Computer Standards & Interfaces 31 (2009) 579–585

Contents lists available at ScienceDirect

Computer Standards & Interfaces

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

An interface designed for networked monitoring and control in wireless sensor networks

Xianghui Cao, Jiming Chen *, Youxian Sun

State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Available online 10 April 2008

Keywords:
Wireless sensor networks
Interface
Server
Client
Monitoring

ABSTRACT

In this paper, a new interface is developed in detail that provides terminal users with ways for direct and convenient interactions with the wireless sensor networks (WSNs). It also embraces an Internet based subsystem for remote monitoring and controlling of the WSNs. Different from existing platforms and interfaces, our design is general, extensible and not constrained for specific applications. More importantly, it paves the way for improving the data processing efficiency and avoiding data execution congestion. It proposes the idea utilizing the advantages of existing professional mathematical software tools, such as Matlab, Mathematica and Maple, to process the WSN data.

1. Introduction

Wireless sensor networks (WSNs) feature in distributed sensing and automatically running. They can be applied in habit monitoring, environment observation, physiological telemonitoring and even drug administration [9]. WSNs are also data centric, i.e. data acquisition and processing is central and fundamental for most WSN applications. However, in certain research fields sensory data processing becomes more and more difficult and computational costly. For instance, in environmental monitoring, due to large-scaled and dense sensor nodes deployment, long monitoring time or high frequency of data acquisition sometimes, the sensor networks would be burdened with very large amount of data [1,4], and the data processing mechanism would be slowed down as a result. Another instance is that in target tracking application using wireless sensor networks, position calculation and prediction which always introduce advanced algorithms such as digital filtering, may cost too much computational resource in the network data processing center.

There are several ways to alleviate above problem. As in [7,12], data compression is a useful way that only the data within certain bound will be transmitted up to the base station for afterwards processing. Yet useful information of the WSNs may be lost. An alternative is called distributed processing, which lets sensor nodes or distributed low-cost stations analyze local sensory data and only deliver the results to the base station. However, centric data management and processing is required in many cases [4].

In this paper, in order to improve the efficiency of the whole network data processing center, a standard interface is developed which is able to accommodate professional mathematical software environments (such as Matlab, Maple, Mathematica, etc.) into the data processing procedure. Besides, the interface as a whole serves as a bridge with the hardware, the WSNs, to one end and the human users to the other. On the one hand, it communicates with the WSNs and processes and visualizes data in real time. On the other hand, it builds up a bidirectional communication mechanism so that terminal users including remote Clients could control the sensor networks through the Internet.

Fig. 1 gives the framework of the whole interface, which comprises three tiers: the hardware tier, the Server tier and the Client tier. In the first tier, data that generated by the sensors that working under some predefined patterns are framed into packets and emitted from the wireless sensor networks to their base stations and finally forwarded to the Server through the Internet or serial cables. Except for few bad ones, these packets may convey information about certain area of environment (such as temperature, light, humidity, etc. [3]) and will be analyzed afterwards. The Server reads the raw packets and then passes them into some above mentioned professional mathematical software for further calculations. Besides, the Server keeps broadcasting new packets containing data after analysis over the Internet that all the Clients connected to the Server could get them. The powerful Server could also accept terminal users' commands and dispatch them to proper sensors. Note, we refer the terminal users here to consumers of either the Server or the Client.

Although, there are several software systems and platforms dealing with WSN observation and data visualization to date, they are very specific and hard to be extended by users who emphasize much on the data processing efficiency. For example, Mote-View by Crossbow [3]

* Corresponding author.
E-mail address: jmchen@iipc.zju.edu.cn (J. Chen).

0920-5489/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2008.03.026

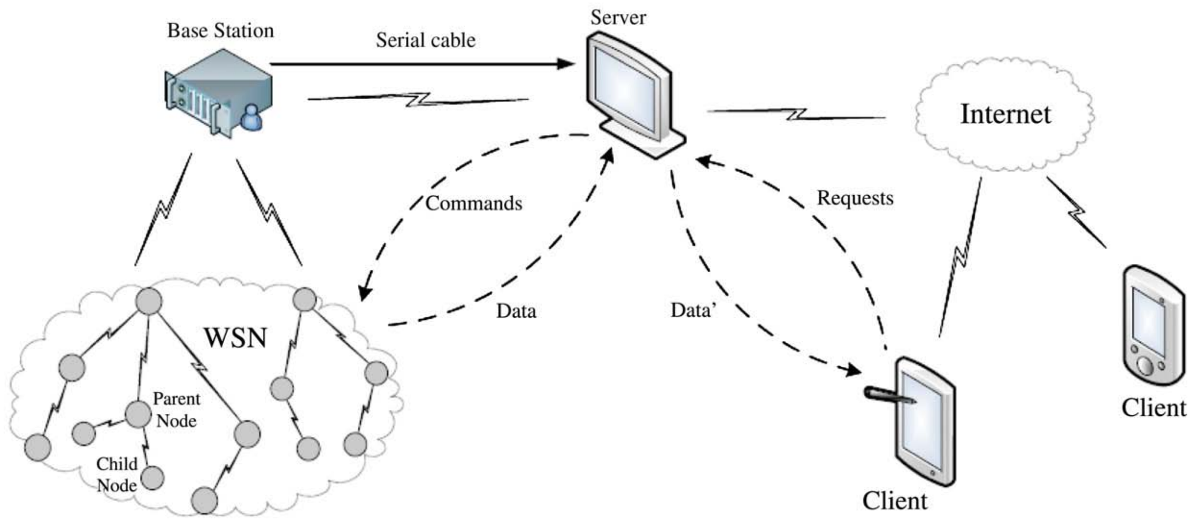

Fig. 1. Architecture of the whole platform. The dashed arrows indicate the interactions between the Server, the WSN, and the Clients, although they are actually indirectly connected. Note that Data and Data¢ are different, but the latter is from the former, see Section 4 for more details.

that can provide real-time charting, data logging and mote configuration, and TinyDB [10] that eases WSN information extracting and data query are of low extensibility limited applications in scientific researches. SpyGlass [2], a WSN visualizer developed with Java, and jWebDust [6], an Internet integrated application environment based on Java, did not put enough attention on improving data processing. Different from the above systems or tools, our system focus more on the data processing efficiency as well as its extensibility.

The rest of this paper is organized as follows. In Section 2, the hardware of the WSNs is discussed briefly. The architecture and details of the Server tier and the Client tier are demonstrated in Sections 3 and 4 respectively. Section 5 presents one way of the system realization and illustrates its experimental results. Finally, Section 6 concludes the paper.

2. Hardware tier synopsis

There have been a number of papers devoting to the design of hardware devices design as well as software operating systems. To date, many types of sensor nodes have been developed, for instance, Rene (Rene2), Mica series motes, WINS [5], etc. They characterize in small sizes, low data rate, low-cost and unreliable wireless communication. Deployed by certain amount, these sensor nodes are able to self-organize and automatically communicate with each other in the multi-hop way. Focusing on many practical applications, whatever the infrastructure and the network protocols, the sensor nodes will send information up to a sink node or to the base station of the whole sensor network. By deciphering and analyzing that information, one can get insights into what we care. For instance, the environmental condition, existence of any enemy and emergencies (such as a fire) of the sensor field; the network current state including failures of sensor nodes, reliability of current multi-hop routing, and the present wireless communication status, etc. In fact, the sensor nodes are so intelligent that they are able to tune the functioning parameters of their own, which greatly enables the so-called over air tuning, i.e. the network users could control the sensor nodes though wireless interaction. This is very useful when the control targets are mobile sensors or actors.

In the following sections, the details about the upper tiers of our proposed platform are presented with above mentioned issues accounted.

3. Server

The Server is basically designed as an interface between the terminal users and the hardware tier. As in Fig. 2, it mainly consists of three layers,

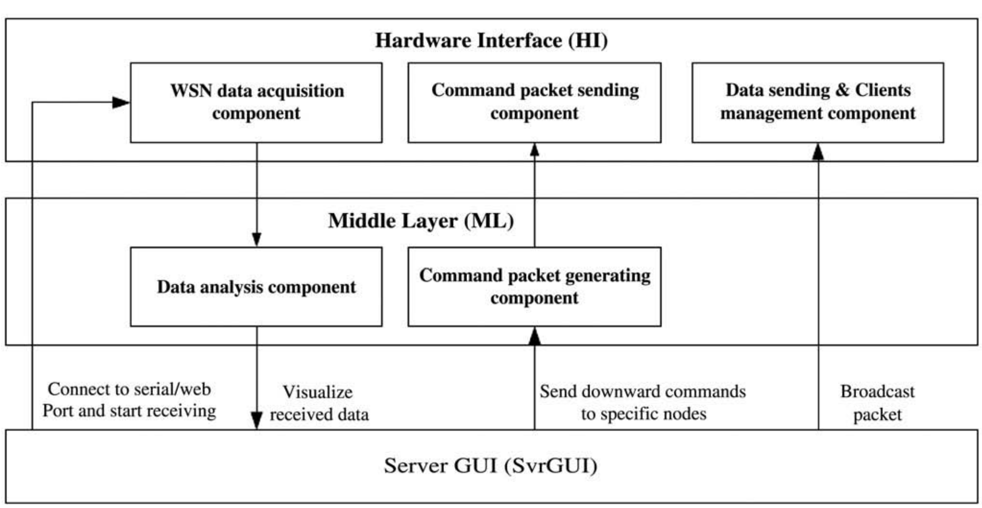

Fig. 2. Layers and components of the Server software.