QoS-Enabled Integration of Wireless Sensor Networks and the Internet

Weilian Su and Bassam Almaharmeh
Naval Postgraduate School
Department of Electrical and Computer Engineering
Monterey, CA 93943
USA

Abstract: Recent developments in sensor networking for both military and civilian applications emphasized the need for a reliable integration of sensor networks with the Internet. For sensor networks deployed in various military applications, it is important that collected information be delivered as fast as possible with minimum delays. In this paper, an integration module is proposed. The objective of the module is to provide preferential services for high-priority traffic. The integration module is implemented and tested using hardware equipments, such as Cisco routers and 10/100 Mbps switches. According to the testbed measurements, the proposed integration module is able to adapt to different traffic needs, thus, ensuring the QoS for different sensor network applications.

Key–Words: Wireless Sensor Networks, QoS, Integration Module.

1 Introduction

Thanks to significant technological advances in integrated circuit technology, the miniaturization of electronics has produced a far-reaching technological revolution in the sensors industry, which has enabled construction of far more capable yet inexpensive sensors, processors, and radios. Currently, very tiny sensors are produced commercially for a wide range of applications, such as habitat monitoring and remote surveillance [1, 2]. It is expected that within the next few years, the size of sensor nodes will continue to shrink, and sensor networks may cover the globe resulting in hundreds of thousands of wireless sensor networks (WSNs) scattered over battlefields, large farms, and warehouses.

One main issue in WSNs is to deliver the collected information efficiently with minimum delays to data centers, where different pieces of information are brought together to provide a better picture. This can be achieved by integrating the WSNs and the Internet with existing Quality of Service (QoS) techniques.

The integration of WSNs and the Internet is becoming more and more important because of the numerous numbers of WSNs that will join the Internet domain. Currently, the data gathered by WSNs are delivered to data centers with best-effort services, which means that delay-sensitive and time-sensitive data are subject to be dropped or delayed in congested networks.

With all these requirements and challenges in mind, an integration module is introduced in this paper. It is implemented and tested using hardware equipments, such as Cisco routers and 10/100 Mbps switches. According to the testbed measurements, the proposed integration module is able to adapt to different traffic needs, thus, ensuring the QoS for different sensor network applications.

The paper is organized as follows: In Section 2, we describe the integration architecture and its hardware and software components. Afterwards, we discuss about the performance analysis and results in Section 3. Lastly, we conclude in Section 4.

2 Integration Architecture

Data collected by sensor networks are required to be transmitted promptly to users of the Internet for analysis and intelligence gathering. The collected data propagate back to a *sink* or *gateway*. The gateway is a more complicated sensor node that has sufficient capabilities to query and communicate with other sensor nodes within the coverage area. Typically, the gateway is connected to an enterprise network or to the Internet in order to send/receive data to/from deployed WSNs. The connection between the gateway and the Internet is either a wired or wireless link. In the following subsections, the proposed integration module is discussed.

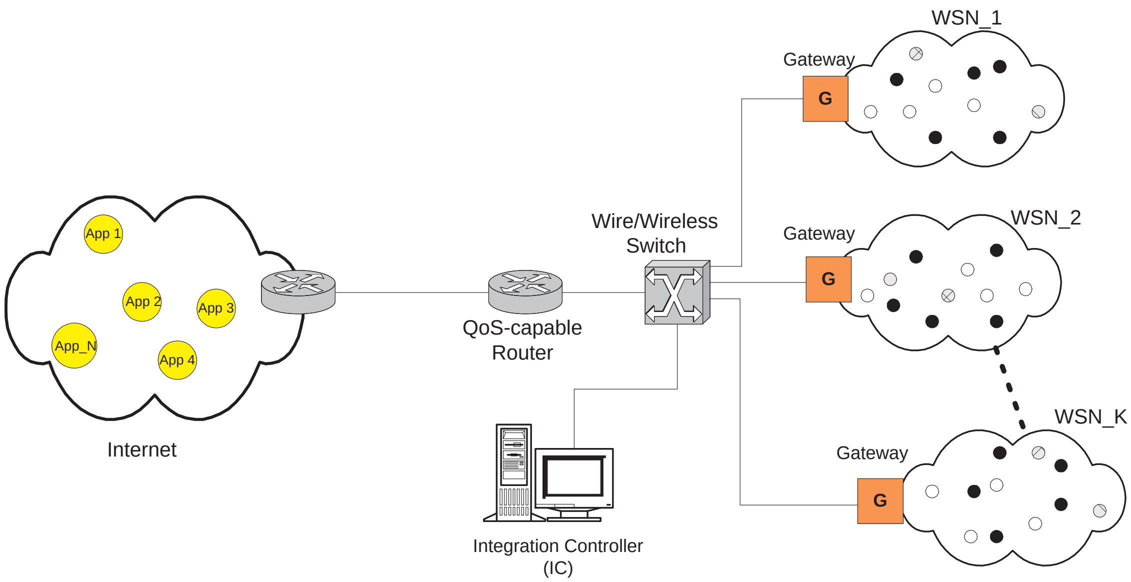

Figure 1: The proposed Integration Module.

2.1 Proposed Integration Module Overview

This paper introduces an integration module for the wireless sensor networks and the Internet. Figure 1 shows the architecture of the proposed integration module. The proposed integration module aims to facilitate the flow of packets between the Internet and the WSNs. It has the following objectives:

- Making the traffic less susceptible to delays and congestions.
- Providing security and reliability through the implementation of the wireless sensor networks registration protocol (WSNRP), which is not discussed in this paper due to space limitation.
- Reducing the setup time of the integration link.
- Self-adapting QoS functions to match changes on traffic pattern.

As shown in Figure 1, the core component in the proposed integration module is the integration controller (IC). With the help of the IC, the integration process goes through three phases: registration, control, and monitor. Initially, all interested sensor-applications and wireless sensor networks are required to register with the registration service manager (RSM) that runs on the integration controller (IC). The registration process helps identify the interests and capabilities of both sensor-applications and WSNs. The registration phase is carried out with the help of the wireless sensor networks registration protocol (WSNRP), which is not discussed in this paper due to space limitation.

After the registration phase, the IC enters the control phase. It reconfigures the QoS parameters on the network edge router to adapt to the new registered information. Next, the IC monitors the traffic on the integration link to determine if there is any abrupt changes to the traffic pattern. If there are link failures or congestions, the IC will enter the control phase to adapt to these changes.

2.2 Hardware and Software Components

As shown in Figure 1, the proposed integration module consists of wireless sensor networks (WSNs), an access point, applications, edge router, and integration controller (IC). The following subsections describe each component in detail.

2.2.1 Wireless Sensor Networks

Each WSN is able to sense a number of attributes, which will be called topics in this paper. The number of supported topics within a network depends on the capabilities of the available sensor nodes. A WSN will declare the supported topics by simply registering them with the registration authority represented by the registration service manager (RSM), which will be discussed later. Each registered topic has two arguments: the priority level and the reliability.

The priority level of a certain topic tells how urgent the data of that topic within a certain network is, while the reliability argument tells how reliable the data collected for that topic on a certain network are. The priority information associated with a topic during a communication session will be used to determine the link resources, such as bandwidth.

The reliability information associated with each topic is crucial to the applications on the Internet. An application on the Internet uses the reliability information to filter out the data that do not satisfy the reliability requirements of that application.

2.2.2 Access Point

To have multiple WSNs connected to the Internet, an access point should have multiple wireless/wired interfaces. The wireless interface could be a multiple access wireless technology such as the IEEE 802.11. Throughout the experimental phase of our research, several wired interfaces were used, i.e., Ethernet, Fast Ethernet, and variable-speed serial connections.

2.2.3 Applications

Users on the Internet run applications that query the WSNs. Applications on the Internet receive the collected data from WSNs for analysis and intelligence gathering. Throughout this paper, the term sensor-application will be used to describe those applications running on the Internet and interested in the WSNs. Sensor-applications could run on different hosts and one host could have more than one sensor-application. It is assumed that there is an unlimited number of sensor-applications on the Internet, which initially have no previous knowledge of the available WSNs. So, when a sensor-application first starts up, it sends a registration request to the registration service manager (RSM) in order to receive a list of valid WSN candidates.

Each sensor-application has a certain priority level, which describes the importance of the application. The priority information is used to provide appropriate QoS for sensor-applications.

2.2.4 Edge Routers (Cisco 2651 and 2811)

The Cisco 2651XM router is used to aggregate the wireless sensor networks' traffic. Also it is used to implement the QoS profiles decided by the QoS control manager (QCM).

The other router was the Cisco 2811. It was used as an entry point to the Internet. Cisco 2651XM and 2811 routers are driven by a powerful CPU processor along with high-performance DSP and auxiliary processors on various interfaces. The most preferable feature on these routers is their support for the QoS, which include: packet classification, admission control, congestion management, and congestion avoidance [7]. They also support advanced QoS features such as the resource reservation protocol (RSVP), weighted fair queuing (WFQ) [6], and IP precedence.

2.2.5 Integration Controller (IC)

The hardware part of the integration controller (IC) is a stand-alone laptop running Linux Redhat 9. The software part is a number of software components programmed using Perl (Practical Extraction and Report Language) language. The IC is a multi-function device that performs a number of tasks at the same time in order to facilitate the integration process. The IC represents the intelligent component of the integration module because it performs most of the sophisticated operations, makes decisions, and sends commands to other integration components.

The IC runs three main software modules: the registration service manager (RSM), QoS control manager (QCM), and network monitor manager (NMM). Each module has its own task and they work together to accomplish the integration mission. The three software modules are discussed in detail in the following subsections.

2.3 The Registration Service Manager (RSM)

The RSM is the registration authority for both sensor-applications and WSNs. The registration process helps in determining the QoS parameters needed for every registered client. Registered parameters, such as priority and reliability levels, play an important role in setting up the QoS functionalities.

The registration process is carried out with the help of the wireless sensor networks registration protocol (WSNRP). The RSM receives registration requests from both sensor-applications running on the Internet and from WSNs. The RSM keeps all the registration information in a local database file called the registry information file (RIF). The QCM and NMM also checks the RIF to see if there is any new or modified registration information. The RIF will be maintained and updated through the update messages received from the clients (either sensor-applications or WSNs).

RSM runs in two modes: server and client. The server side runs on the IC, while the client side runs on both sensor-applications and WSNs. The client side initiates registration request messages to the server, which processes the request and responds back to the initiator.

Before registering a new client, the RSM will look up any similar existing entries in the RIF. A new registry entry will be added to the RIF only if no duplicates exist. Each entry in the RIF table will be associated with a timer, and an entry in the RIF will be deleted if no update message is received within the timer life.

2.4 The Network Monitor Manager (NMM)

The NMM monitors the traffic at the integration backbone between the Internet and the WSNs. The NMM uses the *tcpdump* program, built in Linux, to monitor the traffic and sniff the packets going across the network. The NMM monitors certain traffic patterns and parameters such as the amount of traffic associated with each flow, congestion periods, data rates, and packet sizes.

The NMM sends two types of messages to the QCM: *periodic* and *event-driven messages*. The periodic update message is sent every five seconds, while the event-driven message is sent whenever there is an urgent and sudden change in the traffic patterns.

2.5 QoS Control Manager (QCM)

The QCM determines the best QoS profiles that must be used by the edge router based on the feedback information from both the RSM and NMM. All the configuration commands are sent to the edge router through a telnet session that is carried out with the help of two Perl libraries called Net::Telnet and Net::Telnet::Cisco [4].

QoS profiles mainly consist of queuing disciplines, traffic policing and shaping. Because of their superior performance over other queuing disciplines, both priority queuing (PQ) and class-based weighted fair queuing (CBWFQ) were selected as the queuing systems at the edge router.

Along with other queuing disciplines, PQ and CBWFQ were tested in a 108% overloaded UDP network at different data rates. PQ and CBWFQ outperformed other queuing systems such as the weighted fair queuing (WFQ) and custom queuing (CQ).

In addition to their superior performance, Cisco routers allow for the use of both disciplines simultaneously by enabling the low latency queuing (LLQ) system. This helps avoiding switching from one queuing system to another. Switching between different queuing systems reduces the router's response time, which decreases the performance of the system. Also, in most cases, switching from one queuing system to another brings the router's interface down and causes the link to fail.

The QCM has two phases: the initialization phase and the self-adaptation phase. When the QCM first starts up, it sends an initial QoS profile called QoS-Profile-Initial. The initial profile sets up the classification rules at the edge router (Cisco 2651XM) in accordance with the registered priority level information obtained from the registry information file. The initial profile defines five different classes named: class-urgent, class-high, class-medium, class-low, and class-normal. Each class is mapped to a specific precedence level(s).

Next, the low latency queuing system is enabled. This invokes both PQ and CBWFQ at the edge router's output interface. Each class is allocated a percentage of the queue's bandwidth. Initially, flows that belong to class-urgent, class-high, class-medium, and class-normal are allocated 40%, 20%, 10% and 5% of the queue's bandwidth respectively. The total bandwidth allocations must not exceed 75%. The other 25% is reserved by the router for overhead and best effort traffic. Only class-urgent will be assigned to the high-priority queue.

During the self-adaptation phase, the QCM adapts the QoS profiles at the edge router to address current changes in the network. Based on the periodic and event-driven messages from the NMM and registration information from the RSM, the QCM modifies the bandwidth allocations and forces some traffic policing and shaping on the traffic flows belonging to the classes configured at the initial phase. When receiving update messages from the NMM, the QCM first checks the flows and see if they are registered or not. If they are not registered, then the flows will be neglected and no preferential services are provided. If registered, then the QCM checks the registered priority level. If the registered priority level does not match the precedence level passed by the NMM, then the QCM notifies the RSM to modify the flow's registration information, and the QCM adds the current flow to the appropriate access-list. For example, if the current flow is of the urgent class, then it is added to the access-list that contains all the urgent flows.

3 Performance Evaluation

This section describes the lab setup and procedures to measure the performance of the proposed integration module.

3.1 Laboratory Setup

3.1.1 Equipment and Software Components

Figure 2 shows the simulation setup of the integration module at the Naval Postgraduate School's Advanced Networking Laboratory. This setup will be used to test and measure the performance of the integration module.

As shown in Figure 2, the laboratory setup consists of two QoS-enabled routers (Cisco 2651 and

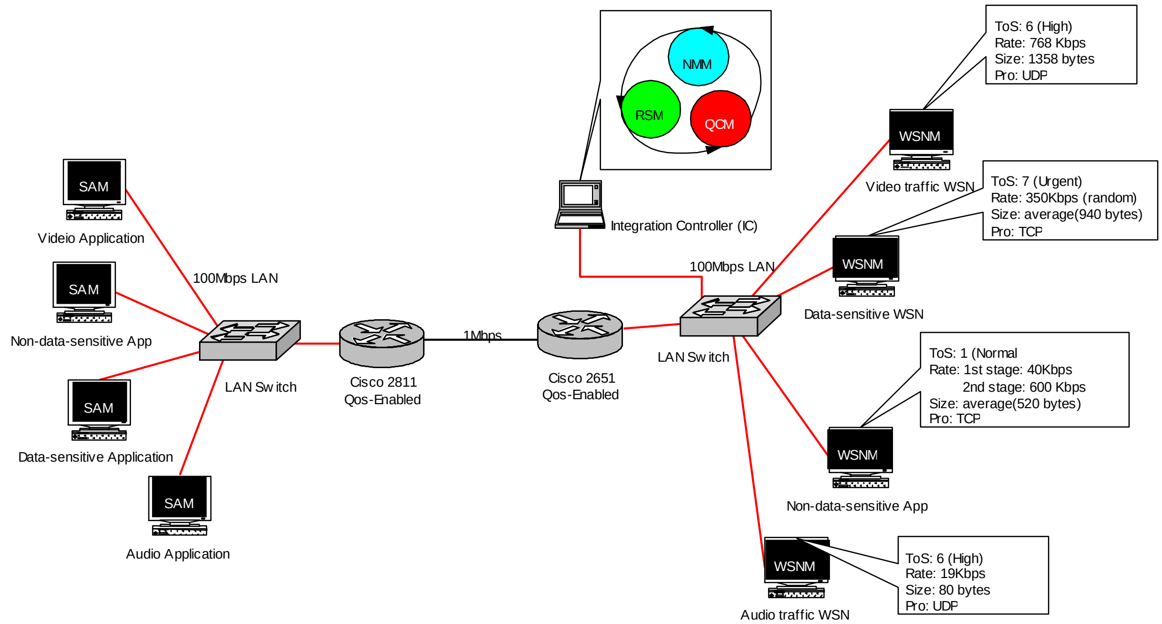

Figure 2: The laboratory setup.

2811) connected by a 1 Mbps serial link, an integration controller (IC), four personal computers (PCs) connected to a LAN switch to simulate applications on the Internet, and another four PCs connected to another LAN switch to simulate WSNs. The IC has three running software components: RSM, NMM, and QCM. Each computer on the wireless sensor network's side has a software component called the *wireless sensor network module* (*WSNM*). Similarly, each computer on the Internet side runs a software component called the *sensor application module* (*SAM*). Both the WSNMs and the SAMs are developed to simulate the wireless sensor networks and the sensor-applications, respectively.

Ethereal, an open source packet sniffer program, was used for capturing packets that traversed the integration link. A number of Perl scripts were developed to obtain the statistical parameters from captured packets. Also, Microsoft Excel's data analysis tools were used to plot some statistical results.

3.2 Results and Analysis

The following sections describe and analyze the results obtained from the two experiments: with and without the IC. The first experiment does not use the IC, and the traffic flows are disciplined by fair queuing (FQ). On the other hand, the second experiment uses the IC.

3.2.1 Video Flow Analysis

Inter-arrival time is the time between adjacent packets. For several applications, such as video and audio streaming, it is important to maintain a certain level of arrival rate for packets in order for those applications to work properly. Thus, we measured the inter-arrival time of the video flow under the FQ system and the integration controller.

The mean inter-arrival time for the video flow in the first experiment, under FQ system, was 20.38 milliseconds as compared to 14.76 milliseconds in the second experiment. At the peak point, 29% of the packets arrived between 10.8 and 12 milliseconds. The packet arrival rate is the reciprocal of the mean inter-arrival time. The packet arrival rate for the video flow was 43 and 68 packets/second for the first and second experiments, respectively. It is obvious that the video flow has suffered more delay under the FQ system than it has when subjected to the integration controller.

The number of video packets that arrived at the destination in the first experiment was 30,201 packets as compared to 38,847 packets that arrived in the second experiment. Thus, the second experiment had a performance improvement of 28%.

In the second experiment (integration controller), 88% of the video flow packets arrived between 10.8 and 12 milliseconds, compared to 29% in the first experiment (FQ).

3.2.2 Audio Flow Analysis

We investigated the packet delay of audio streams for the first and second experiments. The results indicate there is almost no difference between the two cases. FQ in the first experiment has treated the audio flow very well, and this is because FQ gives priority for low-volume flows over the high-volume flows. The mean delay time for the first experiment was 25 milliseconds, while the mean delay time for the second experiment was 23 milliseconds. The mean delay time for both experiments was the same over the first and second five minutes.

The low-rate audio flow has not been affected by the injection of the high data rate burst in both experiments. This is because FQ is an algorithm that has been designed to give low-volume flows preferential treatment over high-volume flows.

3.2.3 Sensitive Data Flow Analysis

The sensitive data flow is a TCP flow. The round-trip time (RTT) is an important parameter for TCP traffic. RTT is the time from sending a packet from a source host to the time an acknowledgment is received at the source host from the destination host. The relationship of RTT versus simulation time for the first and second experiments is studied. The mean RTT for the FQ system is 128 milliseconds, while the mean RTT with the integration controller is 60 milliseconds. The RTT value is almost the same over the entire simulation time for the integration controller case, while for the FQ case, the mean RTT was 120 and 136 milliseconds for the first and second five minutes, respectively.

We also calculated the cumulative distribution function (CDF) of the RTT for both experiments. For the first experiment, the RTT distribution spanned over a large period of time. The RTT range was from 0 to 1.2 seconds. As for the second experiment, it was from 0 to 0.28 seconds. The standard deviations for the first and second experiments were 7.07 and 1.18 milliseconds, respectively. The variances for the first and second experiments were 50 and 1.42 millisecondes, respectively. This means that there was a greater variation in the RTT for the first experiment (FQ) than for the second experiment (integration controller).

4 Conclusions

An integration module was proposed in this paper. The integration module core component was the IC. A testbed network was set up, and its goals were to test and measure the performance of the integration module and compare it with the performance of the fair queuing system. The results obtained from the testbed showed an improved performance with the IC.

References:

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks (Elsevier) Journal,* pp. 393-422, March 2002.

[2] H. Green, "Tech Wave 2: The Sensor Revolution," *Business Week On line,* August 25, 2003.

[3] "Cisco IOS Quality of Service Solutions Configuration Guide", Release 12.2, Cisco Systems, 2005.

[4] http://cpan.uwinnipeg.ca, last accessed on August 2005.

[5] L. Cheng, Y. Mang, T. Lin, and Q. Ye, "Integration of wireless, wireless local area networks and the Internet", *IEEE International Conference on Networking, Sensing & Control,* vol. 1 , pp. 462-467, March 21-23, 2004.

[6] J. B. Nagle, "On Packet Switches with Infinite Storage", *IEEE Transactions on Communication,* vol. 35, no. 4, pp. 435-438, April 1987.

[7] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking* vol. 1, no. 4, pp. 397- 413, August 1993.