Computer Networks 97 (2016) 113–127

Contents lists available at ScienceDirect

Computer Networks

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

Distributed Data Storage Systems for Data Survivability in Wireless Sensor Networks using Decentralized Erasure Codes

Louai Al-Awami[a,b,*], Hossam S. Hassanein[b]

[a] Department of Computer Engineering, King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia
[b] School of Computing, Queen's University, Kingston, ON, Canada

ARTICLE INFO

Article history:
Received 10 February 2015
Revised 14 January 2016
Accepted 19 January 2016
Available online 26 January 2016

Keywords:
Wireless Sensor Network
Data Survivability
Decentralized Erasure Codes
Network Coding
Distributed Data Storage

ABSTRACT

Achieving reliability in Wireless Sensor Networks (WSNs) is challenging due to the limited resources available. In this study, we investigate the design of data survivability schemes using decentralized storage systems in WSNs. We propose a data storage system design based on Decentralized Erasure Codes (DEC) that features a simple and decentralized construction of the target code. The proposed framework allows sensor nodes to cooperate to build an erasure code-based storage that can tolerate a given failure/erasure rate. Code construction and decoding can both be performed randomly allowing for a distributed operation with no prior setup or coordination between source nodes. Further, we present two approaches that utilize Random Linear Network Coding (RLNC) to enhance the proposed scheme in order to achieve energy efficiency. We present the theoretical basis of the schemes then validate and evaluate their performance through simulations.

1. Introduction

Wireless Sensor Network (WSN) technology is being increasingly deployed in a diverse range of applications. Intelligent Transportation Systems (ITSs) [1], Smart Grids [2], and the Internet of Things (IoT) [3] are just a few examples of technologies where WSNs are used. Generally, WSNs are comprised of **sensor nodes** that are equipped with one or multiple sensors, a processing unit, and a wireless communication module. Sensor nodes cooperate in monitoring a phenomenon of interest and in relaying the sensed data to a **sink node** for processing. When produced in large numbers, sensor nodes can be extremely inexpensive, and hence they can be deployed in greater numbers to build large scale networks. WSNs have stringent constraints, especially regarding power consumption and scalability. Furthermore, reliability becomes a key requirement for WSNs when deployed in unattended applications or under harsh working conditions.

To preserve the sensed data captured by sensor nodes, WSNs nodes can benefit from using Distributed Data Storage Systems (DDSSs) technology. Data storage systems represent an essential component of today's networks and they have been researched for a long time. Lately, data storage technology is being revisited especially in the contexts of Content Centric Networking (CCN) [4] and cloud computing [2]. DDSSs utilize *hardware redundancy* and *data replication* to protect data in case of possible failures. More specifically, given a data packet, a DDSS replicates the packet over multiple physical storage devices, such that when a subset of these devices fails, the data packet can be retrieved from the surviving ones.

In this study, our goal is to design a DDSS that is tailored for WSNs data reliability applications. For that, we first introduce the notion of data survivability as a quantitative parameter that links the amount of redundancy required to the maximum failure that can be tolerated. We

[*] Corresponding author at: School of Computing, Queen's University, Kingston, ON, Canada. Tel.: +1 6135336336.
E-mail addresses: louai@kfupm.edu.sa, louai@cs.queensu.ca (L. Al-Awami), hossam@cs.queensu.ca (H.S. Hassanein).

http://dx.doi.org/10.1016/j.comnet.2016.01.008
1389-1286/© 2016 Elsevier B.V. All rights reserved.

then show how data survivability can be useful by implementing a data survivability scheme, called Decentralized 30 Erasure Codes for Data Survivability (DEC-DS). DEC-DS is based on Decentralized Erasure Codes (DEC) [5–7]. Besides being decentralized, DEC has a predictable algebraic structure allowing for quantifiable performance. After that, we present two methods to enhance the energy efficiency of DEC-DS by exploiting Network Coding (NC). The two schemes are referred to as DEC Encode-and-Forward (DEC-EaF) and DEC Encode-and-Disseminate (DEC-EaD). NC [8] has emerged as an information-theoretic tool and has been shown to decrease energy consumption and complexity while increasing throughput and reliability [9]. Random Linear Network Coding (RLNC) [10] has been later proposed as a practical implementation of Network Coding. In this study, we utilize RLNC to increase the efficiency of the proposed storage system by reducing communication overhead and consequently energy requirements. The main contributions of this paper are introducing the notion of data survivability and presenting the three data storage schemes, DEC-DS, DEC-EaF, and DEC-EaD.

The remainder of the paper is organized as follows. In Section 2, we present some background material and review related work. The proposed data survivability framework is discussed in Section 3. Section 4 shows two schemes using RLNC to improve the efficiency of the proposed data survivability application. Experiments and results are discussed in Section 5. Finally, Section 6 concludes the paper. Some important results from the theory of random matrices over finite fields, which will be used in designing the codes, are presented in Appendix A.

2. Background and related work

Before we discuss the proposed schemes, we present the advantages and disadvantages of replication and encoding-based storage. We then present the concept of data survivability and how it differs from network survivability. We also present an overview of Fountain Codes and DEC; and survey related literature on DDSSs in WSNs.

2.1. Replication Vs. encoding

Replicated data can be stored either as is (*replication-based storage*) or encoded using erasure codes (*coding-based storage*). Coding-based solutions can achieve many advantages over replication-based solutions at a slight increase in processing cost. Unlike coding, replication often requires more storage space on every storage node. In other words, to attain the same level of reliability, replication-based schemes require more redundancy than coding-based schemes. In fact, for the same level of redundancy, coding can achieve an order of magnitude higher reliability than replication [11]. In addition, replication-based approaches also need to keep track of where each data exist, resulting in complicated data gathering protocols. Moreover, it has been shown analytically that on average the number of data blocks needed to reconstruct a complete data set from a replication-based distributed storage is more than what is needed when using coding-based distributed storage [12].

2.2. Data Survivability vs. Network Survivability

As aforementioned, WSNs combine a set of unique requirements such as limited energy, dense deployment, and harsh working conditions. Consequently, developing a DDSS for WSNs needs to tackle such requirements. To address data reliability, sensor data in WSNs need to be maintained using a reliability mechanism. This is especially important when a sink node is not available, such as in the case of Delay Tolerant Networks (DTNs). In this regard, we present data survivability as a design parameter that describes the required data resilience against failures. We make a distinction between data and network survivability. *Network survivability* [13] focuses on using redundancy as a means to guarantee network continuity in case of nodes failure. *Data survivability* provides a means to prevent loss of data in the network in case of failure through the use of redundancy. Also, while network survivability requires redundancy in hardware and software, data survivability utilizes redundancy in storage and data. Other similar concepts exist in the literature such as "*service survivability*" which focuses on continuity of the service even when the physical system fails, through using backup servers [14].

2.3. Fountain Codes

There exists some resemblance between Decentralized Erasure Codes (DEC) and Fountain Codes. Therefore, we provide a brief description of Fountain Codes to lay the ground for the discussion on DEC. The literature on DDSSs contains some overlap between the two codes. We believe it is useful to discuss the two families and show why DEC is better suited for data survivability.

Since their introduction in late 1990's, Fountain codes [15] have attracted an increasing interest in the research community. The main attracting attribute of this family of codes is that they are *rateless*, meaning they do not have a fixed rate associated with them a priori. Hence, compared to ordinary erasure codes such as Reed–Solomon Codes [16], rateless codes can adapt to any given erasure channel with an associated erasure probability p_e on-the-fly.

Given a set of k *native* data blocks of equal length $B = \{b_1, b_2, \ldots, b_k\}$ and a probability distribution $\rho(k)$, the encoder of a Fountain code generates n encoded packets as follows. To generate the ith encoded packet, the encoder samples $\rho(k)$ for a value $1 \le d_i \le k$. Then, it uniformly selects d_i random data blocks from B and *xor's* the blocks linearly together under the mathematics of $\mathbb{F}_2$ generating an *encoded* block e_i. d_i is referred to as the *degree* of the encoded block e_i. Similarly, $\rho(k)$ is called the code *degree distribution*. In addition to the encoded block, a k-dimensional binary *encoding* vector $G_i = \{g_{i1}, g_{i2}, \ldots, g_{ik}\}$ is appended to e_i; where every entry g_{ij} is set to 1 if b_j was used to construct e_i and 0 otherwise. g_{ij} is referred to as an *encoding coefficient*. Let $E = \{e_1, e_2, \ldots, e_n\}$ and $G = \{G_1, G_2, \ldots, G_n\}$ be the set of encoded blocks and encoding vectors, respectively. In general, $k < n$. The decoder on the receiving side, keeps receiving encoded blocks until solving the system of linear equations $E_{1\times n} = B_{1\times k}G_{k\times n}$, for B. The number of packets required for decoding beyond k is referred to as *code overhead*. Generally, the decoder requires $n = (1 + \epsilon)k$