Poster Abstract: Rateless Codes for Data Dissemination in Sensor Networks

Andrew Hagedorn, David Starobinski, and Ari Trachtenberg
Department of Electrical and Computer Engineering, Boston University
8 Saint Mary's Street
Boston, MA 02215, USA
{achag ,staro, trachten}@bu.edu

ABSTRACT
This paper discusses the use of rateless codes to increase performance in wireless sensor networks.

Categories and Subject Descriptors
C.2.1 [Computer-Communication Networks]: Network Architecture and Design – Wireless Communication.

General Terms
Algorithms, Design, Experimentation, Performance.

Keywords: Wireless, Motes, Rateless Codes.

1. INTRODUCTION
Efficient data dissemination is intrinsic to a wide variety of wireless sensor networks (WSNs) applications, such as database services, over-the-air programming (OAP), and network management.

The broadcast nature of the channel in WSNs significantly impacts the performance of data dissemination, especially at high network density. For example, as the number of nodes increases, so does contention on the shared channel: receivers keep sending requests for packets they are missing while senders may have to retransmit each packet several times to ensure correct reception by all the nodes. Traditional sensor limitations, such as low memory, fixed battery life, and bounded computation capabilities only exacerbate this problem.

Rateless codes [2] provide an efficient means of addressing channel contention in WSNs, while at the same time minimizing control messages, such as those that contribute to the so-called ACK/NACK implosion problem. Fundamental to this approach is the fact that receivers do not need to indicate which specific packets require retransmission; instead, they just wait for receipt of a sufficient number of different packets, which can then be used to decode the original message. Rateless coding, thus, yields several key benefits, namely, communication and energy savings, lower control overhead, and also certain measures of robustness and security [1].

Our main contribution of this work is to concretely and efficiently implement a real-time rateless coding system on sensor nodes. In this paper we discuss some of the major technical challenges and also present preliminary experimental results on the average delay of data dissemination with and without rateless coding in a single-hop network of n nodes with obstructions.

SenSys'06, November 1–3, 2006, Boulder, Colorado, USA.
ACM 1-59593-343-3/06/0011

2. ENCODING METHOD
This section presents the method for encoding PC files on a base station and transmitting to decoding nodes. Due to the nodes memory constraint of 4 KB, the files on a PC must be parsed into smaller sub-files that are encoded and broadcast individually. Each sub-file is transmitted from the PC to the base station, which encodes its content using random linear codes; this corresponds to parsing the file into k "chunks" and multiplying the resulting vector by a random matrix over some finite field F_q. The code is constructed using a random number generator seeded with a secret key and a unique identifier known only to the base station and decoders. It is important to note that rank is very sensitive to the quality of the randomness and, thus, a strong random number generator is needed. This encoding transforms the sub-files into a sequence of $k' > k$ vector "chunks" of length m which are packaged with a unique identifier and a CRC of the sub-file and broadcast to the decoders. In this implementation a single chunk is equivalent to a packet, however, for different values of m it is possible to have multiple chunks per packet.

To begin decoding, the decoders must receive any k linearly independent encoded packets. The unique identifier of the packets and the secret key are used to regenerate the random code. Gaussian elimination with back substitution is then used to solve the set of linear equations to retrieve a sub-file. A CRC of the decoded sub-file is computed and if it is correct the decoder informs the base station. Alternatively, it may be that the matrix was not of full rank and decoding has failed; in this case the decoder discards its packets and an additional k packets are received. Once all decoders have retrieved the sub-file, a new sub-file can be processed.

3. TRADE-OFFS

One trade-off that significantly affects the performance of the system is that of memory requirements versus computational complexity in the implementation of finite field arithmetic. To perform the arithmetic on the fly is computationally demanding: multiplication is modulo an irreducible polynomial and division requires the extended Euclidian algorithm to calculate multiplicative inverses. This can be alleviated by storing tables of pre-computed values in memory at the expense of the maximum sub-file size (which resides within the same memory). Our implementation uses both methods; multiplication is computed on the fly while the multiplicative inverses are stored as a 256 byte table in memory.

A second trade-off is between the probability for decoding failure and resource use, which depends on the selection of q, the size of the finite field. A large field increases memory and computational complexity, but provides a high probability of decoding correctly. For example, a field size of $q=2^{32}$ yields a probability for decoding failure of 2×10^{-10} for a 25-packet file. On the other hand, small fields reduce resource use, but have a lower probability to decode. In our implementation, the field size is fixed at $q=2^{8}$ matching the 8 bit processor on the motes and allowing for more computationally efficient arithmetic; it also has an acceptable decoding failure probability of 0.00392.

4. EXPERIMENTS

4.1 Comparison Model

To allow for a fair comparison, a plain data dissemination method is used. For each sub-file transmitted from the PC, the base station parses the sub-file into k packets and broadcasts each packet to the receiving nodes. Once all packets have been transmitted, the base station polls the state of the receiving nodes by broadcasting a control packet. The receiving nodes respond in one of two ways: that the sub-file has been fully received or with a bit vector indicating the packets that were missed. The base station broadcasts the packets that were missed and polls the network until all receiving nodes have indicated that all packets have been received.

4.2 Experimental Method

Our experimental results compare the performance of encoded to plain data dissemination on our test bed of ten 900 MHz MICA2 motes; this consists of one base station and up to nine receiving nodes within the base station's range. Each point shown represents an average taken over twenty experiments (with the 90% confidence level marked) with the following parameters: each sub-file is parsed into k=25 chunks and the length of each chunk, m, is 23 bytes. The choice of m corresponds to the maximum transfer unit of the network and results in 575 byte sub-files.

4.3 Results

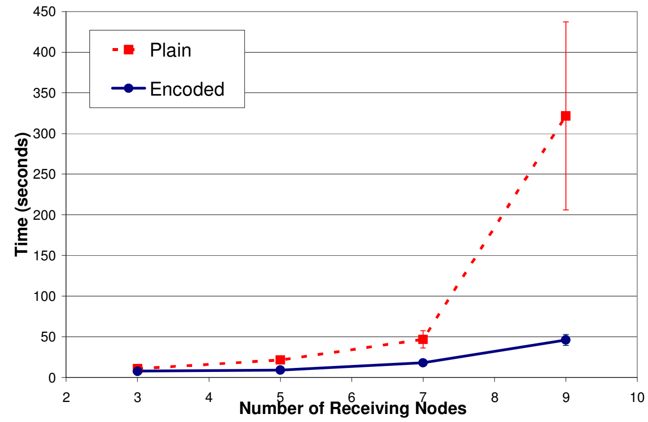

Fig. 1. Average Dissemination time with an obstruction

The first experiment was designed to model an ideal link with negligible packet losses; nodes were placed in close proximity to the base station with no obstruction. In this case, plain data dissemination is faster than encoding by an average of 1.32 seconds. This represents the computational overhead associated with random linear coding.

The second experiment attempted to model a link with non-negligible channel losses; as such, a cardboard box with a single layer of tin foil was placed around the base station. In this case, Figure 1 shows that the rateless codes are clearly faster than the plain implementation. For instance, when there are 9 receiving nodes the encoded sub-file is transferred in 46.14 seconds on average, while the plain sub-file takes 321.59 seconds. The main reason for this significant improvement is that the rateless codes have a reduced amount of control signals; this results in fewer MAC level collisions in addition to lessened bandwidth usage. In general the rateless codes reduce the total amount of transmission needed and, in doing so, reduce the impact of packet losses.

These experimental results show the potential of using rateless codes for data dissemination in environments with non-negligible packet losses.

5. ACKNOWLEDGMENTS

This work is partially funded by the National Science Foundation grants CCR-0133521 and CNS-0435312.

6. REFERENCES

[1] S. Agarwal et al., "Fast data access over asymmetric channels using fair and secure bandwidth sharing," Proc. *ICDCS 2006*.

[2] J. Byers et al., "A Digital Fountain Approach to Reliable Distribution of Bulk Data," Proc. *SIGCOMM 1998*.