The Journal of Systems and Software 83 (2010) 108–120

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

McTorrent: Using multiple communication channels for efficient bulk data dissemination in wireless sensor networks

Leijun Huang, Sanjeev Setia, Robert Simon *

Department of Computer Science, George Mason University, Fairfax, VA 22030, United States

ARTICLE INFO

Article history:
Received 2 January 2008
Received in revised form 9 July 2009
Accepted 10 July 2009
Available online 26 July 2009

Keywords:
Wireless sensor networks
Multi-channel sensor networking

ABSTRACT

This paper presents McTorrent, a reliable bulk data dissemination protocol for sensor networks. The protocol is designed to take advantage of multiple radio channels to reduce packet collisions and improve the latency of large object dissemination. We evaluated the performance of McTorrent via detailed simulations and experiments based upon an implementation on the TinyOS platform. Our results show that in comparison to Deluge, the de facto network reprogramming protocol for TinyOS, McTorrent significantly reduces the number of packet transmissions and the amount of time required to propagate a large data object through a sensor network.

1. Introduction

A wireless sensor network consists of a large number of inexpensive and resource constrained nodes that are self-organized into a multi-hop wireless network. Each node consists of a CPU, memory modules, power supply, sensors for environmental monitoring and a radio. A typical distributed sensor system is tasked to perform a specific application. Example sensor network applications include environmental monitoring, perimeter surveillance, battlefield operations and product inventory maintenance.

In many situations, sensor networks need to run for a long time once deployed. When the sensed environment or application requirements change then updating each node's code image (the software that they are running) becomes necessary. Many deployed systems may have a large number of nodes. Further, there are cases when the physical circumstances of the actual deployment makes physically collecting previously deployed nodes either very difficult or infeasible. In these situations the code image needs to be updated "over-the-air" using the existing network itself. The critical service required for this purpose is a *reliable bulk data dissemination* protocol, i.e., a protocol for reliably distributing a large object to *all the nodes in the network*.

The reliable bulk data dissemination problem is a special case of the reliable downstream data transport problem for sensor networks. The main difference between bulk data dissemination and packet-level data dissemination is that the size of the object being distributed is two to three orders of magnitude larger than the size of a data packet, and is also much larger than the amount of RAM available on a typical sensor node. For example, the code image for a current generation sensor node such as a TelosB mote (Data Sheets, xxxx) could be as large as 48 KB, whereas the default size of packets transmitted by a TelosB is 36 bytes, and the amount of RAM available on a TelosB is 10 KB. Another challenge in reliable bulk data dissemination is that the object being disseminated has to be delivered to all the nodes in the network in as small a time as possible in order to minimize the service interruption to the deployed application. Moreover, the energy expense of the dissemination should be minimized in order to minimize the impact on the lifetime of the sensor nodes.

Several protocols have been designed for reliable bulk data dissemination in sensor networks, such as MOAP (Stathopoulos et al., 2003), Deluge (Hui and Culler, 2004) and MNP (Kulkarni and Wang, 2005). Most existing protocols use a single channel for all packet transmissions, while current generation sensor nodes are usually equipped with multiple-channel radios (for instance, the Crossbow TelosB mote research platform comes equipped with a Chipcon CC2420 radio that has 16 IEEE 802.15.4 compliant channels (Data Sheets, xxxx)). With a single channel, nodes have to compete with all of the other neighboring nodes for the channel before starting transmissions, otherwise collisions may occur if two or more of them are sending simultaneously. Moreover, hidden terminal problems occur where two nodes that are two hops away send packets at the same time, causing collisions at intermediate nodes.

With multiple channels enabled, nodes in a neighborhood can send packets simultaneously without causing collisions, as long as they use different frequencies and the gap between any two frequencies is large enough so that the interference between frequencies is negligible. Moreover, since more transmissions can proceed

* Corresponding author.
E-mail addresses: lhuang2@cs.gmu.edu (L. Huang), setia@cs.gmu.edu (S. Setia), simon@cs.gmu.edu (R. Simon).

0164-1212/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2009.07.036

concurrently by using multiple channels, the overall throughput is increased, leading to a reduced latency of data dissemination.

On the other hand, using multiple channels also requires more complicated coordination between nodes due to the fact that many currently available radios for sensor nodes are equipped with one half-duplex transceiver. Although the transceiver can be programmed on the fly to switch channels, each node can only transmit or listen on one channel at a time. Thus, it is necessary for senders and receivers to coordinate both which channel they are assigned to *and* when they send and receive packets.

In this paper, we present the design, implementation and analysis of *McTorrent*, a multi-hop reliable data dissemination protocol using multiple radio channels. The McTorrent protocol can be used for distributing a large data object to all the nodes in the network, e.g., distributing a new code image to sensor nodes for network reprogramming. It uses a contention-based approach to channel access, and does not require time synchronization. As compared to a single channel approach, McTorrent significantly reduces the total latency for reliably disseminating a large data object from a sink and the total number of packet transmissions required for the dissemination.

The paper proceeds as follows. Section 2 presents related work in data dissemination protocols and multi-channel wireless systems. Section 3 describes the McTorrent protocol. Section 4 describes the software architecture of the implementation of McTorrent. Section 5 evaluates the performance of McTorrent. Finally, Section 6 offers some conclusions and observations.

2. Related work

This section describes related work on data dissemination in sensor networks. We also discuss work on the use of multi-channel wireless networking.

2.1. Data dissemination

Protocols for data dissemination in sensor networks can be divided into two categories based on the size of the data objects that are being disseminated. The first category of protocols is designed to exchange a relatively small amount of data among neighboring nodes, e.g., for synchronizing their configuration information, whereas the second class of protocols is intended for disseminating very large data objects, e.g., entire code images.

Protocols in the first category include the SPIN family of data dissemination protocols (Kulik et al., 2002). The SPIN protocols take advantage of the broadcast nature of the wireless medium, and use various communication suppression mechanisms to reduce latency. SPIN uses a three-phase (advertisement-request-data) handshaking protocol between nodes to disseminate data. The Trickle (Levis et al., 2004) protocol proposes mechanisms for dynamic broadcast rate adjustment with the goal of quickly propagating updates while reducing communication between nodes when there are no updates.

The second category includes several protocols such as MOAP (Stathopoulos et al., 2003), Deluge (Hui and Culler, 2004), and MNP (Kulkarni and Wang, 2005), that have been proposed for multi-hop network reprogramming in a sensor network. Network reprogramming is an important application of reliable data dissemination protocols for sensor networks. All these protocols share some basic characteristics. First, the protocols are used for entire code image delivery as opposed to difference-based application adjustment (Reijers and Langendoen, 2003). Second, these protocols extend the three-phase handshaking protocol proposed in SPIN-BL (Kulik et al., 2002) for handling large data objects. Third, the protocols all borrow ideas such as the use of selective NACKs and hop-by-hop error recovery from prior work in reliable data transfer protocols (Wan et al., 2002; Stann and Heidemann, 2003). Deluge and MNP differ from MOAP in that a node does not need to receive the entire code image before it can rebroadcast it. By breaking the code image up into pages, and allowing pipelined page delivery, Deluge and MNP take advantage of spatial multiplexing to reduce the latency of network reprogramming. Deluge leverages the Trickle protocol (Levis et al., 2004) to limit transmissions between neighboring nodes. MNP differs from Deluge in its approach for sender selection and in allowing radios to be turned off during reprogramming. Finally, Starobinski et al. present some basic performance limitations of multi-channel data dissemination protocols for certain classes of network topologies (Starobinski et al., 2007).

Our protocol for large object dissemination, McTorrent, resembles Deluge and MNP in its design with the crucial difference that McTorrent exploits multiple communication channels for data dissemination. This paper is based on the work presented in (Simon et al., 2005). Two multiple channel protocols were described in (Simon et al., 2005), one for multi-hop data dissemination and the other for data synchronization within a local cluster of nodes. The present work expands upon the McTorrent protocol, adding design details and large scale performance studies.

2.2. Reliable multicast protocols for MANETs

We note that there is a large body of work on reliable multicast protocols for mobile ad hoc networks (MANETs). Reliable multicast protocols for MANETs are usually ARQ-based, gossip-based or FEC-based (Ouyang et al., 2005). ARQ-based protocols (Gopalsamy et al., 2002; Tang et al., 2002) use ACKs (for sender-initiated reliability) or NACKs (for receiver-initiated reliability) from receivers and retransmissions from senders to achieve reliability. In gossip-based protocols (Chandra et al., 2001; Luo et al., 2003), packets are transmitted multiple times hop-by-hop so that the receivers can receive the packets with a high probability. FEC-based protocols (Rizzo and Vicisano, 1998) use error correction techniques to help achieve transmission reliability.

While MANETs have similar characteristics as sensor networks such as wireless media and mobility, sensor networks are different in that sensor nodes have more limited computing capability, memory capacity and power budget, and the wireless links in sensor networks usually have higher loss rate. These constraints require that protocols for sensor networks should be more energy-efficient. However, the main difference between reliable multicast protocols for MANETs and protocols such as Deluge and McTorrent is that the latter protocols are designed for reliable bulk data dissemination as opposed to the reliable delivery on a per-packet basis.

2.3. Multi-channel wireless systems and sensor networks

There are many papers that have dealt with the benefits of using multiple channels as a means to achieve higher throughput, often by also dealing with hidden terminal problems. One way of characterizing multi-channel wireless networking research is by the number of transceivers and the assumed data link protocol. Earlier works focus on CSMA-based 802.11 like networks include (Nasipuri and Das, 1999, 2000). Research presented in (Wu et al., 2000; Jain et al., 2001) use dynamic channel assignment strategies for multiple transceivers, also for 802.11. We also note that a commonly used approach for dynamic channel assignment is to use a coloring algorithm (Mishra et al., 2005). McTorrent uses a simple and effective approach for channel assignment, based on randomization and careful timer management. We have found this