**IIIT\_H/ Spring 2015**

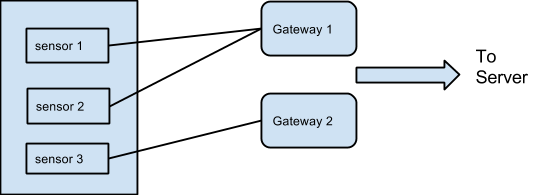
**Internals of Applications Servers**

**Project Module 1: Sensors, Gateways and communication**

1. Sensors, Gateways and Communication Module

This module is the sensors end of the platform. With help of this module, sensors can connect into the network, communicate with the gateway and generate data used in the platform.

Functional overview of Team’s module (Block diagram)



**2. Test cases**

2.1Test cases- used to test the team’s module

Use case : Critical Temperature Alert System

1. Test connect sensors into network.
2. Test disconnect sensors out of the network.
3. Test add new gateways to platform.
4. Test Send data to Gateway.
5. Test link between multiple gateways
6. Test check heartbit of particular sensor
7. Move sensor to another gateway group
8. Simulate various kinds of data using android simulators  
   Generate temperatures data in using simulators
9. Test check send data using faulty sensors simulation

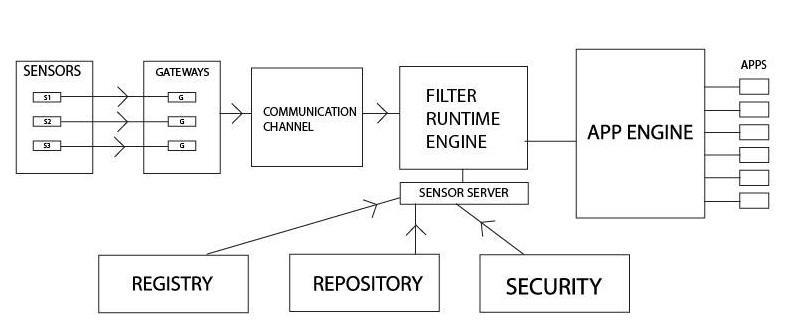
2.2Overall project test cases (relevant to the module)

Integration test cases:

1. Check if API calls are working properly at gateway end
2. Check integration of Gateways and filter server module.
3. Test list all sensors and gateways working at the moment.

**3. Solution design considerations**

3.1 Design big picture



3.2 Environment to be used

* Android device acting as simulator
* or Raspberry pi/simulator,
* RPC or Google protobuff
* gateways

3.3 Technologies to be used

* Node.js and Java
* Java for remote method calls

3.4 Approach for Device gateways

*Android device as a simulator*

*Type:* Gateway

Interfaces with sensors and filter sever

*Information structure:* <IP, port,<type, sensor id, location, sensed data, timestamp, checksum> >

*Raspberry pi*

*Type:* Gateway

Interfaces with sensors and filter sever

*Information structure:* <IP, port,<type, sensor id, location, sensed data, timestamp, checksum> >

3.5 Devices type/interfaces and information structure

- *Sensors*

*Types:* Temperature, humidity, camera, smoke, water level, seismic , traffic

Interfaces with gateways

*Information structure:* sensor id, sensed data

3.6 Approach to get device information (event streams)

Android device as a simulator or Raspberry pi will detect any new sensors in its vicinity and will keep track of live sensors using heartbeats from sensors. When a new sensor boots up , it sends its id and type information to nearest gateway. Gateway in turn will forward this information to filter server and filter server will check the registry. If the sensor information is not present in the registry , it will create a new entry.

3.7 Communication overview

Communication will be done using Sockets , RMI or REST API depending upon appropriate resource availability. If time permits we will try in en-corporate marshalling and un marshalling of data using google protobuf

3.8 Interactions between modules

* + Sensors to Gateway : Sensors send temperature data along with sensor id to the connected gateway over wireless channel. Gateways listen on a specific port for incoming data and provide an interface for sensors to communicate. This interaction will be implemented using java sockets. One sensor will be connected to only one gateway.
  + Inter-Gateway Routing : Gateways are used to route temperature data collected by sensors to filter server. Multiple gateways can be interconnected and can follow a routing protocol so that if any link between gateways goes down, data can still be routed to filter server using some other path. They will recognize other gateway nodes within range, set up simple routing tables of adjacencies, and discover likely paths to the appropriate integrators/ filter server.

3.9 Wire and file formats

* + *Sensors to Gateway:* <sensor id, sensed data>

**6. Key Data structures**

Details of device types :

Sensor type, sensor id stored at registry.

Wire formats

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sensor id | Sensor Location | Sensor data value | Timestamp | Checksum |

Gateway communication APIs and protocol

**7. Interactions & Interfaces**

**APIs**

APIs for sensors are

1. send(byte array[])

send data in the form of byte stream to gateway

1. format()

pack the data into the format provided by gateway.

1. configure(file xmlFile)

configure the sensor based on details provided in xml File.

* *Gateways:*

Gateways are the middle layer devices. They gather data from various sensors and collectively send them to app server.

APIs for gateway are

1. receive(byte array[])

receive stream of bytes from sensors.

1. format()

pack the data into the format provided by app server.

1. configure()

configure a gateway. Configuration include which sensors are connected to this gateway,type of sensor,routing protocol information,port on which the communication is happening etc.

1. forward( destinationIP, routingInfo)

check the destination IP of received packet and forward on the outgoing link based on routing information

Type Handler API

Load new type handler

Stop existing type handler

Get gateway list