Poster Abstract: Infuse: A TDMA Based Reprogramming Service for Sensor Networks*

Mahesh (Umamaheswaran) Arumugam
Department of Computer Science and Engineering
Michigan State University, East Lansing MI 48824
arumugam@cse.msu.edu

ABSTRACT

Programming or upgrading the software in sensor networks is one of the important problems since the sensors are often deployed in large numbers and in hostile environments. In this paper, we present *infuse*, a reliable over the air reprogramming service for sensor networks using time division multiple access (TDMA) protocol. In case of unexpected channel errors, a back pressure mechanism is used to ensure reliable propagation. Moreover, we show that infuse propagates code in a pipeline and in an energy-efficient manner.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: [Network Operations]; D.4.4 [**Operating Systems**]: [Communications Management]

General Terms

Algorithms, Reliability

Keywords

Network programming, TDMA, Sensor networks

1. INTRODUCTION

Programming or upgrading the software in the sensors (e.g., MICA motes [1]) is one of the important problems since the sensors are often deployed in large numbers and in hostile environments [2]. Additionally, the requirements of the application often evolve over time and, hence, the network should have a facility to replace the software. Towards this end, wireless reprogramming is proposed [3]. This solution is applicable only to single-hop networks, Most of the proposed deployments are multihop in nature and, hence, a multihop reprogramming feature is necessary. Many multihop reprogramming algorithms are proposed in the literature (e.g., [4–6]). These algorithms use carrier sense multiple access (CSMA) protocols and, hence, rely on other mechanisms (e.g., error-correcting codes, acknowledgments/negative acknowledgments, etc) for reliable communication.

In this paper, we propose *infuse*,[1] a reprogramming service for sensor networks using time division multiple access (TDMA) (e.g., [7]). TDMA provides significant advantages for reprogramming in sensor networks since it offers high reliability in message communication. Infuse can be used with any TDMA protocol. It requires neighborhood information (successors in the east/south direction), which can be obtained during localization [8]. Although TDMA is collision-free, messages can be lost due to unexpected channel errors (message corruptions, varying signal strengths, etc) caused by changing communication characteristics. To deal with such errors, we propose a back pressure mechanism.

2. INFUSE: ALGORITHM AND RESULTS

Code dissemination algorithm in ideal scenario. The base station located at $\langle 0, 0 \rangle$ (for simplicity) initiates reprogramming by sending a *start-download* message. Whenever a sensor receives the start-download message, it initializes the flash to accept the new program and signals the current application that a download is in progress. Finally, it forwards the message in its next TDMA slot. Once the base station sends the *start-download* message, it starts transmitting program capsules (a capsule is 16-bytes of instructions) of the new program in its TDMA slots one by one. Whenever a sensor receives a capsule, it stores it in the appropriate location and forwards it in its TDMA slot.

Implicit acknowledgments and back pressure mechanism. The above algorithm guarantees reliable code-propagation if there are no channel errors. In presence of such errors, program capsules can be lost. To overcome this difficulty, we introduce back pressure mechanism. In the above algorithm, whenever the east/south neighbor of a sensor (say, j) forwards the program capsule (say, c_l), j gets an implicit acknowledgment for c_l. We use this information to recover from lost capsules. Towards this end, each sensor maintains a buffer of size $2b$, where b is an integer. In other words, sensor j maintains a buffer of $c_{la}+1, \ldots, c_{la}+2b$ capsules, where c_{la} is the highest implicitly acknowledged capsule. In order to deal with lost capsules, j will forward capsule c_f only when its successors (east/south neighbors) have forwarded at least capsule $c_f - b$, i.e., j has received implicit acknowledgment up to capsule $c_f - b$. Otherwise, j will start retransmitting from capsule $c_{la}+1$ (cf. Figure 1).

Dealing with failed sensors. To deal with failed sensors, whenever a sensor fails to get an implicit acknowledg-

---

*Acknowledgments: Dr. Sandeep S. Kulkarni for his assistance and support in this work. This work was partially sponsored by NSF CAREER CCR-0092724, DARPA Grant OSURS01-C-1901, ONR Grant N00014-01-1-0744, NSF Equipment Grant EIA-0130724, and a grant from Michigan State Universit

[1] infuse v.; to cause to be permeated with something (as a principle or quality) that alters usually for the better ⟨infuse the team with confidence⟩. Source: Merriam-Webster Online, `http://www.m-w.com/`.

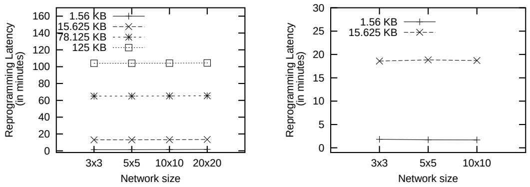
SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
ACM 1-58113-879-2/04/0011.

```
highest_acknowledged_capsule = $min$(highest capsule for
        which implicit acknowledgment is
        received from south/east neighbors);
next_capsule++;
if next_capsule > highest_acknowledged_capsule +$b$
    next_capsule = highest_acknowledged_capsule +1;
forward next_capsule in the next TDMA slot;
```

Figure 1: Back pressure mechanism

ment from its successors after a fixed number of retransmissions, it ignores that neighbor. A sensor will retransmit a capsule only when it does not receive an implicit acknowledgment from its active neighbors.

Theoretical lower-bound on reprogramming latency. With TDMA, the program capsules are transmitted in a pipeline where each intermediate sensor acts as a sender and as a receiver simultaneously. For example, when the base station sends capsule 10, sensors in the east and south of base station receive the capsule simultaneously. These sensors will forward this capsule in their next TDMA slots. Now, the base station can forward capsule 11 in its next TDMA slot. Similarly, all capsules are transmitted in the network in a pipeline fashion. To reprogram a $n*n$ network with a program containing c capsules, the lower bound on reprogramming latency is $(c-1+2(n-1))*P$ amount of time, where P is the period between successive TDMA slots. For example, to reprogram a $10*10$ network with a 1000 capsule program (i.e., 15.625 KB program), the lower bound is 13.22 minutes, if $P=0.78s$. Figure 2(a) shows the lower-bound for different program sizes. For a given program size, the latency increases only slightly when the network size increases. This illustrates the effect of pipelining.



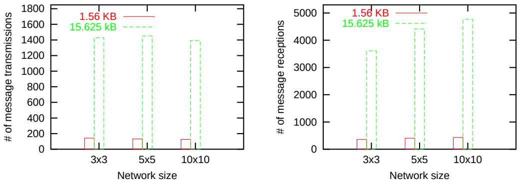
(a) Analytical result (b) Simulation result

Figure 2: Reprogramming latency

Energy-efficiency. Whenever a sensor receives a capsule from north neighbor, it can turn off its radio in the slot alloted to its west neighbor. The sensor should listen in the slots alloted to its south and east neighbors to receive the implicit acknowledgments. In short, if the period between successive slots assigned to a sensor is P, a sensor needs to listen to the radio in *at most* 4 slots during every P slots.

Simulation and implementation results. We simulated infuse on prowler [9]. We measured the reprogramming latency, number of message transmissions and receptions with infuse. The network sizes used in the simulations are 3x3, 5x5, and 10x10 with 95% link reliability. The network is reprogrammed with 1.56 KB (100 capsules) and 15.625 KB (1000 capsules) programs. Figure 2(b) shows the reprogramming latency. For a given program size, the latency remains almost the same for different network sizes. This illustrates the effect of pipelining. Further, the latency is within acceptable limits from the theoretical lower bound.

Figure 3(a) shows the number of message transmissions. The number of transmissions remains almost the same for different networks and is less than 1.5 times the optimum value (=size of the new program). Figure 3(b) shows the number of message receptions. Number of receptions is almost the same for different networks. Thus, infuse propagates program capsules in an energy-efficient manner.



(a) Sends (b) Receives

Figure 3: Energy-efficiency

We have also implemented infuse for mica-2 motes. Currently, we are testing the service for different parameters.

3. CONCLUSION AND FUTURE WORK

In this paper, we proposed infuse that takes advantage of the reliability offered by TDMA protocol. We showed that infuse forwards program capsules in a pipeline and, hence, latency is reduced. Further, we showed that infuse propagtes code in an energy-efficient manner.

We are currently investigating the performance of infuse in large scale networks. One possible scheme is to partition the network into different sections and reprogram the sections independently. Also, we are investigating ways to extend infuse to support dynamic adaptation in sensor networks.

4. REFERENCES

[1] J. Hill and D. E. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 2002.

[2] A. Arora et al. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 2004.

[3] Crossbow Technology, Inc. *Mote In-Network Programming User Reference*, 2003.

[4] J. Kulik, W. R. Heinzelman, and H. Balakrishnan. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8(2-3):169–185, 2002.

[5] T. Stathopoulos, J. Heidemann, and D. Estrin. A remote code update mechanism for wireless sensor networks. Technical Report CENS-TR-30, University of California, Los Angeles, Center for Embedded Networked Computing, November 2003.

[6] S. S. Kulkarni and L. Wang. MNP: Multihop network reprogramming service for sensor networks. Technical Report MSU-CSE-04-19, Michigan State University, 2004.

[7] S. S. Kulkarni and M(U). Arumugam. TDMA service for sensor networks. *In Proceedings of the Third International Workshop on Assurance in Distributed Systems and Networks (ADSN)*, March 2004.

[8] G. Agha et al. Evaluation of localization services (preliminary report), 2003. DARPA NEST Program. Available at: `http://osl.cs.uiuc.edu/docs/nest-localization-report-2003/`.

[9] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi. Simulation-based optimization of communication protocols for large-scale wireless sensors networks. *In Proceedings of the IEEE Aerospace Conference*, 2003.