Gao, Longxiang and Li, Ming 2009, Multi-level virtual ring : a foundation network architecture to support peer-to-peer application in wireless sensor network*, in ATNAC 2009 : Proceedings of the 2009 Australasian Telecommunication Networks and Applications Conference*, IEEE, Piscataway, N. J., pp. 1-6.

Multi-level Virtual Ring: a Foundation Network Architecture to Support Peer-to-Peer Application in Wireless Sensor Network

Longxiang Gao and Ming Li
School of Information Technology
Deakin University, Melbourne Campus
221 Burwood Hwy, Burwood, VIC 3125
Email: lxg@deakin.edu.au

***Abstract*—Two main problems prevent the deployment of peer-to-peer application in a wireless sensor network: the index table, which should be distributed stored rather than uses a central server as the director; the unique node identifier, which cannot use the global addresses. This paper presents a multi-level virtual ring (MVR) structure to solve these two problems.**

The index table in MVR is distributed stored by using the DHT technique. MVR is constructed decentralized and runs on mobile nodes themselves, requiring no central server or interruption. Naming system in MVR uses natural names rather than global addresses to identify sensor nodes. The MVR can route directly on the name identifiers of the sensor nodes without being aware the location. Some sensor nodes are selected as the backbone nodes by the backbone selection algorithm and are placed on the different levels of the virtual rings. MVR hashes nodes' identifiers on the virtual ring, and stores them at the backbone nodes. Furthermore, MVR adopts cross-level routing to improve the routing efficiency.

Experiments using ns2 simulator for up to 200 nodes show that the storage and bandwidth requirements of MVR grow slowly with the size of the network. Furthermore, MVR has demonstrated as self-administrating, fault-tolerant, and resilient under the different workloads.

***Index Terms*—peer-to-peer, name routing, DHT, sensor network, multi-level virtual ring**

I. Introduction

THE demand of peer-to-peer applications in wireless sensor network is high and expected in near future, because the wireless sensor network is easy to be deployed: no network administration is required when nodes join or leave the network[1]. The peer-to-peer application in wireless sensor network requires a distributed index system [2] with unique name identifier and efficient searching and routing scheme.

The distributed index system is used to hash nodes' identifiers on the virtual ring, and uses Distributed Hash Table (DHT) [3], [4], [5], [6] techniques to store them at key nodes. The name identifier system is designed to identify a node into a certain size digital hashed value. Currently, lots of routing schemes are using the geographic address identify system to support searching and routing packets, the address routing schemes in sensor network need to deploy servers or beacons [7], [8] to allocate addresses for sensor nodes. It is suitable for the peer-to-peer application in the wired network, however it is impossible in wireless sensor network, if we consider the mobility character of wireless sensor network. By analyzing these characters, we believe the new peer-to-peer routing scheme should be based on name identifier rather than address.

We analyze MVR, show that it is correct and efficient, and present simulation results to support our analysis by using *ns2* [9] simulator. MVR does not only using the cross-level routing to improve the routing efficiently, but also being scalable in the following senses:

1) Nodes in sensor network does not need other global information to identify their names.
2) The searching process should be efficient, without address identifier and broadcast.
3) The routing process should be smart, without going through every node along the route.

The rest of the paper describes the design and simulated performance of MVR. Section 2 reviews the existing related schemes in sensor network. Section 3 describes the structure and algorithms of MVR in details. Section 4 studies the simulation results of MVR. Section 5 suggests areas for future research. Section 6 summarizes the paper's contributions.

II. Related work

Currently, a variety of peer-to-peer applications have been widely deployed and used in Internet, such as Skype[10]. In these applications, files are stored at the peers rather than at a central server and, as opposed to the traditional client-server model, files are transferred directly between peers. Unfortunately, most of the current peer-to-peer systems are not scalable. For example, in Skype a central server is required to store the login and index information in which user names and passwords are stored. This server also ensures that Skype login names can be mapped to IP address and port number via STUN protocol. Since this paper is focus on the foundation of peer-to-peer application in wireless sensor network, we put our work mainly in the routing scheme of the context of wireless routing.

Based on routing schemes and principles[11], we classify them into two categories: address routing and name routing. All traditional routing schemes, such as Destination Sequenced Distance Vector (DSDV)[12], Ad-hoc On-demand Distance Vector Routing (AODV)[13], Hierarchical

978-1-4244-7322-9/YR/$26.00 ©2009 IEEE

State Routing (HSR)[14], Beacon Vector Routing protocol (BVR)[8], Greedy Perimeter Stateless Routing (GPSR)[7] and so on, are belonging to the address routing category. They use addresses as their unique identifiers, which are widely used in wired network, but it is not suitable in the wireless sensor network. Because, in the wireless sensor network, topologies are not static, instead locations for sensor nodes are dynamic changed. Therefore address identifiers can not be used to unique identify a sensor node.

The naming routing is developed to suit these characters in wireless sensor network[15], where each node can use a name as a unique identifier (i.e., MAC address) and distribute it among a sensor network without pre-defined address information. By using this way, name routing avoids drawbacks of address routing and can be deployed in a large sensor network where every node can identify its name by through self-learning process. Virtual Ring Routing (VRR)[16] is an originator for the name routing. The VRR uses the sensor node's natural identifier and DHT to map sensor nodes into a ring. Based on this ring, VRR performs continuous hash searching and single route routing. This routing protocol can achieve name routing and avoid drawbacks in the address routing.

In this paper, we introduced the multi-level virtual ring (MVR) structure which belongs to the name routing category. MVR improves the routing performance which VRR has suffered from the single level routing. With the backbone selection function, the connectivity is considered in calculation to determine the backbone nodes on the different levels of ring. With the DHT-style searching and mapping, the sensor nodes can be located in maximum $log(N)$ steps in a network with N nodes. With the cross-level routing mechanism, packets do not need to go through every node along the single route.

III. Multi-level Virtual Ring Framework

Some challenge exist to support peer-to-peer applications running in wireless sensor networks: addresses are not suitable to identify mobile nodes because network topologies are not static; sensor nodes have limited resources in terms of memory and power which are sensitive for the size of the routing table; and the routing performance is also a big concern.

The Multi-level Virtual Ring (MVR) structure aims at fixing these problems. First, the MVR gets rid of addresses altogether. That is, we propose to route directly on the name identifiers of mobile nodes. The name identifier system is applied to organize the different node name format into a certain size format. Second, the MVR uses the backbone selection algorithm to select key nodes and place them on the different levels of the ring. Third, with Distributed Hash Table (DHT) [3], [5] techniques, the information of the backbone nodes and peer-to-peer application data can be distributed mapped on the ring which helps to reduce the size of the routing table. Finally, in most cases, packets can be routed crossing multiple levels, which improves the routing performance by a shortcut path and saves power.

In order to implement the MVR structure and exert its advantages, we developed the following components:

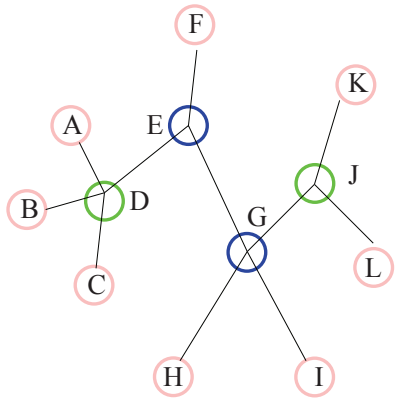

Fig. 1. A sample of sensor network

- Backbone Selection
- Naming
- Mapping
- Searching
- Routing

A. Backbone Selection

The backbone selection is used to differentiate key nodes from normal nodes and organize them as the backbone nodes for each virtual level, according a certain selection criteria. The rest of normal nodes will be treated as children or leaves belonging to backbone nodes.

The backbone selection algorithm is a self-running algorithm, it does not need to set up any infrastructure. Instead, each node runs this algorithm and gets its virtual level information. For a graph $G = (V, E)$, a vertex $v \in V$ represents a node, and an edge (u, v) indicates that two nodes u and v are within their transmission range. A subset of the vertex, $V_b \subseteq V$, is a backbone set if each vertex in $V - V_b$ has at least one neighbor in V_b. A virtual backbone ring is constructed by virtually connecting the backbone nodes. We develop this algorithm for constructing the virtual backbone ring.

The first step of backbone selection algorithm for each sensor node is to send hello message (HelloMessage) to its one hop physical neighbors, where the HelloMessage includes the source node' MAC address. When receiver receives the HelloMessage, it will get the sender's MAC address from the packet header and store this MAC address into its own physical neighbor set. After a certain time, every node gets all its physical neighbors' MAC addresses and stores them into its own physical neighbor set.

The second step of backbone selection algorithm is to send the backbone selection message. The backbone selection message includes the current physical neighbor set it has. When the node receives backbone selection messages, it will get the source node's physical neighbor set from these message and compare them with its own physical neighbor set. If the receiver's physical neighbor set includes the sender's physical neighbor set, that means the receiver is the key node. Then the receiver sends a get rid of message to the sender and assigns the selection round number as its virtual level information, in this case the round number is 1, so the virtual level information

Algorithm 1 Backbone Selection Algorithm

```
int VirtualLevel=1, MyLevel=0;
Receive backbone selection message with the neighbor_set parameter
while MyLevel==0 do
    if my_neighbor_set includes the neighbor_set then
        Inform source sensor node to get rid of itself and VirtualLevel++
    else
        MyLevel = VirtualLevel
    end if
end while
```

for the receiver is 1. After that, the receiver will enter into the next backbone selection round. If the sender's physical neighbor set includes receiver's physical neighbor set, then the receiver will get rid of itself and inform the sender to enter into the next round. When the receiver get rid of itself, it will assign this round number as its current virtual level. If neither the sender or receiver is included each other, then this backbone selection message will be dropped. And both node will enter into the next round. In the second round, nodes will update their backbone selection set, which originally is the same as the physical neighbor set. And they will send and receive backbone selection messages as the first round, except there is no children at all. After several rounds, nodes will be get rid of by themselves or by other nodes. Eventually, there should only left one or zero node in this sensor network. If there is zero node, the backbone selection process will be stopped. If there is one node left, this node will assign the virtual level as that round's number, and the backbone selection process will be stopped. After this backbone selection process, every node will get a virtual level and this virtual level will be used in the naming and mapping process.

For example, in the sensor network shown in Figure 1. In the first round the nodes in red color are grouped into the "leaves" nodes because they are "children" of the other nodes (i.e., green nodes and blue nodes). These nodes (in red color) are removed from the backbone selection. The other nodes (in green color and blue color) form the Level 1 backbone ring. In the second round, the nodes in green color will be removed because they are the "children" of the other nodes (in blue color). So, the nodes (in blue color) form the Level 2 backbone ring. In this case node A,B and C are the children of D; node H and I are children of node G; node K and L are children of J and node F is the child of E. The key node G,J,D and E form the bottom virtual ring; node G and E form the second virtual ring and map to the bottom virtual ring at the same time, as shown in Figure 2.

B. Naming

The naming process is to give node a unique identifier. Based on this process, nodes can be assigned unique names with their embedded identifiers in an organized sensor network. The most important thing is that this naming process is a self-learning process. So, we can avoid to deploy any

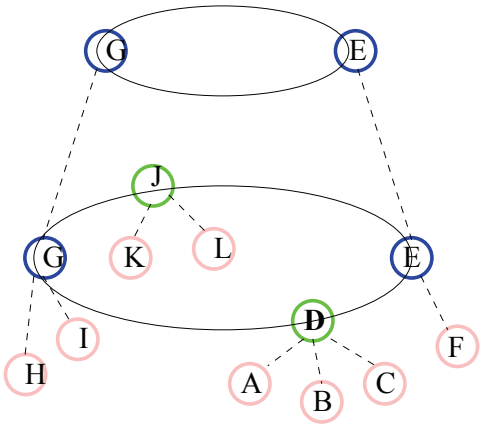

Fig. 2. After the backbone selection process, the position of each node

TABLE I
THE DISTRIBUTED ROUTING

At node	*Other node 1*	*Other node 2*	*Other node 3*
Node(0)	(1)	(2)	(4)
Node(1)	(2)	(3)	(5)
Node(2)	(3)	(4)	(6)
Node(3)	(4)	(5)	(7)
Node(4)	(5)	(6)	(0)
Node(5)	(6)	(7)	(1)
Node(6)	(7)	(0)	(2)
Node(7)	(0)	(1)	(3)

infrastructure, which is the one of the goals for pure peer-to-peer application in wireless sensor network.

The naming process in MVR uses hash function, such as SHA-1 [17], to hash nodes' identifiers. In MVR, we use the embedded hash function, $NodeHash(MAC, Level)$, to generate the nodes' identifiers. In the $NodeHash(MAC, Level)$ function, it has two entries, one is the node's name identifier, MAC address in this case, another is the node's virtual level information which is obtained from the backbone selection step. By using this way, we can give each sensor node a unique identifier which contains the node's MAC address and the virtual level information associated with the node. Since the size of the result from SHA-1 is 128 bits, it can avoid two entries have the same hash value in most cases. After the hashing process, nodes have their own unique hash values as their identifiers in the sensor network.

C. Mapping

The mapping process in MVR applies the DHT techniques [3], [6] to distribute the routing information on the ring. Thus a node resolves the hash function by communicating with a few other nodes. DHT uses a variant of ordered hash values, which is similar to consistent hashing [4], to assign those unique identifiers to sensor nodes, where each node receives roughly the same number of name identifiers($O(log(N))$). On the bottom level of MVR, it becomes a DHT ring, in which all identifiers are mapped into the ring.

After the mapping process, a smart and routable sensor network has been created. Each node maintains a node index table to store the relevant routing information. For example, there

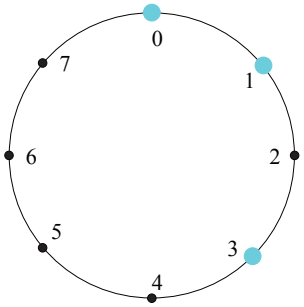

Fig. 3. Key location for a network with nodes 0, 1 and 3

TABLE II
DEFINITION OF VARIABLES FOR NODE N, USING M-BITS IDENTIFIERS

Notation	*Definition*
finger[k].start	(n + 2^{k+1}) mod 2^m, $1 \leq k \leq m$ mod
.intrval	finger[k].start, finger[k+1].start
.node	first node $\geq$ n.finger[k].start
successor	the next node on the identifier circle
predecessor	the previous node on the identifier circle

are 8 nodes in Figure 3. The distributed routing information is shown in Table I in which each node just stores 3 other nodes information ($log(8) = 3$). The i^{th} other node at the node n is calculated as $n+2^{i-1}$. The interval between the i^{th} other node and $(i+1)^{th}$ other node is 2^{i-1}. For example, at node 0, the first other node should be stored is node $0+2^{1-1}$ which is node 1. The second other node at node 0 is node $0+2^{2-1}$ which is node 2. The third other node at node 0 is node $0+2^{3-1}$ which is node 4.

D. Searching

The searching process in MVR is inspired idea from the Chord [3], [6]. A node in Chord needs only a small amount of "routing" information about other nodes' identifiers. MVR borrows this idea, but the purpose of searching in MVR is different. The searching purpose in MVR is to identify the destination node's identifier or level information through searching the node identifier.

According to the Chord, let m be the number of bits in the key node identifiers. Each node, n, maintains a routing table with (at most) m entries, called a finger table. The i^th entry in the table at node n contains the identity of the first node, s, that succeeds n by at least 2^{i-1} on the identifier circle, i.e., $s = successor(n + 2^{i-1})$, where $1 \leq i \leq m$ (and all arithmetic is modulo 2^m). We call node s the i^th finger of node n, and denote it by n.finger[i].node (see Table II).

For example, consider the Figure 3. Suppose node 3 wants to find the successor of identifier 1. Since 1 belongs to the circular interval [7,3), it belongs to 3.finger[3].interval; node 3 therefore checks the third entry in its finger table, which is 0. Because 0 precedes 1, node 3 will ask node 0 to find the successor of 1. In turn, node 0 will infer from its finger table that 1's successor is the node 1 itself, and return node 1 to node 3.

E. Routing

The routing process in MVR inherits the advantages of multi-level virtual ring, where the cross-level routing can be deployed. With cross-level virtual ring, packets can be forwarded efficiently to the destination without flooding. After the searching process, the source node obtains the destination level. Then the sensor node looks up the cross-level interface to decide which physical neighbor is the most suitable node as the next hop.

Every node in MVR maintains a routing table, where the virtual level interface and the physical neighbor set are stored. When a node receives a packet, it will analyze the packet header to find the destination level and compare with its own level by using the level similarity decision algorithm (Algorithm 2) to calculate its next hop, which towards the destination level. When a node, located at the destination level, receives this packet and finds the destination level from the packet header is the same as its own level. Then this node will forward the packet according the short path using the best virtual neighbor, which is similar to the idea of VRR [16]. .

Algorithm 2 Level Similarity Decision Algorithm

1: Receive forward packet with dest_level and dest_identifier parameters
2: **if** The dest_level > my_level **then**
3: Source node looks up its upper interface and compares with its physical neighbor set to find which one should be selected as next hop
4: **else if** The dest_level < my_level **then**
5: Source node looks up its down interface and compares with its physical neighbor set to find which one should be selected as next hop
6: **else**
7: Packet has been arrived on the destination level and should forward this packet to its next hop according its virtual neighbor set
8: **end if**

The idea behind MVR routing forward is to take advantages of the cross-level forwarding. By deploying this mechanism, packets can bypass some "useless" routing nodes and be forwarded directly.

F. Scalability

One of the big characters of peer-to-peer in wireless sensor network is to have a good scalability[18]. This require MVR should be a network structure where every node can join in and leave easily without affecting the sensor network's structure and other nodes' connection. MVR uses two main methods to make sure new nodes can join into the original sensor network. One way is just to link a new node to the bottom level as a child. The other one is that after a certain time, the sensor network will repeat the backbone selection process and create a new multi-level virtual ring to suit the new topology.

When a new node join into a sensor network, that means this node is already linked to another node and located in this sensor network. Since, this new node can be treated as

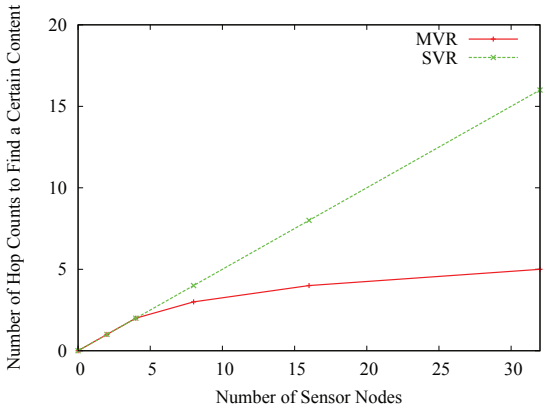

Fig. 4. Hop Counts to Find a Certain Node

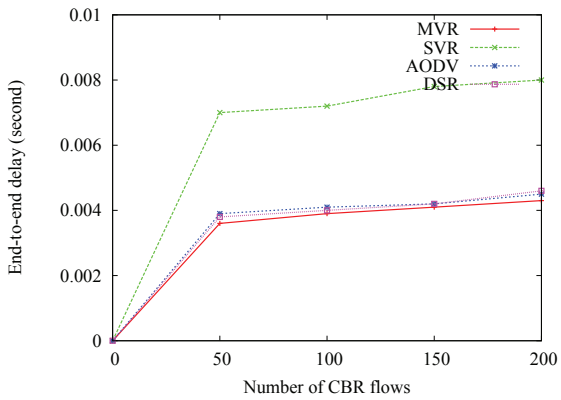

Fig. 5. End-to-end Delay

another node's child, it can be added to the existing network as a "child" and linked to the bottom level with its "parent". After that, this new node will use the $MVRHash()$ to get a unique identifier from the MVR. Then the node sends its MVR identifier to the bottom level through its parent and then the bottom level will add this new identifier into the DHT. After these two steps, this new node can be added as a normal node in this sensor network.

After a certain time, which depends on the size of the network, the sensor network should be updated. The process of updating is the same as forming a new MVR sensor network. It needs backbone selection, naming and mapping. After this two steps, new nodes can be located at the right place, which can improve the routing efficiency.

We use notation employed in asymptotic analysis to describe various scalability characteristics. Thus we use the $\Omega(X)$ to denote a lower asymptotic bound, $O(X)$ to denote an upper asymptotic bound, and $\Theta(X)$ to denote a simultaneous upper and lower asymptotic bound. By asymptotic bound, we refer to the scaling behavior of the protocol with respect to a given variable.

In MVR, the backbone nodes (e.g., N nodes) are formed to level-1 of the ring. Then, up to $N-2$ nodes at maximam are further formed to level-2 of the ring, etc. In general, with a level degree of d, the size of the routing table will be on the order of $O(d * logN)$.

G. Simulations

In order to measure the design goals of name routing MVR, we evaluate our ideas on *ns-2*. The simulations compared the performance of MVR, DSR[19], AODV[13] and Single Virtual Ring (SVR), which is similar to the VRR[16]. Since it is a new peer-to-peer name routing structure in wireless and sensor network, we mainly focus on the reliability[20] and routing performance simulation results in this paper.

1) Evaluation Metrics: We measured the fraction of searching a target node, the end-to-end delay for these packets and the number of processing messages per correct delivery of MVR. Each experiment run three times and we present the average result for each metric. We used the same traffic and topology patterns for all protocols.

2) Performance of searching a target node: One of purposes of the MVR structure is to quickly find and locate the target node. We designed an experiment to test how many hops are needed to find a target node in the MVR structure. We compared the performance of the MVR searching with the SVR searching. The searching performance has been tested in the different sensor networks with different power, mobile speed and different number of sensor nodes, range from 5 to 32 respectively. The results are shown in the Figure 4.

Based on the results, we see the searching performance of MVR is roughly constant to the size of the network. The reason is that during a certain time the MVR structure is steady and it uses DHT technique for the searching process which needs only $O(logN)$ hops, where the N stands for the number of nodes in a sensor network. With the cross-level routing, MVR performs better than the Single-level Virtual Ring (SVR) structure because the cross-level routing can take the shortcut route.

3) Performance of end-to-end delay: The end-to-end delay is a important metric in sensor network. If the end-to-end delay is high, this routing protocol is unacceptable. So, we design an experiment to test the performance of end-to-end delay. This experiment test the traffic messages from 50 to 200.

From the Figure 5, we can find the end-to-end delay for MVR, AODV and DSR are similar but for SVR is higher than previous three routing protocols. This is due that SVR map all nodes' name identifiers to the single ring and perform the routing scheme on the single ring. While MVR uses cross-level forward, where packets always go up before they get to the destination node's level. By using this way, packets can bypass lots of 'useless' routing nodes and always keep a right route. This idea is similar to the address routing but we implement it on the name routing structure.

Based on this experiment, we can find the end-to-end delay of MVR is similar to the address routing protocols and much better than SVR.

4) Performance of message overhead: When analyze the efficient of routing protocol, the message overhead is a key metric to be tested. We design a experiment to test how much packet needed to process for one delivery. The number of traffic message is the same as before, we start from 50 to 200.

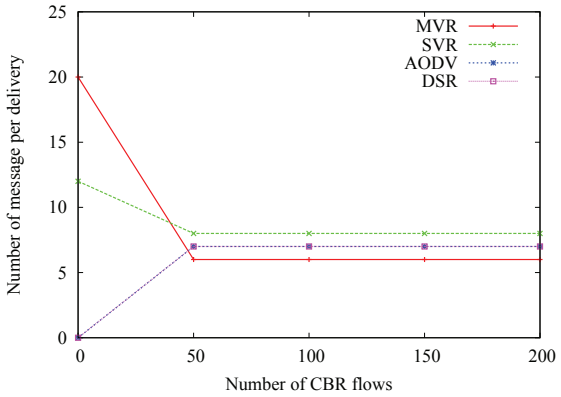

Fig. 6. Message Overhead

From the Figure 6, we can find the AODV and DSR is similar, which both start from 0. The MVR and VRR needs a high exchange rate of message at the beginning. After the name routing structure is set up, MVR can use the least packets exchange to finish a delivery. However, for SVR, it needs much more exchanges compared with other three routing protocols. This is because, the name routing needs to set up the virtual structure, exchange name identifier and so on. Since the MVR needs to run the backbone selection algorithm, which needs large selection message exchanges, it has a high message overhead at start time. After the message exchange process, MVR has the least message overhead, because it uses the cross-level routing forward and it has already run the backbone selection algorithm.

IV. Future Work

MVR is a new peer-to-peer name routing foundation structure in sensor network. It overcomes drawbacks of address routing and improves the VRR routing scheme. With the natural advantages of MVR, data can be easily distributed among multi-level and implement the grid computing or cloud computing. When it needs to process complex computing, nodes in MVR can be organized as a single processing point. Complex data or function can be divided into multiple small data units and each unit will be distributed to some node, which depends on that node's capability. MVR structure is natural suitable for data distributed, load balance and if there has a good mechanism, MVR can save more energy compared with other network structures.

V. Conclusion

We present a new multi-level virtual ring name routing structure to support peer-to-peer application in wireless sensor network. MVR avoids drawbacks of address routing and improve the name routing performance. With the backbone selection, naming and mapping, a MVR structure sensor network can be set up. With exponential searching algorithm and cross-level routing forward, MVR has a good routing decision and forwarding performance. The backbone selection algorithm is developed to suit the natural characters among nodes. We also make use of DHT to enhance the mapping and searching performance.

With the simulation results, we can prove MVR can be implemented and as the foundation of peer-to-peer application in wireless sensor network. MVR also have natural advantages to be extend to other fields compared with other address or name routing protocols.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] A. Harwood and E. Tanin, "Hashing spatial content over peer-to-peer networks," in *In Australian Telecommunications, Networks, and Applications Conference-ATNAC*, 2003, pp. 1–5.

[3] I. Stoica, R. Morris, D. Karger, F. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 149–160, October 2001. [Online]. Available: http://dx.doi.org/10.1145/964723.383071

[4] A. Haeberlen, J. Hoye, A. Mislove, and P. Druschel, "Consistent key mapping in structured overlays," in *In Technical Report TR05-456, Rice CS Department*, 2005.

[5] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, November 2001, pp. 329–350.

[6] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in p2p systems," *Commun. ACM*, vol. 46, no. 2, pp. 43–48, 2003.

[7] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM Press, 2000, pp. 243–254. [Online]. Available: http://dx.doi.org/10.1145/345910.345953

[8] R. Fonseca, S. Ratnasamy, J. Zhao, C. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon vector routing: Scalable point-to-point routing in wireless sensornets," in *In NSDI*, 2005.

[9] S. Mccanne, S. Floyd, and K. Fall, "ns2 (network simulator 2)," http://www-nrg.ee.lbl.gov/ns/. [Online]. Available: http://www-nrg.ee.lbl.gov/ns

[10] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," September 2004.

[11] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 11, no. 6, pp. 6–28, 2004. [Online]. Available: http://dx.doi.org/10.1109/MWC.2004.1368893

[12] C.Perkins and P.Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computer," *ACM Sigcomm'94*, August 1994.

[13] C.Perkins and E.Royer, "Ad hoc on-demand distance vector routing," *Mobile Computing System and Applications*, February 1999.

[14] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable routing strategies for ad-hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1369–1379, 1999.

[15] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," 2001.

[16] M. C. Miguel, "Virtual ring routing: Network routing inspired by dhts." [Online]. Available: http://citeseer.ist.psu.edu/caesar06virtual.html

[17] r. D. Eastlake and P. Jones, "Us secure hash algorithm 1 (sha1)," Published Online, Internet Engineering Task Force, RFC 3174, September 2001. [Online]. Available: http://www.faqs.org/rfcs/rfc3174.html

[18] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems.* New York, NY, USA: ACM Press, 2003, pp. 14–27. [Online]. Available: http://dx.doi.org/http://doi.acm.org/10.1145/958491.958494

[19] D. B. Johnson, D. A. Maltz, and J. Broch, "Dsr: the dynamic source routing protocol for multihop wireless ad hoc networks," pp. 139–172, 2001.

[20] I. Stojmenovi and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Wireless Networks*, 1999, pp. 48–55.