Brief Announcement: Information Dissemination on Multiple Channels

Stephan Holzer[1], Yvonne-Anne Pignolet[2], Jasmin Smula[1], Roger Wattenhofer[1]
[1]Computer Eng. and Networks Laboratory (TIK), ETH Zurich, Switzerland
[2]ABB Corporate Research, Dättwil,Switzerland
{stholzer,smulaj,wattenhofer}@tik.ee.ethz.ch, yvonne-anne.pignolet@ch.abb.com

ABSTRACT

This article presents an algorithm for detecting and disseminating information in a single-hop multi-channel wireless network: k arbitrary nodes have information they want to share with the entire network. Neither the nodes that have information nor the number k of these nodes are known initially. This communication primitive lies between the two other fundamental primitives regarding information dissemination: broadcasting (one-to-all communication) and gossiping (total information exchange). The time complexity of the algorithm is linear in the number of information items and thus asymptotically optimal with respect to time. The algorithm does not require collision detection and thanks to using several channels the lower bound of $\Omega(k + \log n)$ established for single-channel communication can be broken.

Categories and Subject Descriptors

F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures; C.2.4 [**Distributed Systems**]

General Terms

Algorithms, Theory

1. INTRODUCTION

In this paper we study a basic communication primitive for wireless networks without collision detection.

PROBLEM 1.1 (INFORMATION EXCHANGE). *Consider a network of n nodes with an arbitrary subset of $k \leq n$ nodes where each of these k nodes (called* reporters*) is given a distinct piece of information. The* Information Exchange Problem *consists of disseminating these k information items to every node in the network. The subset of the nodes with information items is not known to the network.*

We restrict ourselves to the simplest possible network topology, the single-hop network, where every node can communicate directly with each other node, with multiple communication channels available. I.e., we generalize the *Information Exchange Problem* [3] (also known as *k-Selection* [4] and *Many-to-All Communication* [2]) for networks with several communication channels. We study a static case where a worst-case adversary inserts k information items at the beginning of the first time slot and no more items are inserted later.

Let the network consist of a set of n nodes, each node v with a built-in unique ID id_v known to all other nodes. We assume time to be divided into synchronized time slots and each message can only contain a constant number of information items. In each time slot a node v chooses a channel c and performs one of the actions *transmit* (v broadcasts on channel c) or *receive* (v monitors channel c). A transmission is successful, if exactly one node is transmitting on channel c at a time, and all nodes monitoring this channel receive the message sent. If more than one node transmits on channel c simultaneously, listening nodes can neither receive any message due to interference (called a *collision*) nor do they recognize any communication on the channel (the nodes have *no collision detection* mechanism).

As bandwidth is typically fixed (i.e., a constant number of information items fit into one message), a lower bound on the time complexity for the Information Exchange problem is $\Omega(k)$: at any point in time the message from at most one node can be received successfully on one channel. Furthermore we assume that each node can only monitor one channel at a time and needs to receive all k items. In this paper we propose an algorithm that solves the problem in asymptotically optimal time complexity for any k with high probability in n.[1] In addition we construct an algorithm that solves the Information Exchange problem even if k is unknown.

It turned out that just estimating the number of nodes with information items k (e.g. using [1]) and then let all these nodes send with probability $1/k$ will solve the task in time $O(k)$ whp$_{2^k}$. As long as $k \in o(\log n)$ this is not whp$_n$. Thus a more sophisticated method is necessary to tackle this problem efficiently and have a high success probability for all values of k.

[1]An event $\mathcal{E}$ occurs with high probability in x (whp$_x$), if $\Pr[\mathcal{E}] \geq 1 - \frac{1}{x^\alpha}$ for any fixed constant $\alpha \geq 1$. By choosing α, this probability can be made arbitrarily low. Usually one is interested in whp in n.

PODC'11, June 6–8, 2011, San Jose, California, USA.
ACM 978-1-4503-0719-2/11/06.

2. ALGORITHMS

In a first step we assume the number of information items k to be known up to a constant, i.e., we assume that the algorithm knows a number $\tilde{k} \in \mathbb{N}$ satisfying $\tilde{k}/2 \leq k \leq 2\tilde{k}$. Later we show that this bound is not necessary.

Depending on the value of $\tilde{k}$, different strategies are applied to guarantee a timely detection and distribution of information items whp$_n$. More precisely, we devise three algorithms, each suitable for a different range of k: Algorithms $\mathcal{A}_{tiny}$, $\mathcal{A}_{small}$ and $\mathcal{A}_{tree}$. All algorithms run in time $O(\tilde{k})$ and run correctly whp$_n$. The constant β influences the success probability.

Theorem 2.1. *For $\tilde{k} < \sqrt{\log n}$, Algorithm $\mathcal{A}_{\text{tiny}}$ distributes all information items in $\Theta(\tilde{k})$ time slots whp$_n$, for $\sqrt{\log n} \leq \tilde{k} < \frac{\log n - 3}{\beta}$, Algorithm $\mathcal{A}_{\text{small}}$ achieves the same (constant β defined later). The deterministic Algorithm $\mathcal{A}_{\text{tree}}$ completes in $\Theta(\tilde{k} + \log n)$ time slots.*

The above algorithms can be combined to solve the selection problem for unknown k even without needing given bounds on k like $\tilde{k}/2 \leq k \leq 2\tilde{k}$. For this task we can extend the randomized algorithms to detect if k is too large, i.e., if $k > 2\tilde{k}$ and the algorithm is not able to work correctly whp$_n$ (see full version). With these extensions we can construct Algorithm $\mathcal{A}$ for unknown k. We start with estimating $\tilde{k}$ to be $\tilde{k} = 2$ and double $\tilde{k}$ until the testing procedure of the algorithms presented confirms that $\tilde{k}$ is in the right order of magnitude.

Theorem 2.2. *Algorithm $\mathcal{A}$ completes in $\Theta(k)$ time slots after which all information items have been detected and distributed whp$_n$ even if k is unknown and no bounds on k are given.*

The number of channels our randomized algorithms need is large in order to guarantee high success probability ($\Theta(\sqrt{n})$ channels). The deterministic algorithm presented requires even more channels for a timely distribution. Such large numbers of channels are rarely available in practice. Thus we mainly view our work as a first step to generalizing the information exchange problem to multiple channels and as a proof that time-optimal distribution is possible. Reducing the number of channels necessary is left as an open problem for future research.

2.1 Deterministic Tree Dissemination $\mathcal{A}_{tree}$

We can use a balanced binary tree of depth $\log n$ and n channels (one for each node) to disseminate information deterministically in time $O(\tilde{k} + \log n)$. The IDs of the nodes determine their position in the tree as well as the channel assignment and a schedule specifying when each node transmits its message on its own channel or listens to its children or parent nodes on their channel. Once the root of the tree has obtained all information items, it broadcasts them on channel 1 and all other nodes listen. $\mathcal{A}_{tree}$ distributes all information items within $O(\tilde{k} + \log n)$ time slots without collisions.

2.2 $\mathcal{A}_{tiny}$ for $\tilde{k} < \sqrt{\log n}$

Each reporter selects a random channel from a set of $K := n^{1/(2\tilde{k})}$ channels, such that at least half of the reporters choose a unique channel. We call a transmission of a reporter that chooses a unique channel a "successful transmission" since in this case no collision occurs. The number K is selected in such a way that it is small enough to ensure that for each of the $\sum_{i=0}^{\tilde{k}} \binom{K}{i}$ possible subsets of at most $\tilde{k}$ channels with a successful transmission there is a non-reporter node in the network that can be assigned to listen to that subset. Each such listener then listens on all channels from the assigned subset one after another. We argue that there is a exactly one node (called the "boss") that listens exactly on those channels on which the information items were transmitted successfully. Thus this boss collects the information of all successful reporters (at least half of all reporters transmitted successfully) and broadcasts it subsequently. Since the boss is unique it can successfully transmit the gathered information to the network and the number of reporters is cut in half in time $O(\tilde{k})$. Repeating this procedure until no reporters are left takes time $O(\tilde{k})/2^0 + O(\tilde{k})/2^1 + O(\tilde{k})/2^2 + \cdots + O(\tilde{k})/2^{\log O(\tilde{k})} = O(\tilde{k})$.

2.3 $\mathcal{A}_{small}$ for $\sqrt{\log n} \leq \tilde{k} < \frac{\log n - 3}{\beta}$

$\mathcal{A}_{small}$ consists of four steps. In step 1, the nodes determine which role they are going to play during the execution (there are k reporters, $2^{\beta\tilde{k}/2}$ listeners and $n - k - 2^{\beta\tilde{k}/2}$ others). In step 2, each of the k reporters tries to tell a randomly picked listener its information item (a balls-into-bins-style procedure: each listener node listens on a unique channel—these are the bins, each reporter chooses a random channel to send its information item, i.e., it throws ball into a random bin, repeated $\tilde{k}$ times). In step 3, the listeners send all collected information items to the boss with the tree dissemination algorithm $\mathcal{A}_{tree}$. In step 4, the boss broadcasts the collected information items.

We can show that each step completes in $O(\tilde{k})$ time slots and all reporters are successful at least once during step 2 whp$_n$ for k in the range considered. Furthermore, no collisions occur in step 3 and 4 and $\mathcal{A}_{small}$ achieves its goal.

3. REFERENCES

[1] I. Caragiannis, C. Galdi, and C. Kaklamanis. Basic computations in wireless networks. In *ISAAC 05*, volume 3827, page 533, 2005.

[2] B.S. Chlebus, D.R. Kowalski, and T. Radzik. Many-to-Many Communication in Radio Networks. *Algorithmica*, 2009.

[3] S. Gilbert and D. Kowalski. Trusted Computing for Fault-Prone Wireless Networks. *Distributed Computing*, pages 359–373, 2010.

[4] D.R. Kowalski. On selection problem in radio networks. In *PODC*, 2005.