Ad Hoc Networks 54 (2017) 85–98

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Enabling sustainable bulk transfer in environmentally-powered wireless sensor networks

Alvin C. Valera[a,b,*], Wee-Seng Soh[b], Hwee-Pink Tan[a]

[a] School of Information Systems, Singapore Management University, 178902, Singapore
[b] Department of Electrical and Computer Engineering, National University of Singapore, 117583, Singapore

ARTICLE INFO

Article history:
Received 29 March 2016
Revised 6 October 2016
Accepted 7 October 2016
Available online 8 October 2016

Keywords:
Bulk transfer
Energy-harvesting
Adaptive control
Dynamic duty cycling
Sensor network

ABSTRACT

We address the problem of transferring bulk data in environmentally-powered wireless sensor networks where duty cycle compliance is critical for their uninterrupted operation. We propose Pump-and-Nap, a packet train forwarding technique that maximizes throughput while simultaneously enforcing compliance to dynamic duty cycle limitations. A node using Pump-and-Nap operates by *pumping* a train of packets followed by a *napping* period where the node forgoes any transmission. Pump-and-Nap employs an *adaptive controller* to periodically compute the *optimal capacity*, that is, the maximum number of packets a node can receive and transmit in a train, given its duty cycle constraint. The controller uses prior input-output observations (capacity allocations and their corresponding duty cycle usage) to continuously tune its performance and adapt to wireless link quality variations. Its use of local information makes the controller easily deployable in a distributed fashion. We implemented Pump-and-Nap in TinyOS and evaluated its performance through experiments and testbed simulations. Results show that Pump-and-Nap provides high transfer throughput while it simultaneously tracks the target duty cycle. More importantly, Pump-and-Nap enables sustainable bulk transfer compared to state-of-the-art techniques that greedily maximize throughput at the expense of downtime due to energy depletion.

1. Introduction

Wireless sensor networks are becoming ubiquitous because of their diverse applications in areas such as agriculture, environmental monitoring, industrial and home automation, military, and structural health monitoring, to name a few [1]. A critical issue that plagues many deployments, however, is the *limited lifetime problem* due to the finite battery capacity of sensor nodes [2]. Fortunately, advances in energy harvesting and storage technologies are enabling the deployment of *environmentally-powered* wireless sensor networks (EPWSN), wherein the sensor nodes harvest energy from the environment to recharge their batteries or energy stores. Recently, the use of supercapacitors as primary energy store is becoming popular because of their significantly higher number of recharge cycles compared to batteries [3]. Some example nodes that solely rely on supercapacitors are Everlast, SolarBiscuit, and Sunflower [2].

In many applications (*e.g.*, [4,5]), sensor nodes are tasked to record time-series data at high sampling rates, resulting in large or bulk sensor data. These bulk data, typically in the order of tens to hundreds of kilobytes, need to be transferred in real-time to a gateway for eventual transmission to the backend, where further processing and analysis can be undertaken. Due to storage limitations in sensor nodes, bulk data must be immediately transferred to avoid overflow and data loss. Bulk transfer in EPWSNs is challenging because the nodes perform adaptive duty cycling to ensure uninterrupted operation [6–8]. This is especially imperative in deployments wherein the sensor nodes rely solely on low capacity energy stores such as supercapacitors [2]. Such nodes must strictly operate according to a specified duty cycle, or risk downtime due to short-term energy shortage.

In this work, we tackle the problem of bulk data transfer in EPWSNs where adherence to duty cycle constraints is a primary concern. While several bulk transfer schemes have been proposed [9–14], they focus mainly on maximizing the throughput, neglecting the duty cycle constraints of sensor nodes. The use of existing schemes may therefore cause uncontrolled and rapid draining of the energy reserves, leading to the temporary unavailability of nodes along the transfer path. Ultimately, this will result in transfer disruptions which render the transfer of arbitrarily-sized sensor data difficult, if not infeasible.

* Corresponding author.
E-mail addresses: alvinvalera@smu.edu.sg (A.C. Valera), elesohws@nus.edu.sg (W.-S. Soh), hptan@smu.edu.sg (H.-P. Tan).

http://dx.doi.org/10.1016/j.adhoc.2016.10.008
1570-8705/

Recently, the use of *packet bursting* or *packet trains* in conjunction with radio duty cycling [12] have been proposed to attain low power, high transfer throughput. While the technique yields low energy consumption, the outcome is *incidental* rather than *intentional, i.e.*, the use of packet trains does not actively control the energy usage to be within specified bounds. We therefore introduce Pump-and-Nap, a forwarding technique that uses *controlled packet trains* to simultaneously maximize throughput and enforce compliance to dynamic duty cycle limitations. At the heart of Pump-and-Nap is an *adaptive controller* that determines a node's *optimal capacity*, defined as the maximum number of packets the node can receive and transmit in a train within its duty cycle constraints. The controller uses prior input-output observations (capacity allocations and their corresponding duty cycle usage) to continuously tune its performance and adapt to wireless link quality variations.

We implement Pump-and-Nap in TinyOS [15] and perform experiments in the Indriya testbed [16], a 139-node indoor testbed, to evaluate its performance. Experimental results show that Pump-and-Nap can adaptively track duty cycles and provide high bulk transfer throughput at the same time. More importantly, we demonstrate in energy harvesting experiments and testbed simulations that Pump-and-Nap can truly enable sustainable bulk transfer compared to state-of-the-art techniques [9,12] that greedily maximize throughput at the expense of downtime due to energy depletion.

The rest of the paper is organized as follows. In Section 2, we review the state-of-the-art duty cycling and bulk transfer in sensor networks and identify the challenges in the context of EPWSN. In Section 3, we describe Pump-and-Nap in detail while in Section 4, we present its implementation, along with the experiments designed to evaluate and compare its performance. In Section 5, we present and discuss the experimental results. We conclude the paper in Section 6 and state several possible future work.

2. State-of-the-art and challenges

To understand how bulk transfer protocols will perform in EPWSN, we survey the state-of-the-art in duty cycling and bulk transfer. The ultimate aim of this section is to expose the shortcomings of existing bulk transfer schemes when duty cycle compliance is of paramount importance.

2.1. Duty cycling MAC protocols

A duty-cycling node may employ any of the state-of-the-art duty cycling medium access control (MAC) protocols to control the sleep and wakeup of its radio. Duty cycling MAC protocols can be either *synchronous* [17–19] or *asynchronous* [20–23]. In the former, the nodes sleep and wakeup at the same time while in the latter, the nodes may sleep and wakeup at different times. In this work, we motivate our design using asynchronous schemes because they offer two distinct advantages over synchronous schemes: (*i*) they do not require periodic re-synchronization which can entail significant energy consumption [24]; and (*ii*) they do not require the storage and exchange of wakeup schedules which can entail significant memory and communication overhead [25]. Nevertheless, our resulting scheme can also be used on top of synchronous MAC protocols after slight modifications.

In asynchronous schemes, a packet transmission is preceded either by a *beacon listening phase* or a *preamble(s) transmission phase*[1]. The former is employed in *receiver-initiated* schemes (*e.g.*, [22]) while the latter is used in *transmitter-initiated* schemes (*e.g.*, [20,21,23]). Regardless, the transmitting node always incurs this overhead before it can have the opportunity to transmit its packets. For simplicity, we introduce a common term to refer to either overhead:

Definition 1 (Pre-transmission overhead)**.** The duration from the moment a transmitting node v has a packet ready for transmission until the time the receiving node w wakes up. During this time, v's radio is active, either awaiting for a beacon (receiver-initiated) or transmitting preamble(s) (transmitter-initiated).

2.2. Bulk transfer

Bulk transfer refers to the transmission of large amount of sensor data from a source node to a destination node, typically a gateway or a base station. Bulk transfer can actually be performed using *generic* transport protocols (Wang et al. [26] provides a good survey on this subject) but specific application requirements and tight resource constraints in terms of memory, channel capacity and energy have led to the development of *specialized* protocols for bulk transfers.

Koala [27] is one of the earliest schemes for bulk transfer. It uses round-trip time (RTT) to control the sending rate from the source to the sink. Specifically, Koala sends packets at a rate of RTT/2, relying on its underlying flexible control protocol to provide reliability. Koala supports duty cycling and uses low-power probing, a technique akin to beacon transmission in receiver-initiated MAC protocols.

Unfortunately, RTT-based rate control performs poorly over long paths. As such, newer schemes such as Flush [9] and PIP [10] introduced the idea of "pipelining" packets to improve throughput. Flush [9] proposed a method to probe the interference range of a path and uses two simple rules to maximize the sending rate of a node: (*i*) transmit when the successor node is free from interference, and (*ii*) transmit at rate below the successor node's sending rate. PIP [10] took the idea of packet pipelining further through the use of a MAC protocol that is TDMA-based, centralized, connection-oriented and uses multiple channels. PIP essentially aims to tightly coordinate the packet pipelining from the source to the sink and further reduce intra-flow and inter-flow interference. The former occurs when transmissions of different nodes from the same flow interfere with each other, while the latter occurs in the case of transmissions from different nodes belonging to different concurrent flows.

Flush and PIP are designed to maximize throughput without regard to the energy consumption of the sensor nodes. To achieve the desired packet pipelining effect, they need the radio to be turned on for the entire transfer duration. To remedy this problem, Duquennoy et al. [12] proposed the use of *packet bursting,i.e.*, rapid transmission of successive packets after a single wakeup, in conjunction with duty cycling. Results show that packet bursting in conjunction with duty cycling in ContikiMAC [23] can provide low power and high throughput performance. We note however that although [12] can provide low power consumption, it is *incidental* rather than *intentional, i.e.*, it does not actively control consumption to be within specified bounds.

2.3. Bulk transfer in EPWSN

We can group the bulk transfer schemes that we have presented previously into two categories, namely, *single packet-based* and *packet train-based*. Koala, Flush and PIP fall under the first category while the scheme by Duquennoy et al. falls under the latter. In what follows, we identify the issues of using either scheme in the context of EPWSN.

Consider a multi-hop bulk transfer from node s to t. Supposing that we can modify the single packet-based schemes Flush and PIP

[1] In [23], preambles are replaced by actual data packets.