Demo Abstract: The Sensor Andrew Infrastructure for Large-Scale Campus-Wide Sensing and Actuation

Anthony Rowe† Mario Berges‡ Gaurav Bhatia† Ethan Goldman‡
Ragunathan (Raj) Rajkumar† Lucio Soibelman‡

† Electrical and Computer Engineering Department
‡ Civil and Environmental Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213

{agr,mberges,gnb,ejgoldman,rajkumar,lucio}@andrew.cmu.edu

ABSTRACT

We demonstrate Sensor Andrew, an infrastructure for Internet-scale sensing and actuation across a wide range of heterogeneous devices. The goal of Sensor Andrew is to enable a variety of ubiquitous large-scale monitoring and control applications in a way that is extensible, easy to use, and secure while maintaining privacy. The core architecture is built around the eXtensible Messaging and Presence Protocol (XMPP) where transducers are modeled as event nodes in a push-based publish-subscribe architecture. Sensor Andrew allows for easy integration of new sensors as well as support for legacy systems. A data handler provides registration, discovery and data logging facilities for each device. The major elements of this architecture have been deployed in five buildings at Carnegie Mellon University, and are comprised of over 1000 sensing points reporting data from multiple communication interfaces. Our demonstration will showcase already active sensor systems on campus using Sensor Andrew through a web interface as well as the ability to locally configure sensor and actuator interactions as an example of how larger-scale applications could be built.

1. INTRODUCTION

Sensor Andrew is a large-scale effort to widely deploy sensing devices across Carnegie Mellon University. We envision a broad set of applications ranging from infrastructure monitoring, first-responder support, quality of life for the disabled, water distribution monitoring, building power monitoring and control, social networking and biometric systems for campus security. Researchers have already successfully built many similar sensor networking applications, but they are typically isolated, small-scale and short-lived experiments. One of the primary goals of Sensor Andrew is to have a living laboratory where applications can be rapidly prototyped at scale. Imagine an infrastructure monitoring system that could immediately alert the campus facilities personnel in the event of broken water pipes or power outages. At the same time, another application leveraging some of the same sensors could suggesting strategies for reducing building energy costs. Application developers should be able to directly utilize physical data from the environment without having to re-invent lower-level interfaces. To make this possible, there needs to be a common language to communicate transducer data that is scalable, extensible, easy to integrate with pro-

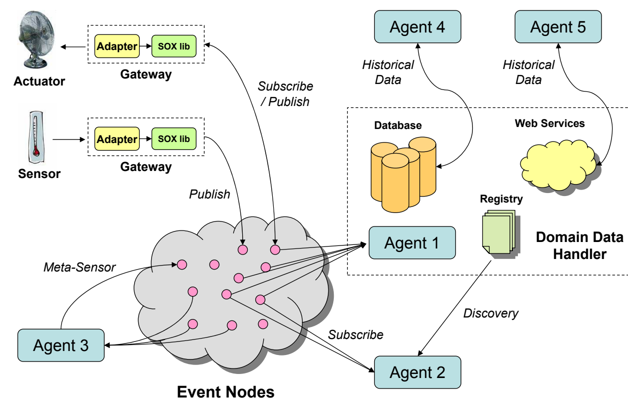

Figure 1: XMPP Publish-Subscribe transactions to support collection of sensor networking data.

cessing agents, that supports fine-grained access control and maintains privacy.

2. ARCHITECTURE

Sensor Andrew consists of a classical three-tired architecture comprised of servers, gateway devices and transducers. Servers and gateways are assumed to be Internet connected while devices at the transducer layer can be resource-constrained and communicate using any number of protocols or physical networks to a gateway. We leverage the eXtensible Messaging and Presence Protocol (XMPP) [1] Internet messaging protocol to provide both point-to-point as well as publish-subscribe communication. Addressing in XMPP takes the form of `name@server/resource` allowing for an extremely large and scalable namespace. Figure 1 shows the interaction between different components in the Sensor Andrew architecture. Each transducer is represented by an *event node* which is an XMPP addressable entity which allows clients to publish or subscribe to data feeds. Event nodes provide control for how applications are allowed to access transducers including access control lists. For example, a gateway that is subscribed to a resource-constrained wireless sensor node's transducer can shape how control messages are passed into the low-level sensor network to conserve battery life. Gateway class devices run *adapters* built using our Sensor Over XMPP (SOX) library that can interact with one or more event nodes. Sensor processing software called *agents* use the SOX library running on servers to subscribe to event feeds. Agents can process sensor data and in return publish high-level meta-sensor values back to event nodes. For example, an agent could combine temperature and humidity sensors to calculate dew point or use a light and CO_2 sensor to detect occupancy. A domain data handler is responsible for registering new transducers as well as logging values in a database. The domain data handler provides database, web service and XMPP access to historical data and provides device discovery.

One application running in Sensor Andrew is trying to optimize energy savings through non-intrusive electrical load monitoring of a building in conjunction with user feedback. To support this application, a number of different commercial and research-grade sensing devices are used: (1) Two electrical panels located in the Porter Hall building on the campus were retrofitted with EnerSure [2] electric circuit monitors. These devices acquire different power metrics from all the individual circuits in the panel, and can be polled via TCP/IP or RS-232 using Modbus. (2) A number of FireFly [3] power sensing nodes called *JigaWatts*, shown in Figure 2, are used to measure the consumption of separate appliances throughout the building as a way to obtain ground truth data. (3) Other power meters are used intermittently to obtain load profiles for certain appliances and/or as ground truth sources. (4) Finally, there are light intensity and temperature sensors scattered across different rooms in the building, which can provide useful information that will help to disaggregate various electrical sources (e.g. if a light is turned on, it will manifest itself as both a power draw and as a light intensity change).

Figure 2: FireFly wireless outlet energy monitoring and control device.

3. DEMONSTRATION

Our demonstration consists of two main parts. First, a web portal will show registration, visualization and configuration of transducers currently running in Sensor Andrew. Streaming cameras will provide real-time feedback showing how the portal can control lighting and report power information at different locations on campus. Next, we will show a local demonstration of a hazard monitoring agent that can be configured on-demand by a user to sense and actuate devices to react to events such as a water-pipe burst. Users will be able to build events and actions locally based on wireless environmental sensors and the *JigaWatt*. We can show a scenario where if water touches a leak sensor, then the system will be configured to disable wall-power to nearby appliances and send the user a text-message notification. These interactions will be comprised of multiple independent sensing systems that can now collaborate using the Sensor Andrew infrastructure.

4. REFERENCES

[1] Jabber Software Foundation . Extensible Messaging and Presence Protocol (XMPP): Core . *IETF RFC3920*, 2004.

[2] http://www.trendpoint.com/EnerSure.html (viewed 10/21/2008).

[3] Rowe A., Mangharam R., Rajkumar R. FireFly: A Time Synchronized Real-Time Sensor Networking Platform. *Wireless Ad Hoc Networking: Personal-Area, Local-Area, and the Sensory-Area Networks, CRC Press Book Chapter*, 2006.