Page Size Optimization for Code Dissemination in Wireless Sensor Networks

Wei Dong¹, Xi-bin Zhao², and Min Xi³

¹ Zhejiang University & HKUST
² Key Laboratory for Information System Security of Ministry of Education, School of Software, Tsinghua University
³ Department of Computer Science, Xi’an Jiaotong University
dongw@zju.edu.cn

Abstract. Wireless sensor networks (WSNs) have recently gained a great deal of research attention, with a wide range of applications being explored. In most applications, WSNs are deployed in inaccessible areas for a long lifetime. Software maintenance and update in WSNs are challenging. Network reprogramming is an important way to address this challenge. Code dissemination is a critical service to enable network reprogramming. Most code dissemination protocols employ segmentation and pipelining to improve the reprogramming efficiency. As we show in this paper, the choice of the page size in these segmented and pipelined dissemination protocols is of vital importance to the overall dissemination time. Hence, we explore the tradeoff in determining the optimal page size in terms of the overall dissemination time. We investigate the impact of page size for two typical code dissemination protocols in WSNs. Results show that the optimal page size decreases when the maximum hop count from the source node increases; and the optimal page size increases when the program size increases. The absolute value of the optimal page size is determined by the network scale, program image size, and protocol details.

1 Introduction

Wireless sensor networks (WSNs) have recently gained a great deal of research attention [12][4][5], with a wide range of applications being explored, such as military surveillance, habitat monitoring, and infrastructure protection, etc. WSN applications need to be updated after deployment for a variety of reasons, such as reconfiguring parameters, modifying tasks of individual nodes, and patching security holes. Many sensor networks, however, are deployed in environments where physically collecting previously deployed nodes is either very difficult or infeasible. Network reprogramming is an important way to address such challenges [7].

The critical service required to enable network reprogramming is a code dissemination protocol. Example protocols include Deluge [8] and MNP [9], which distribute new program binaries into a network, enabling complete system reprogramming.

Segmentation is a common technique that is employed in most existing code dissemination protocols [7]. There are two major benefits of segmentation. First, without segmentation, a large program breaks into thousands of packets. As a consequence,
each node needs a large number of states to record the packet information (received or not) [7]. Second, taking advantage of spatial multiplexing, segmentation and pipelining could increase the overall throughput significantly for a large program [7]. With segmentation and pipelining, a program is divided into several pages, each of which contains a fixed number of packets. Instead of receiving a complete program image before forwarding it, a node becomes a source node after it receives one complete page and uses per page negotiation for serving other nodes.

Previous works on dissemination protocol design [8,9,10] use fixed page sizes. For example, the default page size in Deluge is fixed as 48 packets (48 pkts × 23 bytes/pkt = 1104 bytes). In this work, we explore the tradeoff in determining the optimal page size in terms of the overall dissemination time (and also the dissemination throughput). A large page size (which translates to small number of pages for a given program image) will limit pipelining for spatial multiplexing while a small page size (which translates to large number of pages for a given program image) adds per page negotiation overheads.

To investigate the impact of the page size on the overall dissemination time, we use the analytical model presented in [8]. This model fits well with TOSSIM [11] simulation results. Results show that the optimal page size decreases when the hop count from the source node increases; and the optimal page size increases when the program size increases. The results also indicate that, for disseminating a large program image in large-scale sensor networks, page size optimization is of vital importance because the choice of page size has a great impact on the overall dissemination time.

The rest of this paper is structured as follows. Section 2 gives the background and describes the related work. Section 3 presents the solution for page size optimization for two typical dissemination protocols—Deluge [8] and MNP [9]. Section 4 shows the optimization results. Finally, we conclude this paper in Section 5.

2 Backgrounds and Related Work

This section gives a brief overview of code dissemination in WSNs. We divide prior work into two categories—dissemination protocol design and dissemination performance analysis, which will be described in the following two subsections, respectively.

2.1 Dissemination Protocol Design

Deluge [8] is perhaps the most popular code dissemination protocol used for reliable code updates in WSNs. It uses per page negotiation (i.e., ADV-REQ-DATA three way handshake) for reliability, and employs segmentation (into pages) and pipelining for spatial multiplexing. It is highly optimized and can achieve one ninth the maximum transmission rate of the radio supported under TinyOS [12].

MNP [9] shares many common points with Deluge, e.g., segmentation/pipelining, and per page negotiation. Besides, in order to avoid message collisions and to reduce energy consumption, it provides a detailed sender selection algorithm to choose a local source of the code which can satisfy the maximum number of nodes.

There are many other code dissemination protocols proposed recently, e.g., Stream [13], Rateless Deluge [10]. Segmentation/pipelining and per page negotiation are two