VCP.

5. CONCLUSION

In this paper we have implemented VCP on Mica2 motes to demonstrate the applicability of the P2P approach on WSNs for efficient DTM. The implementation process unveils the problems in the VCP and provides insights to overcome them. The results expose the need of reliability mechanisms to improve the performance of the protocol. Since, VCP is designed for static WSNs, our future work involves improving VCP design for mobile WSNs and to integrate the reliability mechanisms.

ACKNOWLEDGEMENTS

This research is supported in parts by Mehran University of Engineering & Technology, Jamshoro, Pakistan.

REFERENCES

[1] Akyildiz, I.F., and Vuran, M.C., "Wireless Sensor Networks", John Wiley and Sons, 2010.

[2] Shaikh, F.K., Khelil, A., Ayari, B., Szczytowski, P., and Suri, N., "Generic Information Transport for Wireless Sensor Networks", Proceedings of SUTC, pp. 27-34, 2010.

[3] Shaikh, F.K., Khelil, A., Ali, A., P., and Suri, N., "TRCCIT: Tunable Reliability with Congestion Control for Information Transport in Wireless Sensor Networks", Proceedings of WICON, pp. 1-9, 2010.

[4] Awad, A., Sommer, C., German, R., and Dressler, F., "Virtual Cord Protocol (VCP): A Flexible DHT-like Routing Service for Sensor Networks", Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pp. 133-142, Atlanta, GA, 2008.

[5] Madden, S.R., et. al., "TinyDB: An Acquisitional Query Processing System for Sensor Networks", ACM Transactions on Database Systems, Volume 30, No. 1, pp. 122-173, 2005.

[6] Langendoen, K., "Apples, Oranges, and Testbeds", Proceedings of IEEE International Conference on Mobile Ad Hoc and Sensor Systems, pp. 387-396, 2006.

[7] Dabek, F., "A Distributed Hash Table", Ph.D. Thesis, Massachusetts Institute of Technology, September, 2005.

[8] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., and Balakrishnan, H., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", Proceedings of ACM SIGCOMM, San Diego, CA, 2001.

[9] Caesar, M., Castro, M., Nightingale, E.B., O'Shea, G., and Rowstron, A., "Virtual Ring Routing: Network Routing Inspired by DHTs", Proceedings of ACM SIGCOMM, pp. 1-8, Pisa, Italy, 2006.

[10] Ratnasamy, S., et. al., "GHT: A Geographic Hash Table for Data-Centric Storage", Proceedings of ACM International Workshop on wireless Sensor Networks and Applications, pp. 1-8, Atlanta, GA, 2002.

[11] Ratnasamy S., et. al., "Data-Centric Storage in Sensornets with GHT, A Geographic Hash Table", Proceedings of ACM/ Springer Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks, 2003.

[12] Rao, A., et. al., "Geographic Routing Without Location Information", Proceedings of MobiCom, pp. 14-19, San Diego, CA, USA, 2003.

[13] Awad, A., German, R., and Dressler, F., "Efficient Service Discovery in Sensor Networks using VCP", Proceedings of ACM International Conference on Mobile Computing and Networking, Demo Session, Beijing, China, 2009.

[14] Awad, A., German, R., and Dressler, F., "P2P-Based Routing and Data Management using the Virtual Cord Protocol (VCP)," Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 443-444, Hong Kong, China, 2008.