J. Parallel Distrib. Comput. 74 (2014) 1789–1801

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

Efficient fault-tolerant collision-free data aggregation scheduling for wireless sensor networks☆,☆☆

Arshad Jhumka*, Matthew Bradbury, Sain Saginbekov

Department of Computer Science, University of Warwick, Coventry CV4 7AL, UK

HIGHLIGHTS

- We formalise the problem of data aggregation scheduling and prove some impossibility results.
- We develop an efficient modular algorithm that solves stabilising data aggregation scheduling in the presence of crash failures.
- We show, through simulation and an actual deployment, the viability of our approach.

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form
22 July 2013
Accepted 24 September 2013
Available online 11 October 2013

Keywords:
Wireless sensor networks
Data aggregation scheduling
Fault tolerance
Crashes
Collision freedom
Impossibility
Correctness

ABSTRACT

This paper investigates the design of fault-tolerant TDMA-based data aggregation scheduling (DAS) protocols for wireless sensor networks (WSNs). DAS is a fundamental pattern of communication in wireless sensor networks where sensor nodes aggregate and relay data to a sink node. However, any such DAS protocol needs to be cognisant of the fact that crash failures can occur. We make the following contributions: (i) we identify a necessary condition to solve the DAS problem, (ii) we introduce a strong and weak version of the DAS problem, (iii) we show several impossibility results due to the crash failures, (iv) we develop a modular local algorithm that solves *stabilising weak DAS* and (v) we show, through simulations and an actual deployment on a small testbed, how specific instantiations of parameters can lead to the algorithm achieving very efficient stabilisation.

1. Introduction

Data gathering is a basic capability expected of any wireless sensor network (WSN). The usual way of performing data gathering is to have nodes send their measurements (possibly over multiple hops) to a particular node called a *sink*. This type of communication, called *convergecast*, is fundamental to WSNs. Convergecast generally works by constructing a logical tree (called a convergecast tree) on top of the physical topology, with the sink located at the root, and data is then routed to the sink along the tree. However, to save energy, the data is typically aggregated along the route at specific nodes. This means that an aggregator node needs to have all the values from its children before aggregating the data. However, due to the broadcast nature of the communication medium, data transmissions need to be mediated among the children (and possibly other nodes) to avoid message collisions and interference, events that typically lead to energy exhaustion and which could also bias the aggregation.

Mediating these transmissions can be achieved through the use of an appropriate media access control (MAC) protocol. For WSN applications that need fast response times (e.g., disaster recovery), timeliness is of utmost importance. To this end, we investigate *Time Division Multiple Access (TDMA)-based MAC protocols for data aggregation scheduling*, in which each node is allocated a specific time slot in which it can transmit its data. Another advantage of a TDMA schedule for WSNs is that the transceivers can be turned on only when needed, thus saving energy. There exist several algorithms for convergecast in multi-hop radio networks, e.g., [18,13,19] that can be used for WSNs. A common pattern in TDMA-based convergecast algorithms is the decomposition of

☆ This work was supported by a grant from the University of Warwick.

☆☆ This is an extended version of a paper [15] that was published in the Proceedings of the Symposium on Reliable Distributed Systems (SRDS), 2010.

* Corresponding author.
E-mail addresses: arshad@dcs.warwick.ac.uk, hajhumka@gmail.com (A. Jhumka), M.Bradbury@warwick.ac.uk (M. Bradbury), sain@dcs.warwick.ac.uk (S. Saginbekov).

0743-7315/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jpdc.2013.09.011

the problem into two independent subproblems: (i) a logical tree construction, and (ii) time slot allocation along the constructed tree. For example, the tree in [18] is constructed based on the positions of the nodes in a 2-D plane. Various objectives of convergecast scheduling algorithms exist, e.g., minimising time for completing a convergecast [13], and maximising throughput [19], which determine the slot assignment (i.e., the schedule) along the tree.

When slots are assigned to nodes for data aggregation, in what we term as *data aggregation scheduling* (DAS), a node will first aggregate the data obtained from its children before relaying it to its parent. The objective, in this case, is that a node can only transmit a message *after* collecting data from *all* of its children. However, whenever a node crashes in a WSN (e.g., due to energy depletion), the values from a whole subtree disappear. Thus, it is important for a DAS algorithm to be fault-tolerant to ensure that correct nodes have a proper path to the sink, in the sense that their parent transmits the aggregated data after their own transmission.

1.1. Contributions

Several works have addressed the problem of convergecast, with a subset of these addressing the problem of data aggregation scheduling. However, to the best of our knowledge, no work has investigated the problem of *DAS in the presence of crash failures*, on which we focus in this paper. Crash failures in WSNs can be brought about by, for example, defective hardware or battery exhaustion. In this context, we make an in-depth study of DAS in the presence of crash failures. We make a number of contributions in three categories:

- Theory
  1. We identify a necessary condition for solving the data aggregation scheduling problem. This condition provides the theoretical basis that explains the structure of several data aggregation scheduling.
  2. We provide two variants of the DAS problem, namely (i) strong data aggregation scheduling, and (ii) weak data aggregation scheduling.
  3. We show that it is impossible to solve the strong data aggregation scheduling problem.
  4. We show that it is impossible to solve the weak data aggregation scheduling problem in the presence of crash failures.
  5. We introduce the problem of stabilising weak data aggregation scheduling and show that, in general, there is no 2-local algorithm that solves the problem.
- Algorithm
  1. We develop a modular d-local algorithm that solves weak DAS and achieves efficient stabilisation, where d is the diameter of the affected area.
- Results and validation
  1. Using both simulation and an actual deployment on a small testbed, we show that, under appropriate parameterisation, the d-local algorithm can achieve 2-local stabilisation.

Our paper is structured as follows. We present our system and fault models in Section 2. We formalise the problems of strong and weak data aggregation scheduling in Section 3. In Section 4, we focus on variants of the weak data aggregation convergecast. In Section 5, we present and prove the various impossibility results. In Section 6, we provide a d-local algorithm that achieves efficient stabilisation. We present the performance of our algorithm in Section 7. In Section 8, we survey related work in the area, and put our work in the proper context. We discuss the impact of the results in Section 9. We finally summarise the paper in Section 10.

2. Models: system and faults

2.1. Graphs and networks

A wireless sensor node is a computing device that is equipped with a wireless interface and is associated with a unique identifier. A wireless sensor network (WSN) consists of a set of wireless sensor nodes that communicate among themselves via their wireless interface. Communication in wireless networks is typically modelled with a circular communication range centred on a node. With this model, a node is thought as able to exchange data with all devices within its communication range.

A wireless sensor network is then typically modelled as an undirected graph $G = (V, E)$ where V is a set of Γ wireless sensor nodes and E is a set of edges or links, each link being a pair of distinct nodes. Two nodes $m, n \in V$ are said to be 1-hop neighbours (or neighbours) iff $(m, n) \in E$, i.e., m and n are in each other's communication range. We denote by M the set of m's neighbours, and we denote by M^d, the d-hop neighbourhood of m. We say that two nodes m and n can collide at node p if $(p \in M) \wedge (p \in N)$[1]. In general, two nodes m and n can collide if they are in the 2-hop neighbourhood of each other. We then define the collision group of a node n as follows:

$$CG(n) = \{m \in V | ((n, m) \in E) \vee (2hopN(m, n))\},$$

where $2hopN(m, n)$ is a predicate that returns true if m, n are in each other's 2-hop neighbourhood.

We denote by Δ_m the degree of node m, i.e., the size of M. We also denote by Δ_G, the degree of G, i.e., $\Delta_G = \max(\{\Delta_m, m \in V\})$. We also denote by η_G, the maximum of nodes at any hop distance in G. We assume a distinguish node $S \in V$, called a *sink*. A path of length k is a sequence of nodes $n_k \ldots n_0$ such that $\forall j, 0 < j \leq k, n_j$ and n_{j-1} are neighbours. We say a path $n_k \ldots n_0$ is an S-path if $n_0 = S$. The path $n_k \ldots n_0$ is said to be *forward* if $\forall i, j, 0 < i \leq j \leq k, n_i \neq n_j$. A path $n_k \ldots n_0$ is called a *cycle* if the path is forward and $n_0 = n_k$. In this paper, we focus on forward S-paths (henceforth, paths). Specifically, we are only interested in paths from a node to the sink, hence forward S-paths. For an S-path $n_k \ldots S$, we say that n_k has an S-path.

Given an undirected graph $G = (V, E)$: G is connected iff there exists a path in G between each pair of distinct nodes. In general, we are only interested in paths that end with the sink. We say that G is S-connected iff every node in G has an S-path, and we say that G is S^k-connected iff G is S-connected, and all nodes have k node-disjoint S-paths. Two paths are node-disjoint only if the end nodes of the two paths are the same while all other nodes differ. In this paper, when we mention disjoint paths, we mean node-disjoint paths. The distance between two nodes m and n in G is the length of the smallest path between m and n in G. We denote the distance between m and n by $d(m, n)$. The diameter D_G of G is equal to $\max(\{d(m, n), m \in V \wedge n \in V\})$.

2.2. Distributed programs

2.2.1. Syntax

We model the processing on a WSN node as a process containing non-empty sets of variables and actions. A distributed program is thus a finite set of N communicating processes. We represent the communication network topology of a distributed program by an undirected connected graph $G = (V, E)$, where V is the set of N processes and E is a set of edges such that $\forall m, n \in V, (m, n) \in E$ iff m and n can directly communicate together, i.e., nodes m and n are neighbours.

[1] We will say two nodes m and n can collide if such a node p exists.