Project 1 (50 points)

General Instruction

Students can use C or C++ as the base Programming language. Please make sure that the code can be compiled with Ubuntu gcc or g++. Students can form groups of at most two. Please follow the instructions of the project carefully**. [This project is borrowed and adapted from the CS677 course taught by Dr. Prashant Shenoy]**

Project given out: ***September 17th, 2015***
Project due: ***October 8th, 11:59 pm***

Project Description

Internet of Things (IoT) broadly refers to sensors and "smart" devices with network capabilities that can be employed to design various applications. Here we will use IoT concepts to design a distributed system that can be deployed in a "smart home" to automate simple tasks.

We will assume three classes of IoT objects:

1. ***Sensor***, which can sense surrounding conditions and report it.
2. ***Smart Device***, which can be remotely turned on or off
3. ***Gateway***, which is a tiny server that interacts with sensors and smart devices.

Sensors and Smart devices connect to the Gateway using sockets. Gateway should be multi-threaded such that it can accept requests from multiple sensors. Each item (Sensor, Smart Device or Gateway) is identified by an IP Address and a Port number. The different messages used for communication are given in the table given below.

Message Format

Each message should have the format given below.
"Type:<message_type>;Action:<action>"

Applicable Devices	Message Type	Action	Description
Sensor, Device	Switch	on, off	Switch the item on or off
Sensor, Device	currState	on, off	Sensor or device returns the current state of the device.
Sensor	currValue	<Integer value>	Sensor returns with current value being monitored. The values are integers.
Sensor	setInterval	<Integer value>	Set the interval in which the sensor need to respond with currValue
Sensor, Device	register	deviceType-ip-port-area deviceType:- sensor device IP :- Ipaddress Port :- Port number Area :- AreaID of item	Register the device to the gateway. Each item on registering send the message. Eg:- "sensor-10.23.23.33-80-1"

Example

Type:register;Action:sensor-10.23.23.33-80-1

This is a register message which tell the gateway that it's a sensor which is identified with IP Address 10.23.23.33 and Port number 80 connected to Area number 1.

Type:switch;Action:on

This message will switch on the item to which it is send.

Sensor

Sensors take two files as input.

1. Configuration File: it will have information about Gateway to be connected in the format given below

Line1 : gatewayIPAddress:GatewayPortNumber
Line2 : itemType:SensorIPAddress:SensorPortNumber:AreaID

Example Sensor Configuration file is attached.

2. Input File : This file contains the data which is input to the Sensor in the given format

Line1 : startTime1;EndTime1;Value1
Line2 : startTime2;EndTime2;Value2
.
.
.
LineN: startTimeN;EndTimeN;ValueN

In the input file StartTime(i) will be same as EndTime(i-1). When the specified time expires, it should start from Line1. Example file is attached. Each sensor should have a thread which updates the values.

When a sensor is switched on, it sends a message to the Gateway using the message type "register". Once the connection succeeds, the sensor will start sending data using "currValue" message at a time interval of 5 seconds. If it receives a message "setInterval", sensor should alter the time interval at which the messages need to be send. It can also receive "switch" messages from gateway, which switches the sensor on or off.

Smart Device.

Device takes one file as input.

1. Configuration File: it will have information about Gateway to be connected in the format given below

Line1 : gatewayIPAddress:GatewayPortNumber
Line2 : itemType:DeviceIPAddress:DevicePortNumber:AreaID

When a device comes up, it sends a "register" message to the gateway, with the information from the configuration file. It will then work according to the task. It sends the "currState" message whenever its state is changed to the gateway.

Gateway.

The Gateway takes one configuration file as input.

1. Configuration File: it will have information about Gateway to be connected in the format given below

Line1: gatewayIPAddress:GatewayPortNumber

When the Gateway process switches on, it will listen on the port for "register" requests. Whenever a new item registers with the gateway, it will assign an integer value in sequential order to the item. Gateway takes the inputs from the sensors and decides if it should "switch" the devices "on" or "off" depending on the task. Gateway should display the current state whenever some value from the sensors or the smart device changes. Example display is attached

Line 1: ----------
Line 2: <Item ID1>----<Ipaddress:portNumber>----<Type of item>----<AreaID>----<status>
Line 3: <Item ID2>----< Ipaddress:portNumber >----<Type of item>----<AreaID>----<status>
.
.
.
Line n+1: <Item IDn>----< Ipaddress:portNumber >----<Type of item>----<AreaID>----<status>
Line n+2:----------

<Item Id> - The integer ID assigned when the item is registered to the gateway.
< Ipaddress:portNumber> - IP Address and Port number of the current item.
<Type of Item> - Type of the item. Either "sensor" or "smartDevice"
<Area ID> - This is the area to which the item is assigned (Send by the item during registration)
<status> - If the item is a sensor, the value should be displayed. If the item is a smart device, it should display if it is on or off.

Please note that when the state is displayed, the first and last lines should be a line of "-"'s and in other lines, each quantity is separated with 4 dashes ("----")

Task

Your code must implement this scenario.

***Preventing water pipe bursts**:* The winter has seen many snow storms and artic colds and you are worried that the cold temperature may cause the water pipes to freeze and burst causing your home to flood. In order to prevent water pipe from freezing, you decide to buy a pipe heater, which is simply a small heater that you plug into a power outlet and it keeps the area near water pipes warm and avoid freezing of water pipes. However, rather than keeping the heater on all winter, which is a waste of electricity, you deploy a temperature sensor near the pipes and plug the heater into a smart outlet. When the sensed temperature drops below a certain value, the heater is turned on by the gateway actuating the smart outlet. When the temperature rises above a second threshold, it is turned off. Assume that the heater needs to turn on whenever the temperature drops below 32 degree Fahrenheit and is turned off when the temperature rises above 34 degree Fahrenheit. This way, the heater is on only during nights when the temperatures drop below freezing and is turned off during day hours when the temperature rises a little above freezing. There can be multiple areas, which can be identified numerically in the system with each area having a separate heater device. Each area can have multiple heat sensors connected to it. The AreaId is sent to the gateway when the device is being registered.

Deliverables

All groups should submit a ".zip" file which should contain the following.

1. src folder : It should contain all the source files (can have other folders inside it)

2. Instructions.txt: It should contain details on how to compile the source, step by step. For example
   Step 1: Open terminal and navigate to the src folder
   Step 2: Command to compile Gateway: <full command to compile>
   Step 3: Command to compile Sensor: <full command to compile>
   Step 4: Command to compile Smart Device: <full command to compile>
   Step 5: Command to run Gateway: <full command to compile>
   Step 6: Command to run Sensor: <full command to compile>
   Step 7: Command to run Smart Device: <full command to compile>

3. Readme.txt: This file should contain any additional instructions or comments you might have.
4. SampleInput Folder: This folder should have configuration files to run a system with 2 areas (meaning 2 smart devices) & 2 sensors connected to each of them (making 4

sensors). Also include a txt file “SampleTestRun.txt” which contains commands to run them and “SampleTestRunOutput.txt” with the display of gateway.

Submission system.

You will use blackboard to submit the zip file described above. The zip file should be called Project1_submission.zip.

PS: If you have any questions, please email TA sooner, than waiting for the last day of submission.