PUSHNOTIFY IN WIRELESS SENSOR NETWORKS

BY: MOHAMMAD DANISH   SUPERVISOR: MRS. PRITEE PARWEKAR

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

ABSTRACT

This research work proposes an architecture to push Wireless Sensor Network's data onto the web. It integrates large-scale sensor networks with sensing applications and web. It collects and processes data from various sensor networks and enables large-scale data sharing and collaborations among users and applications on the web.

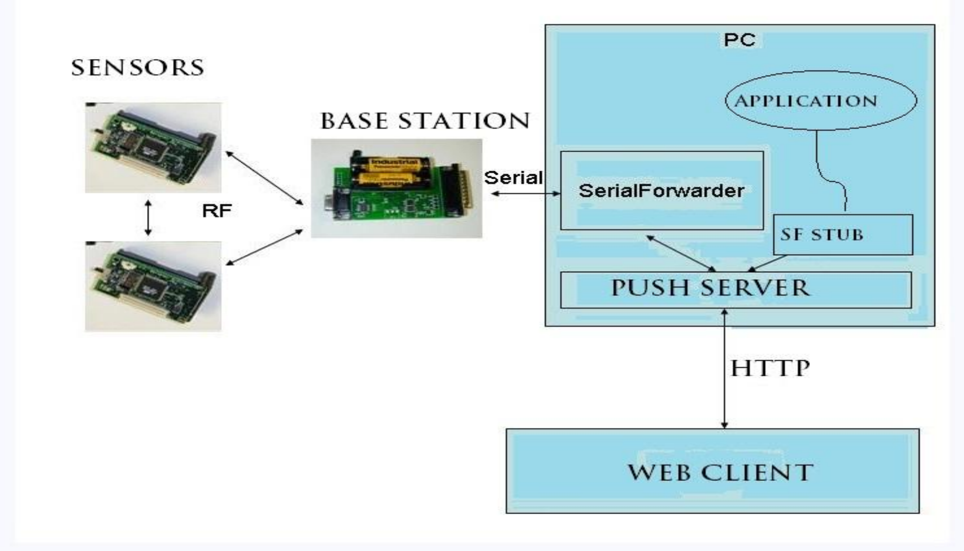
PushNotify is push server application which collects data from sensing application and sends that data onto the web. Initially PushNotify was developed to send updates/notices from fileserver of JIIT University to browser extension so that student/faculty can be notified of study material changes/urgent notices in real time. Research paper related to this has been published in IEEE Digital Explore. Now PushNotify is extended and modified to send sensing data onto the web.

MOTIVATION

Due to the rapid development of sensor technology, current sensor nodes are much more sophisticated in terms of CPU, memory, and wireless transceiver. Sensor networks are long running computing systems that consist of a collection of sensing nodes working together to collect information about, for instance, light, temperature, images and other relevant data according to specific applications. The ability of the sensor networks to collect information accurately and reliably enables building both real-time detection andearly warning systems.

However, the heterogeneous features of sensors and sensor networks turn the efficient collection and analysis of the information generated by various sensor nodes into a rather challenging task. The main reasons for that are the lack of both uniform operations and a standard representation for sensor data that can be used by diverse sensor applications. There exists no means to achieving resource reallocation and resource sharing among applications as the deployment and usage of the resources has been tightly coupled with the specific location, sensor application, and devices used.

IMPLEMENTATION



The implementation is divided into two parts:

- Sensor Application and
- PushNotify: Push Server

Sensor Applications:

Tinyos is used as a computing platform and nesc language is used to develop a sensing application which is to be deployed on sensors. We have scenario in which we have two sensing motes: one is temperature-humidity sensor and another will act as a BaseStation. We have two applications written in nesc, one for sensing temperature-humidity and another for Basestation.

The first step is to check that we are able to get our PC to communicate with a mote. Most motes have a serial port or similar interface. Telos motes also have a serial interface, but it talks to their USB hardware, which is similar in functionality but very different in terms of cables and connectors. The basic abstraction for mote-PC communication is a packet source.

A packet source is exactly that: a communication medium over which an application can receive packets from and send packets to a mote. Examples of packet sources include serial ports, TCP sockets, and the SerialForwarder tool.

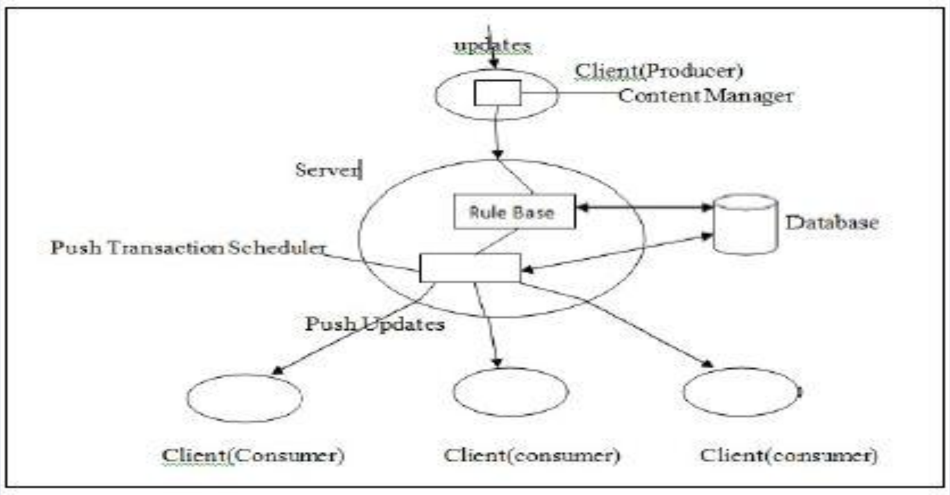
BaseStation is a basic TinyOS utility application. It acts as a bridge between the serial port and radio network. When it receives a packet from the serial port, it transmits it on the radio; when it receives a packets over the radio, it transmits it to the serial port. Because TinyOS has a toolchain for generating and sending packets to a mote over a serial port, using a BaseStation allows PC tools to communicate directly with mote networks.

Sensor node has SenseToRadio application installed on it to sense humidity and temperature in real time. This sensed data as a packets is send over radio frequency to base station. the data is sent by the mote in big-endian format. This format is independent of the endian-ness of the processor, because the packet format is an nx_struct, which is a network format, that is, big-endian and byte-aligned. Using nx_struct for a message payload ensures that it will work across platforms.

BaseSation on receiving these packets, sends them to the serial port. If we use directly there is a problem that only one PC can interact with the mote. Additionally, it requires you to run the application on the PC which is physically connected to the mote. The SerialForwarder tool is a simple way to remove both of these limitations. Most generally, the SerialForwarder program opens a packet source and lets many applications connect to it over a TCP/IP stream in order to use that source. For example, you can run a SerialForwarder whose packet source is the serial port; instead of connecting to the serial port directly, applications connect to the SerialForwarder, which acts as a proxy to read and write packets.

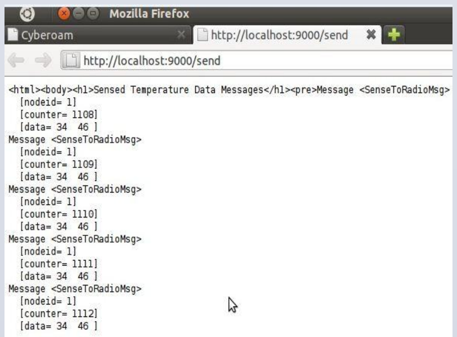
PushNotify Push Server:

Now PushNotify serve as a push server to collect data from serial forwarder and to push that packets to a appropriate subscriber connected through a web client. PushNotify receives request , process requets and builds rule base based on some predicates in request and sends response when predicate condition is met with some data update(data is push).

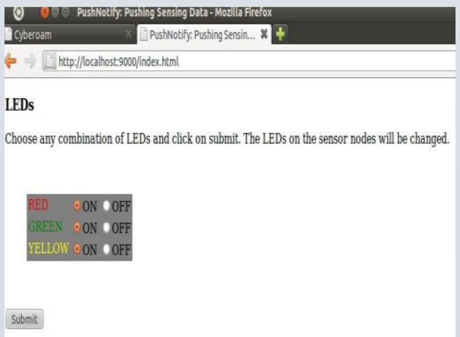


RESULTS

Data sensed by Telosb motes is send to the basestation and from where serial forward collects that data. PushNotify receives request from many users and based on request predicates, collects data from serial forwarder and forms a response and sends it to the appropriate user.



```
<html><body><h1>Sensed Temperature Data Messages</h1><pre>Message <SenseToRadioMsg>
  [nodeid= 1]
  [counter= 1108]
  [data= 34  46 ]
Message <SenseToRadioMsg>
  [nodeid= 1]
  [counter= 1109]
  [data= 34  46 ]
Message <SenseToRadioMsg>
  [nodeid= 1]
  [counter= 1110]
  [data= 34  46 ]
Message <SenseToRadioMsg>
  [nodeid= 1]
  [counter= 1111]
  [data= 34  46 ]
Message <SenseToRadioMsg>
  [nodeid= 1]
  [counter= 1112]
  [data= 34  46 ]
```


PushNotify: Pushing Sensing Data - Mozilla Firefox
Cyberoam
PushNotify: Pushing Sensin...
http://localhost:9000/index.html
Submit

LEDs

Choose any combination of LEDs and click on submit. The LEDs on the sensor nodes will be changed.

RED ● ON ○ OFF
GREEN ● ON ○ OFF
YELLOW ● ON ○ OFF

REFERENCES

1. R. Narayana Swamy, Dr. G.Mahadevan, "Event Driven Architecture using HTML5 Web Sockets for Wireless Sensor Networks " in IJETSE International Journal of Emerging Technologies in Sciences and Engineering, 2011.
2. Duquennoy, S., Grimaud, G. , Vandewalle, J.-J., "The Web of Things: interconnecting devices with high usability and performance" in International Conference on Embedded Software and Systems, 2009.
3. Xingchen Chu and Rajkumar Buyya, "Service Oriented Sensor Web" in Sensor Network and Configuration: Fundamentals, Techniques, Platforms, and Experiments, N. P. Mahalik (ed), Springer-Verlag, Germany, 2007.
4. Xingchen Chu, Tom Kobialka, Bohdan Durnota, and Rajkumar Buyya1, "Open Sensor Web Architecture: Core Services" in Proceedings of the 4th International Conference on Intel ligent Sensing and Information Processing, 2006.
5. Sachin Agarwal, "Towards a Push-Scalable Global Internet" in 14th IEEE Global Internet Symposium(GI) at IEEE INFOCOM 2011.
6. Mohammad Danish, Pritee Parwekar, "PushNotify: Push Server Application" in 3rd IEEE International Advance Computing Conference, 2013.