

Multicasting in Wireless Sensor Networks

Woohyung Chun
Dept. of ECE
SUNY Stony Brook NY
wchun@ece.sunysb.edu

Wendy Tang
Dept. of ECE
SUNY Stony Brook NY
wtang@ece.sunysb.edu

Abstract— Wireless sensor networks are embedded networks that are highly restricted to energy, bandwidth and processing power. Even though wireless sensor networks operate with limited resources, sensor nodes are required to do more functions compared to nodes in other data networks. Sensor nodes work as both hosts for processing sensed data and routers for forwarding packets. To decrease such a workload in sensor nodes, we propose a multicasting method that minimizes the number of transmission and forwarding in each sensor node. The proposed multicasting algorithm is simulated in the TOSSIM simulator and implemented with Crossbow MICA2 microsensors.

1 Introduction

Wireless Sensor Networks (WSNs) consist of battery-operated sensor nodes with limited processing capability. Such resource constrained characteristics distinguished WSNs from other data networks.

Previous studies [1, 2] introduced energy efficient Medium Access Control (MAC) protocols to prolong WSNs lifetime. The underlying assumptions were that most of the energy consumed by WSNs occurs during communication between sensor nodes and that the energy consumed for handling sensed data was negligible. Designing less energy consumed MAC protocols might be the most effective solution to reduce energy consumption in WSNs.

However, to reduce the actual energy consumption for non-communication processing is *not* trivial. Conversely, the study about habitat monitoring in [3] showed that the energy consumed by WSNs is two times more than that for communications between nodes. Such behavior was also reported for distributed computation and sensor collaboration [4].

To reduce workload in sensor nodes, distributed computation that divides one application into several smaller applications distributed over the sensor

network is feasible. However, since such distributed computation requires more sensor node collaboration and hence increase the amount of sensor to sensor communication, more efficient communication pattern that reduces the amount of transmission and forwarding is desired.

In this paper, we propose to adapt the Internet Protocol (IP) multicasting algorithm to WSNs to reduce the amount of transmission and forwarding in certain applications. Section 2 describes our proposed algorithm. Section 3 discusses the simulation and implementation of the algorithm and its performance compared with traditional unicasting method. Section 4 summarizes our findings.

2 Proposed Algorithm

The main advantage of multicasting is to reduce the number of transmitting and forwarding packets. Our proposed multicasting algorithm uses static multicasting groups defined prior to node deployment. By avoiding dynamic changes in multicasting group such as those in IP multicasting, we eliminated the required control packets for managing multicasting groups and hence reduced network delay and energy consumption. Since we implement multicasting in the application layer above the routing layer in TinyOS [6], actual multicast packets are sent via reliable paths based on unicast routing tables. Such application layer implementation allows our algorithm to be independent of the MAC and routing layers.

2.0 Multicasting Scenario

When multicast packets are generated by applications, the multicast packets are handled by each sensor node. In Figure 1, node 1 sends packet P to multicast group 0x0011 (node 1,4,5,6, and 7). Since nodes 2 and 3 are not the members of the multicast group (address 0x0011), they are just forwarding the multicast packets. Multicast group members (node 1, 4, 5, 6, and 7) do not only run the multicast

application as a host but also relay packets to other multicast group members.

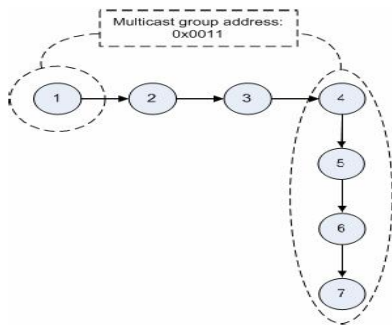


Figure 1. Sending packet P from node 1 to node 4,5,6 and 7 via multicasting transmission.

2.1 Multicasting loop

In Figure 1, if node 1 is within the transmission range of node 7, node 7 can forward the multicast packets to node 1 resulting in an undesirable multicasting loop.

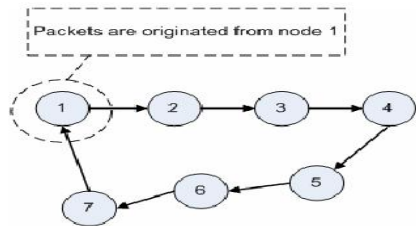


Figure 2. Multicasting loop from source node to source node

To eliminate multicasting loop, we devise Temporary Path Marker (TPM) that records the history of consuming packets in multicast hosts. TPM is a table that contains the group address and payload of received multicast packets. Every sensor node in a multicast group records TPM whenever multicast packets are consumed in the node. In Figure 2, node 1 records TPM when it sends multicast packet P to other multicast group members. Node 4, 5, 6 and 7 also record the same TPM as in node 1. When node 1 receives packet P from node 7, packet P's multicast address and payload are compared with the TPM of node 1. If packet P's multicast address and payload are the same as the TPM of node 1, node 1 will identify packet P received from node 7 as a multicasting loop packet. After that, node 1 will neither forward nor consume the packet P. Packet P is, therefore, discarded in node 1.

Figure 3 shows the proposed multicasting algorithm

including the solution of avoiding multicasting loops.

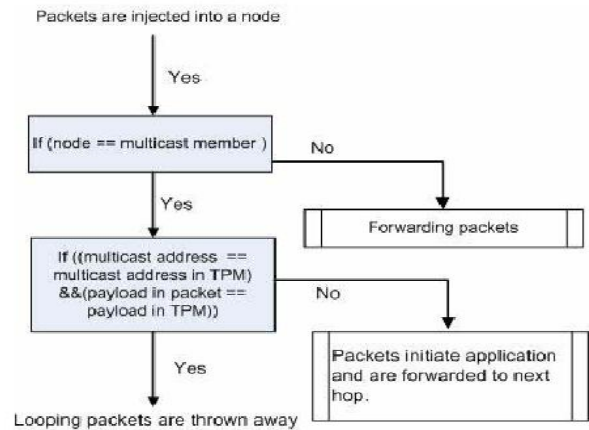


Figure 3. Proposed multicasting algorithm

3 Simulation and Implementation

To verify our multicasting algorithm, we apply our algorithm in a testing benchmark application with four Crossbow Technology Inc's MICA2 motes as sensor nodes, node 0, 1, 2, 3 in a wireless sensor network. Each of these MICA2 motes has three LEDs, *red*, *yellow*, and *green*. The testing benchmark application consists of three functions. First function A (Group address 0x0011 in multicasting): to turn on the red LEDs of nodes 3, 1, and 0. Second function B (Group address 0x0022 in multicasting): to turn on the yellow LEDs of nodes 3, 2, and 1. Third function C (Group address 0x0033 in multicasting): to turn on the green LEDs of nodes 3, 2, and 0. All commands for the three functions were sent from node 3. Figure 4 shows an overview of our testing benchmark application.

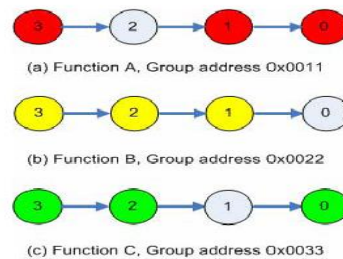


Figure 4. Testing benchmark application.

To compare our multicasting algorithm with typical unicasting, we implemented both multicasting and unicasting implementation for the testing benchmark application. Our testing benchmark application ran on the TOSSIM simulator with power profiling. For the simulation, we used the MICA2 microsensor radio model provided by Crossbow [7]. After the simulation

was completed, the testing benchmark application codes were ported to real MICA2 microsensors. Figure 5 shows the simulation results of both the proposed multicasting and typical unicasting methods.

moteid	radio	cpu	led	eeprom	total
0	414.78	166.72	70.46	0.00	651.97
1	337.17	135.46	57.24	0.00	529.88
2	415.08	166.72	70.58	0.00	652.39
3	415.71	166.79	107.55	0.00	690.04

(a) Power profiling for multicasting

moteid	radio	cpu	led	eeprom	total
0	946.84	380.60	109.24	0.00	1436.68
1	933.23	375.12	114.68	0.00	1423.03
2	927.45	372.80	172.42	0.00	1472.67
3	1014.11	407.27	316.99	0.00	1738.37

(b) Power profiling for unicasting

Figure 5. Simulation results in TOSSIM

To measure the end-to-end delay from the start to the end of the testing benchmark application, we configured the break point in TOSSIM simulator. The simulation results shows that our proposed multicasting is 2.5 times (total execution time for unicasting / total execution time for multicasting = $81.306/32.351 \approx 2.5$) faster than unicasting. Moreover, total energy consumption in the multicasting is only 2524.28 uJ. It is less than half of the total energy consumption in the unicasting (6070.75 uJ). Figure 6 shows the real implementation with MICA2 microsensors.

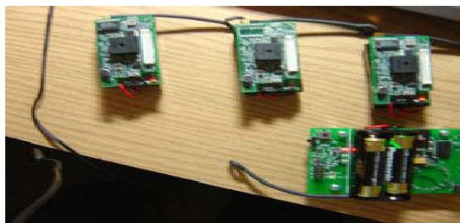


Figure 6. Implementation of multicasting application for test bench

4 Conclusion

We presented a multicasting algorithm that minimizes the number of transmission and retransmission in sensor nodes. Through simulation with the TOSSIM simulator and implementation on MICA2 sensor nodes, we evaluated the performance of our multicasting algorithm and compare its

performance with the typical unicasting method.

We found that, for our testing benchmark application, our proposed multicasting algorithm outperforms unicasting in both total energy consumption and execution time. We also expect that this performance advantage will be more significant in densely deployed sensor networks. In the future work, we plan to apply our multicasting algorithm to more elaborated applications.

ACKNOWLEDGEMENT

This project is partially funded by the National Science Foundation under Grant No. EEC-0332605. Any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- [1] Alec Woo, David E. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks", ACM Proceedings of the 7th annual international conference on Mobile computing and networking, July 2001
- [2] Wei Ye, John Heidemann, and Deborah Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, Vol. 12, No. 3, pp. 493-506, June 2004.
- [3] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, John Anderson, and David Culler, "An Analysis of a Large Scale Habitat Monitoring Application", ACM SenSys'04, November 3-5 2004.
- [4] Arvind Giridhar and P. R. Kumar, "Towards a Theory of In-Network Computation in Wireless Sensor Networks", IEEE Communications Magazine, vol. 44, no. 4, pp. 98-107, April 2006.
- [5] Victor Shnayder, Mark Hempstead, Bor-rong Chen and Matt Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", ACM Proceedings of the 2nd international conference on Embedded networked sensor systems, November 2004.
- [6] TinyOS Web, <http://www.tinyos.net>
- [7] Crossbow Web, <http://www.xbow.com>