Reactive ID Assignment for Sensor Networks

Hongbo Zhou
Dept. of Computer Science
Slippery Rock University
Slippery Rock, US
hongbo.zhou@sru.edu

Matt W. Mutka
Dept. of Computer Science & Engineering
Michigan State University
East Lansing, US
mutka@cse.msu.edu

Lionel M. Ni
Dept. of Computer Science
Hong Kong University of Science & Technology
Hong Kong SAR, China
ni@cs.ust.hk

Abstract— **Globally unique ID allocation is usually not applicable in a sensor network due to the massive production of cheap sensor nodes, the limited bandwidth, and the size of the payload. However, locally unique IDs are still necessary for nodes to implement unicast communications to save power consumption. Several solutions have been proposed for locally unique ID assignment in sensor networks. However, they bring much communication overhead, which is not desirable due to the limited power supply in a sensor node. Combined with a directed diffusion communication paradigm, a reactive ID assignment scheme with security mechanisms is proposed in this paper. It defers ID conflict resolution until data communications are initiated and thus saves communication overhead.**

Keywords-ID assignment; sensor networks

I. INTRODUCTION

A sensor network consists of a large number of sensor nodes engaged in environment monitoring and wireless communications, simultaneously. Due to the low cost of a sensor node and the convenience of deployment, a sensor network will find applications in many areas, such as battlefield surveillance, precision agriculture, smart transportation, and wildlife study [1].

Although a sensor network is similar to a mobile ad-hoc network (MANET) since both are multi-hop wireless networks, they are different in their architectures and data communication schemes.

A MANET is usually an IP-based network. Any node in the network can initiate communications with any other node. Thus, every node must be assigned an IP address that is globally unique. To initiate data communications, the IP address of the destination must be identified and the path to it must be built.

In a sensor network, communication is data-centric instead of address-centric, which means the user is interested in the location and the data collected by the sensor node, but does not care about the address of the sensor node. Moreover, since the payload length in the data packet is usually small, it wastes bandwidth and power if the data are encapsulated in a TCP/UDP/IP packet. Thus, customized network protocols are adopted instead of TCP/IP to save the communication overhead and energy consumption.

Without the IP routing function provided by the IP layer, a node in the sensor network still needs an approach to find the path toward the destination. The directed diffusion paradigm [2] was proposed to replace IP routing for data communications in sensor networks. According to the scheme, a sink node broadcasts an interest message that is flooded throughout the network. Every node records the upstream node as the next hop back towards the sink. After the path is built, the reply is sent back from the source along the reverse path to the sink. The sink node and source nodes can "reinforce" the path, so subsequent queries can be unicast packets.

One assumption in directed diffusion is that every node has a locally unique ID. Moreover, locally unique IDs will save communication overhead, as illustrated in Fig. 1.

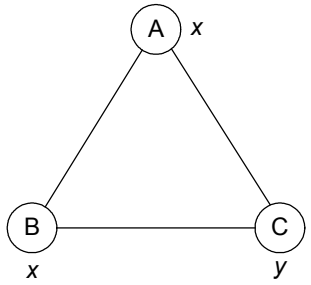

Figure 1. ID conflicts in a sensor network

In Fig. 1, nodes A, B, and C are connected to each other. Nodes A and B have the same ID of *x*, node C has a different ID of *y*. If node A wants to send a packet to node B, because the destination has the same address of the source, a traditional network layer protocol usually considers packet destined for itself and will not deliver the packet to the underlying data link layer[1]. If node C wants to send a packet to either node A or node B, because they both have the same address, both will receive the packet and process it, which will waste power. In contrast, if they have different addresses, one of them will ignore the packet at the data link layer once the destination address is identified, and thus power is saved.

Although there have been many autoconfiguration algorithms proposed for ID assignment in MANETs ([3] – [11]), they aim at assignment of globally unique IP addresses to nodes in the network, and are not appropriate for ID assignment in a sensor network due to the following reasons:

This research is supported in part by NSF grant no. 0334035, Hong Kong RGC grants HKUST6264/04E, and AoE/E-01/99.

[1] Although it is possible to customize the network layer protocol to send the packet with the destination address equal to its own, it is contrary to most traditional network layer protocols that intend to provide an IPC mechanism.

(1) The number of nodes in a sensor network is much larger than that in a MANET, so the probability of duplicate addresses may be very high with a limited number of address bits. Thus, high communication overhead will occur to resolve conflicts if the scheme wants to achieve global uniqueness;

(2) After deployment, the nodes in a sensor network start up simultaneously, which means that most of them join the network and require IDs almost at the same time. All the autoconfiguration schemes for MANETs assume that the arrival rate of new nodes is medium to low and that the new node obtains its address from configured nodes;

(3) Since an IP address is not used in the sensor network, it is meaningless to achieve global uniqueness of IDs of the sensor nodes. For the directed diffusion communication paradigm, locally unique IDs will suffice;

(4) The size for the address field should be smaller in a sensor network compared to a MANET because the payload length is very small in the former. Otherwise, data communications will be less efficient.

Although all the sensor nodes may be equipped with a locating device such as GPS, it is not adequate to simply use the location of a sensor node or the hash value of the location information as its ID because:

(1) The number of bits required to represent the location information in the address field may be large, and thus considerable power will be wasted;

(2) In a sensor network with high node density, the nodes that are close to each other may have the same location information due to the low resolution of the locating device;

(3) For the hash value of the location information, without comparison, it is still unknown if the hash values of adjacent nodes are different or not.

This paper proposes *reactive ID assignment*, which is an efficient ID assignment in a sensor network. The paper is constructed as follows: Section 2 introduces work related to ID assignment in sensor networks. Reactive ID assignment is described in Section 3. It defers ID conflict resolution until data communications are initiated to save communication overhead and power consumption. The efficiency of the scheme is supported by simulation results in Section 4. Section 5 concludes the paper.

II. Related Work

The scheme proposed in [12] utilized a proactive conflict detection method for a general sensor network, including a mobile sensor network, and a stationary sensor network with new members joining. When a node boots up, it first chooses a random physical address and then announces it with periodic broadcasts of HELLO messages with the interval of 10 seconds. All the nodes record the source address of the HELLO message in a neighbor table, which is included in the subsequent HELLO messages. Therefore, every node will have 2-hop neighbor information, which is utilized to resolve address conflicts among 2-hop neighbors. If a node finds that one of its neighbors chooses a duplicate address, it will notify this neighbor to change the address. To further decrease the average address field length, the scheme encodes the physical address using Huffman coding.

In [13], the scheme is modified to specifically suit stationary sensor networks in which periodical broadcasts of HELLO messages are replaced by a fixed number of broadcasts, with 4 cycles and the interval of 8 seconds recommended. The address is encoded as a Huffman code as well.

To implement Huffman coding, the user of the sensor network must first run a simulation with the expectant node density to compute the Huffman code table. The code table is then input into every node for coding and decoding of the physical address in the packet header for data communications. However, according to the simulation results, only 0.3 to 0.8 bit is saved for one physical address on average with Huffman coding compared with fixed-length format. In the case where the actual density of nodes is not the same as expected, the authors admitted that no benefit in codeword length would be achieved.

III. Reactive ID Assignment

In this section, the assumptions for our reactive ID assignment scheme are given first, and then the procedures are described in detail with an example.

A. Assumptions

Although the schemes in [12] [13] recommended Huffman coded addresses for sensor nodes, we still adhere to fixed-length IDs due to the following reasons:

(1) The nodes in a sensor network are usually manufactured in batches. Although the average length of Huffman-coded address is less than the size of a fixed address format, it would be much easier for designers to allocate the fixed-length field for the MAC address in the physical layer in advance;

(2) The apriori Huffman code table may not be optimal for the nodes in a sensor network, and cannot be calculated in some cases (e.g., the sensor nodes may be dropped from airplanes or missiles);

(3) The optimal Huffman coding can save only 0.3 to 0.8 bit for one address, and it will become non-optimal as some sensor nodes die with time and new nodes join the network; and

(4) The Huffman code table must be stored in the memory of the sensor node during all its lifetime.

We define 1-hop uniqueness as address uniqueness among direct neighbors, and 2-hop uniqueness as address uniqueness among 2-hop neighbors. The assumption for 1-hop uniqueness is that the number of nodes in the largest complete sub-graph in the sensor network should be less than the range of the addresses (or the range of addresses minus 1, if a special address is designated as the broadcast address). The assumption for 2-hop uniqueness is that the maximum sum of the number of 1-hop neighbors and the number of 2-hop neighbors should be less than the range of the address.

If the nodes in a sensor network are deployed too densely, the assumptions may be violated. However, if the power of the sending packets can be adjusted, which is a desirable attribute for sensor nodes, the sending nodes can lower the power to satisfy the assumption.

We also assume that during the flooding of an interest message in the directed diffusion paradigm, every node broadcasts only once, which is the basic optimization in flooding [14].

B. Scheme

In the schemes proposed in [12][13], the nodes resolve ID conflicts proactively after they boot up, so the address of a node is ready for data communications. However, in a sensor network, due to the limited power supply, it is a waste of power to establish addresses until data communications begin. We need a new method to resolve conflicts and simultaneously preserve as much power as possible. The solution is to delay ID conflict resolution until data communications are necessary.

We can use an example to illustrate how to combine directed diffusion with conflict resolution in reactive ID assignment. For the small network in Fig. 2, suppose that node A is the sink, node B is the source. Nodes A and B choose the same random address of a_1, nodes C and D choose the same random address of a_2. There are duplicate addresses among direct neighbors A and B, and among 2-hop neighbors C and D.

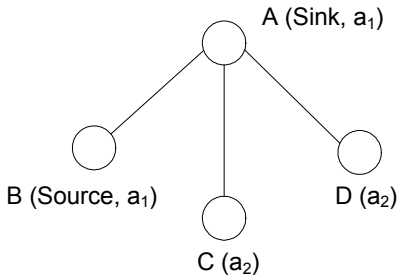

Figure 2. A small sensor network

Because all the addresses are randomly chosen and no communication has occurred yet, a node is not sure if its address is 1-hop unique, so it can use the address only temporarily. When the sink node broadcasts an interest message, the node can eliminate duplicate addresses among its direct neighbors because the receiver can choose another address randomly if it receives a packet with the same address. In the example in Fig. 2, once the sink node A broadcasts the interest message, node B must change its address (to a_3, for example). This applies to other nodes when the interest packet is forwarded.

We can also utilize forwarding to eliminate duplicate addresses among 2-hop neighbors according to the assumption that every node forwards the interest message only once. After nodes C and D forward the message, node A will receive the same message with the same source address twice. Thus, node A will be aware that there are two direct neighbors with the same address. Node A can unicast a special control message (RESOLVE message) to notify them that there exists a 2-hop conflict. On receipt of the RESOLVE message from node A, both node C and node D need to change to another random address.

If nodes C and D unfortunately choose the same address again, or choose the new address of node B (a_3), there will still be a 2-hop conflict[2]. To prevent the conflict, we require that nodes C and D broadcast an announcement of their changes (CHANGE message), which can be collected and checked by their neighbors.

Because the locally unique ID is used in the construction of the path between the sink and source, a mechanism is necessary to identify the origin of the change message. As illustrated in Fig. 3, if node B records node A (with the address of *x*) as its next hop back to the sink and then it finds that there are two 1-hop neighbors with the same address of *x*, it notifies them to change. Node A may change to the address of *y*, node C to *z*. On receipt of both change messages, which new address should be used as the next hop for node B?

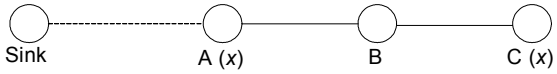

Figure 3. An example of 2-hop conflict

There are two methods to solve this problem:

(1) We can designate that every node choose a random number in addition to its random address. If the length for the random number is long enough (e.g., 16 bits), the probability that two neighboring nodes choose the same random number will be very low. If the random number is piggybacked in the broadcast of the interest message and change message, a node is differentiated among its neighbors. For a stationary sensor network, the random number is needed only once for the first broadcast/ forwarding of the interest message. In case that new nodes may join the stationary network later or a node misses copies of the first interest message, a node will receive a message without the piggybacked random number as the first message. The node can just drop the message and broadcast a special control message requesting its neighbors' random number for once. Thus, the overhead caused by the random number in the interest message is trivial. However, it will bring too much communication overhead in a mobile sensor network since every interest message must include the random number.

(2) The alternative for mobile sensor networks is to use a hop count field that is usually found in routing messages in a MANET. As the interest message passes a node, the hop count field is increased by 1, which is also recorded in the node. The hop count field is also included in the change announcement message. Therefore, if the hop count in the change message is equal to the receiver's hop count minus 1, then the next hop address is updated if it is the same as the old address contained in the change message. We can limit the size for the hop count field to only 4 bits, and utilize the modulo operation. For example, in Fig. 3, if the hop count is 15 for node A, 0 for node

[2] They will not change to node A's address since they are aware that the address of a_1 is already occupied.

B, and 1 for node C, on receipt of both change messages, node B will update its next hop address to *y* because 0 = (15 + 1) mod 16. If the hop counts for both nodes A and C are 2, and it is 3 for node B, either one could be the next hop for node B because neither of them uses node B as the next hop. With the hop count field, the neighbors can be differentiated, and the possibility of the path loop is minimized.

The reactive ID assignment scheme can be applied to mobile sensor networks as well. As the sensor nodes move around, there will still be local ID conflicts after previous conflict resolution. However, as long as there is no data communication, the ID conflict brings no harm to the sensor network, and it will be resolved during the next data communication.

The differences between the reactive ID assignment and the proactive method are:

(1) The ID conflict resolution is postponed until data communication is initiated in the reactive scheme; and

(2) The 1-hop neighbor table is not included in any control messages in the reactive scheme, which achieves shorter message length and less power consumption;

The performance of the two schemes is compared in Section 4.

C. Procedures

In summary, the procedure works as follows:

(1) In the beginning, every node chooses a random ID;

(2) The sink node broadcasts an INTEREST message;

(3) All the neighbor nodes record the sender's ID. If the sender's ID is the same as its own, it chooses another one randomly, and broadcasts a CHANGE message (this solves 1-hop conflict);

(4) The neighbor waits for a random delay and rebroadcasts the INTEREST message;

(5) If a node receives an INTEREST message with the same source ID more than once, it puts the ID in a RESOLVE message and broadcasts to its neighbors (this solves 2-hop conflict)[3];

(6) If a node receives a RESOLVE message containing its ID, it chooses another one randomly (because it records all the 1-hop neighbors' IDs, so it will not lead to 1-hop conflict), and broadcasts a CHANGE message (to avoid further potential 2-hop conflict);

(7) After the intended source node receives the INTEREST message, it unicasts a REPLY message back to the sink (every node records the sender's ID of the first copy of the INTEREST message as the next hop back to the sink);

(8) On receipt of a CHANGE message, a node updates its next hop back to the sink, if necessary.

[3] To be more exact, it should be a unicast message received by more than one recipient. However, during the simulation, we use broadcast instead.

IV. SIMULATION

The simulations for both our reactive ID assignment scheme and the schemes proposed in [13] are implemented to compare their performance in *ns*–2 (version 2.27) with CMU extension for ad hoc networks [15]. The sensor nodes are placed in a grid for a stationary sensor network.

A. Simulation verification

A simple simulation scenario is run to verify the correctness of the implementation, which has 15 nodes in a 5 × 3 grid, as illustrated in Fig. 4. The numbers shown in the figure are the unique IDs (UID) of the nodes, which are used for analysis only and should not appear in reality. The distance between two nodes is 200 meters so that a node in the middle of the network has 4 direct neighbors. The size for the address is 4 bits.

In the beginning, all the nodes choose a random ID, which is placed in the parentheses following its UID, as in Fig. 5. After initialization, there are two cases of 1-hop conflicts (between nodes 3 and 8, nodes 4 and 9), and one case of 2-hop conflict (between nodes 6 and 10).

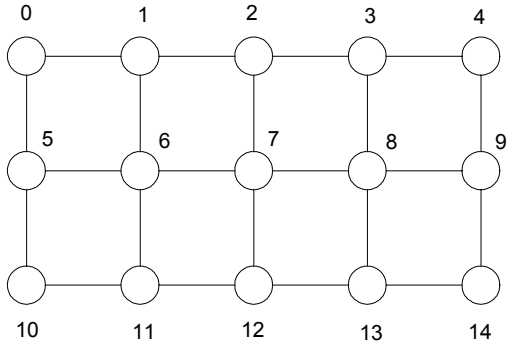

Figure 4. A sensor network in 5 × 3 grid

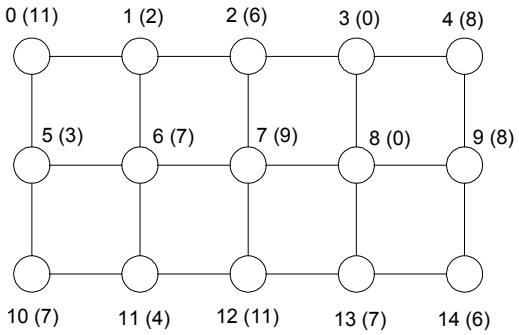

Figure 5. Every node chooses a random ID

Once node 0 broadcasts an INTEREST message for destination node 9, all the nodes forward the message. After node 3 receives the INTEREST message from node 8, it knows there is a 1-hop conflict. Node 3 first changes its ID to 6 randomly, which conflicts with node 2's ID. Because node 3 has recorded node 2's ID, it then changes its ID to 13 randomly, which is appropriate. Similarly, node 8 changes its ID from 0 to

2^4. After node 4 broadcasts, node 9 changes its ID from 8 to 0, then to 1. Node 3 receives the INTEREST message from a node with ID of 7 twice, so it can conclude that there is a case of 2-hop ID conflict. It then broadcasts a RESOLVE message to its neighbors. On receipt of the RESOLVE message, both nodes 6 and 10 change their IDs. All these changes result in the IDs illustrated in Fig. 6.

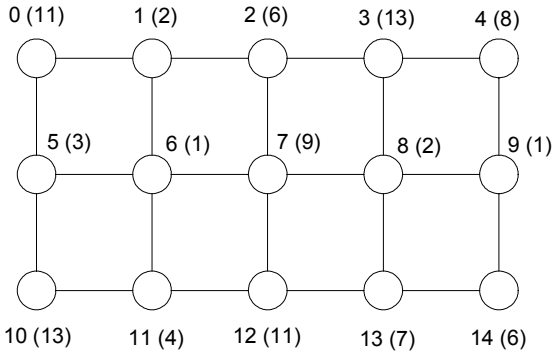

Figure 6. Every node has a locally unique ID

Fig. 7 shows the number of packets received at each node for two broadcasts of INTEREST messages[5]. The payload packets include both INTEREST messages and REPLY messages. The overhead packets include both CHANGE and RESOLVE messages. Compared with the number of payload packets, the number of overhead packets is small. Because 1-hop and 2-hop conflicts are resolved during the first broadcast, and all the nodes are stationary, there is no increase on overhead during the second broadcast.

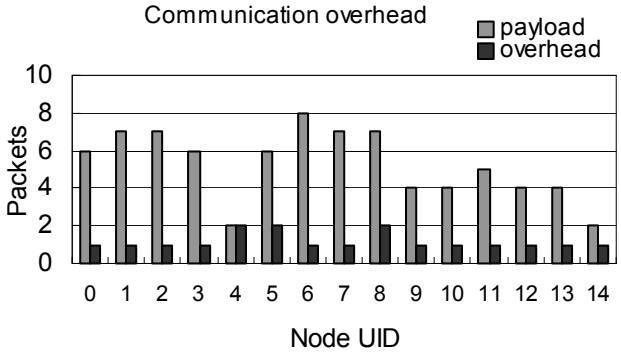

Figure 7. The number of packets received at each node for 2 broadcasts

As the address range increases, the communication overhead decreases. When the size for the address field is set to 5 bits in the simulation, there is no communication overhead because the randomly chosen IDs satisfy 2-hop uniqueness.

[4] Because nodes 3 and 8 broadcast almost simultaneously, they both change their IDs. If there is an interval between their broadcasts, only one needs to change.

[5] We use the number of received packet as the metric for communication overhead because each received packet consumes receiver's power.

B. *Simulation of large-scale stationary sensor networks with high node density*

The simulations of large-scale stationary sensor networks with high node density were done for 100 (10×10), 225 (15×15), and 324 (18×18) nodes deployed in a grid. The length for the address field is 8 bits. The transmission range is 250 meters, while the nodes are placed in the interval of 80 meters in the grid. Thus, one node has a minimum number of 10 1-hop neighbors and a maximum number of 28 1-hop neighbors, which is confirmed by the simulation results.

In the proactive scheme, every node begins to broadcasts periodic HELLO messages containing its neighbor table. The interval of HELLO messages is 8 seconds, according to [13]. 2 seconds after the last broadcast, the neighbor table that is similar to Table 1 is printed out for analysis. In the reactive scheme, each node broadcasts a HELLO message in the end of the simulation to build the neighbor table for analysis. A Perl script is utilized to locate 1-hop and 2-hop conflicts in the neighbor table.

Fig. 8 shows the sum of received control packets at all the nodes for each simulation. Notice that the 4 cycles recommended for the proactive scheme are not adequate to eliminate 1-hop or 2-hop conflicts with high node density for 225 nodes and 324 nodes. The simulations show that 13 cycles are minimum for 225 nodes, and that 15 cycles are minimum for 324 nodes. The reason is that although each node waits for a random delay before broadcasting HELLO messages, the HELLO messages are still lost at some neighbors due to high node density. According the simulation results, the number of control packets received at each node is far fewer for the reactive scheme than proactive schemes. Thus, both bandwidth and power are saved in the reactive scheme. Furthermore, no neighbor information is carried in the control messages, which leads to shorter length and less power consumption for each individual control message. In summary, longer lifetime can be achieved with the reactive scheme.

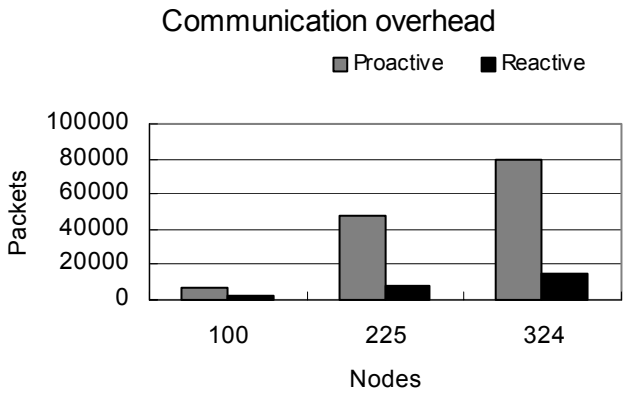

Figure 8. Number of received control packets

V. CONCLUSION

Due to the differences between a MANET and a sensor network, the pre-existing autoconfiguration algorithms for the

former cannot be simply applied to the latter. However, a mechanism is still necessary to assign locally unique addresses to sensor nodes efficiently. Compared with proactive schemes, a reactive ID assignment approach is proposed to accomplish the goal and preserve more power by means of delaying ID conflict resolution until necessary. It has no requirement on apriori unique IDs of the sensor nodes, and is easy to integrate with the directed diffusion communication paradigm.

REFERENCES

[1] J. Kumagai, "Life of birds," IEEE Spectrum, Vol. 41, Issue 4, pp. 42-49, April 2004

[2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," in Proceedings of MobiCom 2000, Boston, MA, August 2000

[3] S. Thomson and T. Narten, "IPv6 stateless address autoconfiguration," Network Working Group RFC 2462, December 1998

[4] C. Perkins, J. Malinen, R. Wakikawa, E. M. Belding-Royer, and Y. Sun, "IP address autoconfiguration for ad hoc networks," draft-ietf-manet-autoconf-01.txt, November 2001 (work in progress)

[5] K. Weniger and M. Zitterbart, "IPv6 autoconfiguration in large scale mobile ad-hoc networks," In Proceedings of European Wireless 2002, Florence, Italy, February 2002

[6] N. Vaidya, "Duplicate address detection in mobile ad hoc networks," In Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'02), Lausanne, Switzerland, June 2002

[7] A. Misra, S. Das, A. McAuley, and S. K. Das, "Autoconfiguration, registration, and mobility management for pervasive computing," IEEE Personal Communication System Magazine, Vol. 8, pp. 24-31, August 2001

[8] M. Mohsin and R. Prakash, "IP address assignment in a mobile ad hoc network," In Proceedings of MILCOM 2002, Anaheim, CA, October 2002

[9] S. Nesargi and R. Prakash, "MANETconf: configuration of hosts in a mobile ad hoc network," In Proceedings of the 21st Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM 2002), New York, NY, June 2002

[10] H. Zhou, L. M. Ni, and M. W. Mutka, "Prophet address allocation for large scale MANETs," In Proceedings of the 22nd Annual Joint Conference of IEEE Computer and Communication Societies (INFOCOM 2003), San Francisco, CA, April 2003

[11] H. Zhou, L. M. Ni, and M. W. Mutka, "Prophet address allocation for large scale MANETs," Ad Hoc Networks Journal, Vol. 1, Issue 4, pp 423-434, November 2003

[12] C. Schurgers, G. Kulkarni, and M. B. Srivastava, "Distributed Assignment of Encoded MAC Addresses in Sensor Networks," In Proceedings of MobiHOC 2001, Long Beach, CA, October 2001

[13] C. Schurgers, G. Kulkarni, and M. B. Srivastava, "Distributed On-demand Address Assignment in Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, Vol.13, No.10, pp. 1056-1065, October 2002

[14] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir R. Das, "IP Flooding in Ad hoc Mobile Networks," draft-ietf-manet-bcast-00.txt, November 2001 (expired)

[15] K. Fall and K. Varadhan (editors), The *ns* Manual - the VINT Project, http://www.isi.edu/nsnam/ns/ns-documentation.html, August 2005