Distributed CDMA Code Assignment for Wireless Sensor Networks

Hwee-Xian Tan[2,1], Winston K. G. Seah[1,2] and Kwang-Mien Chan[1]
{stuhxt, winston, chankm} @ i2r.a-star.edu.sg

[1]Networking Department, Institute for Infocomm Research, A*STAR, Singapore

[2]Department of Computer Science, School of Computing, National University of Singapore

***Abstract*—Wireless Sensor Networks are deployed for many collaborative tasks such as monitoring, target tracking and surveillance. They communicate via multi-hops to one or more sinks over a shared wireless medium. Code Division Multiple Access (CDMA) allows many sensor nodes to transmit simultaneously over the common shared medium without excessive collisions that often result in packet loss and lowered throughput. However, this requires the proper assignment of orthogonal codes to the sensor nodes to minimize interference among adjacent neighbors. Due to the distributed and autonomous nature of sensor networks, CDMA codes cannot be assigned in the same way as cellular networks. We propose a distributed methodology of assigning CDMA codes to the nodes, which has low message overheads, allows for spatial reuse of the channel and does not depend on the availability of neighbor information. Our approach is also able to handle topological changes that could be induced by nodes entering or leaving the network. We study the effectiveness of our algorithm through simulations and analysis.**

***Keywords*—Code division multiaccess, distributed algorithms, communication systems, wireless sensor networks.**

I. INTRODUCTION

Wireless sensor networks are often densely deployed in a distributed manner, for collaborative missions such as monitoring, target tracking and surveillance. They usually consist of a group of sensor nodes that communicate via wireless multi-hops to one or more centralized sinks, which collect and analyze data in real time.

The nature of the applications of sensor networks as well as their operating environment poses much design constraints which are quite different from that of wired networks. Sensor networks are self-configuring and self-organizing, while wired networks usually have centralized control, which makes it easier for coordination and synchronization purposes. Sensor networks are also more resource-constrained than the latter (in terms of bandwidth, memory and energy). Therefore, the unique characteristics of sensor networks must be taken into consideration during the architectural and operating design of the network.

Nodes in wireless networks typically communicate via a shared communication channel, resulting in the well-known hidden/exposed terminal problems. To alleviate the complications associated with hidden/exposed nodes, many Medium Access Control (MAC) protocols have been proposed in the literature. Classical MAC protocols can be classified as deterministic (e.g. TDMA and FDMA) or non-deterministic (e.g. ALOHA, slotted ALOHA, CSMA, CSMA/CA, MACA, MACAW, BTMA and IEEE 802.11) [1]. The latter is also known as random access protocols which are contention-based (i.e.: nodes compete to transmit data at various times) and do not guarantee access to the communication channel. While they can be implemented easily in a distributed network, they are unable to avoid collisions completely. Deterministic protocols are more effective in eliminating collisions but require more complexity and coordination.

As these schemes are unlikely to work well in distributed, unsynchronized and low-bandwidth network conditions, CDMA (Code Division Multiple Access), which is a hybrid between random access and fixed channel allocation schemes, can be used to allow many sensor nodes to transmit simultaneously over the common shared wireless medium without collisions. However, due to the concurrent transmission of data packets by neighboring nodes, CDMA-based networks suffer from higher multi-user interference as compared to TDMA and FDMA. This can be reduced significantly by the proper assignment of orthogonal codes to the sensor nodes.

While CDMA has been widely used in cellular networks which have base stations to coordinate the access and distribute codes to the mobile terminals, these methods are not directly applicable to sensor networks which are distributed and autonomous in nature. In this paper, we propose a distributed algorithm for assigning CDMA codes to sensor nodes; this algorithm has low message overheads, allows for spatial reuse of codes and does not depend on the availability of neighbor information.

The rest of this paper is organized as follows: Section II discusses related work on code assignment in CDMA-based networks. In Section III, we describe our distributed code assignment algorithm for sensor networks. Simulation results and analysis are presented in Section IV. We conclude with directions for future work in Section V.

II. Related Work And Motivation

CDMA is the de-facto standard for multiple access in many mobile cellular systems throughout the world. Each mobile user encodes its signal with a pseudo-random codeword that is almost orthogonal to other users to reduce the amount of multi-user interference and thus provide good quality signals. Spatial reuse of the shared wireless medium is achieved by the proper assignment of codewords by the base station to the different users in the network.

In a CDMA network, nodes are expected to be able to transmit and/or receive data over multiple codes. In Receiver-Oriented Code Assignment (ROCA) schemes, the receiver is able to receive only on a fixed, pre-allocated code. The transmitter must therefore be able to transmit at different codes to the intended destinations. In Transmitter-Oriented Code Assignment (TOCA) schemes, the transmitter is able to send only in one intended code; therefore the receivers must be able to receive and decode multiple types of codes. The Pairwise-Oriented Code Assignment (POCA) scheme assigns codes in a pairwise manner to each transmitter-receiver pair.

Hu [2] introduces a POCA-based, two-phase algorithm which makes use of a breadth-first-search tree to maintain the order of code assignment, thus incurring extra time and message overheads. Code inconsistencies are gradually resolved through iterative cycles and fewer codes are required at the expense of lowered throughput.

Bertossi and Bonuccelli in [3] propose distributed heuristic algorithms for code assignment in TOCA-based networks which are assumed to be asynchronous and can communicate via the exchange of control messages. However, their scheme which requires knowledge of neighbour information has high message complexity and does not cater to dynamic topologies.

Other distributed and unsynchronized code assignment algorithms have also been proposed, such as in [4] and [5], which can be used for both TOCA and ROCA based systems. However, a common prerequisite is the knowledge of neighbor information via a neighbor discovery mechanism, before the actual code assignment commences.

More recently, Guo et al in [6] have described an algorithm that relies on nodes having to listen to a common channel and periodically broadcasting channel assignment packets to its neighbors. The common channel is monitored for a random time with a predefined contention window, thus increasing the time taken for network initialization. In addition, periodic broadcast of the assignment packets can increase contention in the shared channel and expend more energy in the nodes. Yet another scheme [7] also requires the use of a separate control channel and knowledge of two-hop neighboring information in order to assign codes to the nodes in the network.

While many code assignment algorithms have been proposed in the literature, most are designed for the mobile cellular network, which have centralized base stations for the coordination of message or data exchanges. Some distributed code assignment algorithms have also been proposed, but they often require a neighbor discovery mechanism and/or a common control channel. In the next section, we outline some of the design principles that we have adopted and describe our code assignment algorithm in detail.

III. A Distributed Code Assignment for CDMA-based Sensor Networks

A. Problem Formulation

We consider a half-duplex system where nodes can either receive or transmit at any one time. The network can be modeled as an undirected graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ = {set of vertices of nodes in the network} and $\boldsymbol{E}$ = {set of edges or links between the nodes in the network}. Therefore, two adjacent nodes $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ share a common edge [$\boldsymbol{e_{12}}$]. Hidden terminal interference may occur when two non-adjacent nodes share a common intermediate node, i.e.: nodes $\boldsymbol{v_i}$, $\boldsymbol{v_j}$ and $\boldsymbol{v_k}$ are connected in such a way that [$\boldsymbol{e_{ij}}$] and [$\boldsymbol{e_{jk}}$] $\in \boldsymbol{E}$, and [e_{ik}] $\notin \boldsymbol{E}$. Therefore, the purpose of the code assignment algorithm in a network is to assign a fixed set of finite codes to all the nodes in the network, such that $\boldsymbol{v_i}$ and $\boldsymbol{v_k}$ have different codes. For a full-duplex system whereby nodes can transmit and receive at the same time (which is not recommended for CDMA networks due to the near-far problem), the problem is further constrained such that $\boldsymbol{v_i}$, $\boldsymbol{v_j}$ and $\boldsymbol{v_k}$ must have different codes.

B. Design Principles for Distributed Code Assignment

We base our distributed code assignment algorithm on the following design principles:

1) Simplicity – as little assumptions should be made as possible, for example, here we do not consider the presence of an explicit neighbor discovery mechanism;
2) Time complexity – the algorithm should take a finite and reasonably short time to converge or stabilize; i.e.: all nodes in the network should receive a permanent code after a finite, fixed time $\boldsymbol{\tau}$;
3) Message complexity – message overhead is minimized as far as possible without retransmissions and acknowledgements, and control messages are sent only when it is necessary;
4) Storage complexity – as little storage overhead should be incurred as far as possible;
5) Robustness – the algorithm is expected to be able to work even in harsh network conditions with partitions (partitioned nodes are assigned with a code after a finite time has passed) and low bandwidth;
6) Stability – the algorithm should converge after some time, i.e.: all nodes should be assigned a fixed code without further changes, by the end of the network

initialization phase;

7) Constrained resources – we assume the presence of a finite set of codes, and handle cases whereby the number of orthogonal codes is not sufficient to support the large number of nodes; and
8) Dynamic – the algorithm is able to handle topological changes, which could be induced by new nodes entering the network, etc.

C. Code Assignment

We propose a distributed and unsynchronized way of code assignment to nodes in the network such that each node will not use the same code as its two-hop neighborhood. Two types of messages are used:

1) Update messages, which contain the sender's code and codes chosen by the sender's one hop neighbors; and
2) Conflict messages, which also contain the sender's code and the codes that are used by the one-hop neighbors.

We assume that the network initialization phase is triggered by one or more reference nodes (which can be sinks that are likely to be placed in the corners of the terrain) that first select their own codes. Update messages are initiated when: (i) a timeout occurs at node S_1; (ii) S_1 (including the reference nodes) selects a new code; (iii) S_1 changes its code (due to a conflict with one of its two-hop neighbors); or (iv) the one hop neighbours of S_1 select or change their codes. A new node that enters the network after the initialization phase will also perform the same operations to select a code for transmission.

When the receiver R_1 receives this message, it will check whether the code selected by sender S_1 is the same as one of the codes used by the one hop neighbours (excluding S_1) of receiver R_1. If there are no conflicts, R_1 will update its one-hop and two-hop neighborhoods. However, if R_1 detects a conflict in code reuse, it will send a conflict message to S_1 immediately.

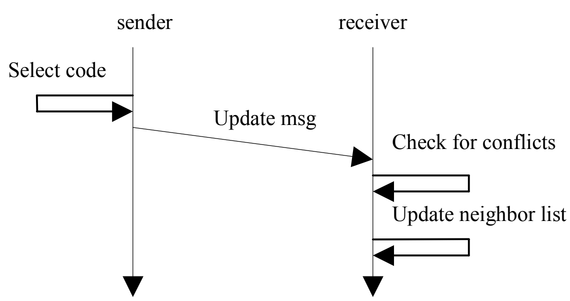

Figure 1 Interaction between sender and receiver without code conflict

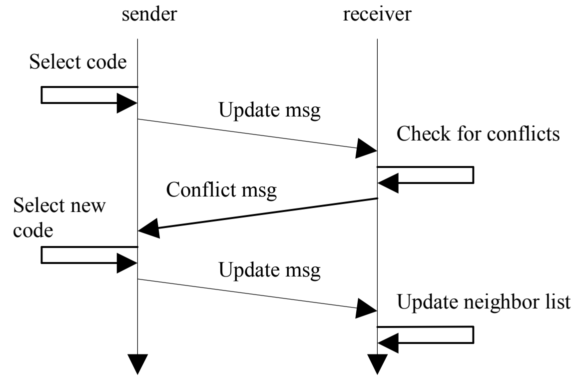

Figure 2 Interaction between sender and receiver during code conflict

Upon receiving the conflict message, S_1 will update its one-hop and two-hop neighborhood, including the codes that have been used by each of these neighbors. It will then select another code based on its two-hop neighbor lists, and send another update message with its newly selected code. The interactions between each pair of sender S and receiver R are shown in Figure 1 and Figure 2.

Upon receiving an update message, each node also waits for a random delay before sending an update message (if one should be initiated according to one of the four pre-defined conditions stated earlier). This reduces the probability of node ***i*** and node ***k*** (both of which have chosen the same code) sending update messages at the same time, which could cause collisions at intermediate node ***j***, and consequently prevent the detection of conflicts should one occur (see Figure 3).

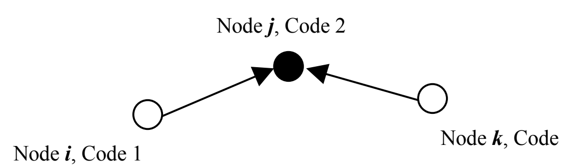

Figure 3 Occurrence of a collision during conflict

In our approach, we have avoided use of explicit acknowledgements or confirmation messages between nodes, as these messages are likely to congest the limited bandwidth. Instead, we have chosen to use a feedback mechanism, i.e.: conflict messages, as a means of resolving code conflicts during the initialization phase, because we do not expect a high frequency of occurrence of code conflicts in our scenario. This can be done by choosing an appropriate size for the code sequence, based on the expected topology (assuming a known fixed terrain size and an approximate number of nodes in the region). Parallelism is exploited with the distributed nature of the algorithm, as well as the use of more than one reference node to trigger the initialization phase from different corners of the network (in the case of a single sink, this could be the only reference node). The independence on an explicit neighbour discovery mechanism also helps to shorten the code assignment phase and reduce message overheads considerably.

IV. Simulation Results and Analysis

We study the effectiveness of our code distribution/assignment algorithm through simulations that are performed in Qualnet [10]. The terrain size is 2000 metres × 2000 metres, and the transmission range of each node is approximately 280 metres. There are four reference nodes, each of which is placed in a corner of the network. The number of available codes for use, $\boldsymbol{p}$ = 64 throughout our simulations.

We study the performance of our distributed code assignment scheme according to the following performance measures:

1) Control overheads – the total amount of update messages and conflict messages that are generated by the nodes during the process of code assignment; and
2) Accuracy – percentage of nodes that have been allocated correct codes after the code assignment phase.

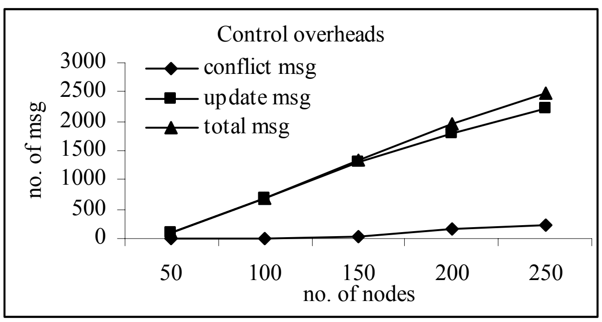

Figure 4 Control overheads vs number of nodes

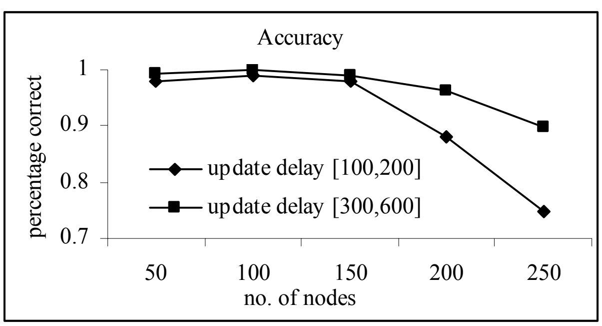

Figure 5 Accuracy of scheme vs number of nodes

Figure 4 and Figure 5 show the control overheads and the accuracy of the scheme respectively, when the number of nodes in the network increases. As can be seen from Figure 4, most of the control overheads are contributed by update messages that are generated by the nodes, and there is relatively less conflict messages that are in the network. In Figure 5, we vary the range of the update delay that each node waits before sending an update message, and see that larger update delays can result in better accuracy of the scheme (although this may lead to a longer initialization phase). In our simulation studies, it is sometimes impossible to achieve full accuracy because the nodes in the network are randomly distributed and there may be insufficient codes to allocate to the nodes.

V. Conclusion and Future Work

Code assignment is a non-trivial problem in CDMA networks, especially for wireless ad hoc networks which do not have centralized control. Due to the collision-free property of CDMA, it is fast becoming an alternative to the traditional random access and deterministic MAC protocols, especially in harsh environments with low bandwidth, such as in underwater scenarios [8][9]. Consequently, efficient algorithms for code assignment need to be developed for enabling communications in such situations.

We have proposed a novel and efficient code assignment algorithm that can operate without the assumption of an explicit neighbor discovery mechanism or control channel. The scheme is simple, distributed and exploits parallelism to shorten the network initialization phase. The correctness and efficiency of the algorithm has also been shown through some analysis and simulations.

Future work will include the study of the convergence time analysis of our scheme and different methods of handling the case of insufficient codes during the code assignment phase.

References

[1] S. Kumar, V. S. Raghavan and J. Deng, Medium Access Control protocols for ad hoc wireless networks: a survey, Ad Hoc Networks Journal (Elsevier), Article in Press, Dec 2004.

[2] L. Hu, Distributed Code Assignments for CDMA Packet Radio Networks, IEEE/ACM Transactions on Networking, Vol. 1, No. 6, Dec 1993.

[3] A. A. Bertossi and M. A. Bonuccelli, Code Assignment for Hidden Terminal Interference Avoidance in Multihop Packet Radio Networks, IEEE/ACM Transactions on Networking, Vol. 3, No. 4, Aug 1995.

[4] J. J. Garcia-Luna-Aceves and J. Raju, Distributed Assignment of Codes for Multihop Packet-Radio Networks, Proceedings of IEEE MILCOM ’97, Monterey, California, Nov 1997.

[5] S. Khuri, M. Moh and F. Chung, Code Assignments in CDMA Networks: Distributed Algorithms and Genetic Algorithm Heuristics, Proceedings of the IEEE Singapore International Conference on Networks (SICON), Singapore, Jul 1998.

[6] C. Guo, L. C. Zhong and J. M. Rabaey, Low Power Distributed MAC for Ad Hoc Sensor radio Networks, Proceedings of IEEE Global Telecommunications Conference (GLOBECOM ’01), San Antonio, TX, USA, Nov 25-29, 2001.

[7] X. Y. Li and P. J. Wan, Theoretically Good Distributed CDMA/OVSF Code Assignment for Wireless Ad Hoc Networks, Proceedings of 11th International Computing and Combinatorics Conference (COCOON 2005), Kunming, Yunnan, Aug 16-19, 2005 (to appear).

[8] F. Salva-Garau and M. Stojanovic, Multi-Cluster Protocol for Ad Hoc Mobile Underwater Acoustic Networks, Proceedings of IEEE OCEANS ’03 Conference, San Diego, CA, Sep 2003.

[9] F. Akyildiz, D. Pompili and T. Melodia, Underwater Acoustic Sensor Networks: Research Challenges, Ad Hoc Networks Journal (Elsevier), Article in Press, 2005.

[10] Qualnet Network Simulator, Scalable Network Technologies, Inc. [www.scalable-networks.com]