# Programming Assignment 3: Internet of Things – Fault tolerance, Replication, and Consistency

Fubao Wu 28277607 Ashraf Ali Shaik 28211687

# **Replication and fault tolerant Architecture**

The Gateway is made fault tolerant by adding an replication of data base that can take care of all the devices and sensors when failures occur. Gateway and its replication can also share load between them by sharing the sensors and devices which they can co-ordinate. A Cache is present between database and gateway to improve performance.



# **Load Balancing**

Load balancing is achieved by sharing the number of devices and sensors each gate way and replica will handle. Initially all the sensors and devices will talk to the gateway. The gateway takes control of first registered device/sensor and then assigns the control of second registered device/sensor to Gateway replication. Third to gateway, fourth to replication and so on. This ensures load is equally distributed. The devices/sensors are registered at Gateway or Gateway Replication as shown in the figure.



Initially Devices/Sensors have Ip address and port details of only Gateway. But after assigning some of them to replication. The Gateway replica will send its ip address and port number to devices/sensors under its co ordination so that they can directly interact with Replication.

# **Consistency (Eventual Consistency)**

Consistency is achieved by maintaining the state information of sensors and devices at both Gateway and Gateway Replica. When ever a Sensor or device pull or push its state to the gateway or replica we have updated the state in another by a non blocking thread. An example test case below explains how consistency is achieved in the design.

We have cache consistency and disk database consistency. When there is write update in one gateway, it would write into the cache and immediately send to its gateway replica to keep consistency. Every replica should be consistent to the last updated values.

For database replication, we have cache written, we would check database whether it is needed to be update or not. If needed, we will write back to database. Write-write conflict is not frequent. If there is multiple sensors or devices write to different gateway replicas, we will write into different cache using locks.. But we we do consistency, receving message and write it into cache before check whether it is locked or not. Also try to use database file locking or avoid writing database at the same time.



Drawing 4: LRU Cache Implementation

#### Fault tolerance:

To make the system operate continuously, the gateway is replicated. But if the gateway is fault, how do we do the detection and recovery.We use event consistency to keep consistent between gateway and its gateway replication called asGw\_replica.We will talk about how do we detect the replica. We send the heart message between two replicas.

Gateway A push heartbeat message every time\_interval(set as 3seconds), Gw\_replica A' also sends a heartbeat message every other a time\_inteval(set as 3 seconds as well).A and A's will also initialize a separate thread task to detect the pushed message from the other side. The Time interval is also set 3 seconds here.But it will detect for the maximum times(set as 3 here), it reaches 3 times continuously, it will judge the other side is disconnected, we here simply think it is crashes!If one replica crashes, the other would take over the functionality of crashed one.

If Gw\_replica is initially registered (connected) with motion sensor, bulb smart. After Gw\_replica crashes, these motion sensor and bulb would be notified by the Gateway. Then all the sensors and devices would communicate with the gateway. After the gateway replica recovers, the motion sensor and bulb smart would be notified again. It will be notified again to connect to replica gateway replicas.

#### **Explain the effect of fault:**

Because we use event-consistency which is weak consistency. It doesn't fullfill the stronger constraints. We can basically guarantee exchanging versions or updates of data between gateway. If gateway has updated ,it will immediately send to gateway replica. Or gateway replica has updates, it will also send to gateway to keep consistent immediately.

In our system there are not frequent writing. But if there are multiple sensors write to gateway and gateway replica's at the same time, one gateway replica's value is likely to be lost. The impact of such data loss in our system is not huge. Because if we find data consistency is not consistent, we do query data from sensors to check which one is consisten with the sensor and update the new value with the sensors!



Drawing 5: FaultTolerance operation Timing Diagram



# **Paxos Design details**

Assuming there are 4 replicas. As shown in above figure

*Step 1:* A Device queries for its state to gateway and all replicas. Take Temperature sensor as example and we are querying for temperature.

*Step 2:* A leader or proposer will be selected randomly. Let the leader in this case be Gateway replica4. It sends prepare message to all the other gateway replicas including the election (agreement) number.

*Step 3:* If the election (agreement) number is greater than the election number present at the acceptors then acceptors will send the temperature along with OK message.

*Step 4:* The Leader then find outs the value with majority and broadcasts all the intended results to all other replicas.

*Step 5:* Acceptors acknowledge with an OK message it they have no concern with the value selected by the proposer.

*Step 6:* Leader will send the Agreed value to all other replicas. Upon receiving the value all the replicas can transmit the temperature value to the temperature sensor.

Temperature sensor can then only take the first temperature value and discard the next duplicate responses. Temperature value sent by all replicas is redundant but it makes the operation to be almost guaranteed even if one packet of data looses due to network failure. The consecutive responses can be identified and discarded by caching the received value at the sensor end and then matching the identical received packets and making them eligible to discard.

## **Implementation Platform details:**

- → Java
- → RMI
- $\rightarrow$  Linux Platform supported scripts.

## Instructions for running the Code and Test scripts:

### The code can be can using two different scripts:

**1.run\_LoadBalance\_Consistency\_Caches.sh** --- This script is used to demonstrate the implementation of loadBalance, Consistency and Querying for state of a device through Cache.

**2.run-part\_FT.sh** --- This script is used to demonstrate our application is **fault tolerant**. Even if one of the Gateway replicas crashes, the other one will take control.

We have decentralized our code into various Java Packages where each Package corresponds either to a component (the gateway, a sensor or a device or Back end database). In order to avoid the complexity and exceptions due to the supporting code being in different packages, **In addition to the source code** we have also provided the executable jar files for each component above described. We have Submitted Source code to verify code and Jar files to execute.

#### **IP Address Recognition and Allocation:**

We are needed to provide only the IP Address of the Gateway in the Configuration file (**configips.csv**). The Default IP Address in configips.csv is local host. This IP Address is needed for all the other components. Each Component can figure out the value of its IP Address when initiated and will register at the Gateway. Gateway stores the IP Address and can access the other components when required.

#### Jar files & command line arguments in various cases:

#### For LoadBalance, Consistency, Caching (run\_LoadBalance\_Consistency\_Caches.sh):

The following Jar files will take the command line argument as Path to the Configuration file . **Please place all the Jar files, configips.csv in the same Directory.** 

```
gnome-terminal -x sh -c "java -jar GatewayServer.jar configips.csv; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar GatewayServerReplica.jar configips.csv; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar motionSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar tempeSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar DoorSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar HeaterSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar BackendDatabase.jar configips.csv; bash"&
```

**For Fault Tolerance (run-part\_FT.sh):** The following Jar files will take the command line arguments as Path to the Configuration file and "lab3\_test". "lab3\_test" is used to distinguish between the above mentioned LoadBalance mode and Fault tolerance mode. **Please place all the Jar files, configips.csv in the same Directory.** 

```
gnome-terminal -x sh -c "java -jar GatewayServer.jar configips.csv lab3_test; bash"&
sleep 3
gnome-terminal -x sh -c "java -jar GatewayServerReplica.jar configips.csv lab3_test;
bash"&
sleep 3
gnome-terminal -x sh -c "java -jar motionSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar tempeSensor.jar configips.csv; bash"&
```

```
gnome-terminal -x sh -c "java -jar HeaterSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar bulbSmart.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar DoorSensor.jar configips.csv; bash"&
gnome-terminal -x sh -c "java -jar BackendDatabase.jar configips.csv; bash"&
```

### **Executable Script files:**

**run\_LoadBalance\_Consistency\_Caches.sh** : This Script files corresponds to the automation of the *"For LoadBalance, Consistency, Caching"* case discussed above. This script is executable and has all permissions. User Just need to run this script and Terminals Pop up.

(i)Load Balance will occur automatically and The nodes which are assigned to Gateway Replica will show which they are assigned . The Gateway and Gateway Replica terminal also displays the registered nodes under them. See Appendix 1 screenshot of obtained results.

(ii)For consistency Testing User need to manually enter and report .For example take temperature sensor and it is assigned to Gateway. Even if we report to Gateway the value will also be updated in Gateway replica. User can see that value is updated in both replicas. User Input Needed. The few starting output in terminal will convey the operation performed by that process. See Appendix 2 screenshots for obtained results.

(iii) For Cache testing User need to follow the instructions on the screen in **Gateway terminal or Gateway Replica terminal**. On entering one of the Integer values of **Device/Sensor Ids** as mentioned in the screen Querying takes place and Initially user will get a cache miss as this is the first entry and we read from database. If user query using the same **Device/Sensor Ids** then a cache hit occurs and we get data from cache. User Input Needed. The few starting output in terminal will convey the operation performed by that process. See Appendix 3 screenshots for obtained results.

**run-part\_FT.sh** : This Script files corresponds to the automation of the "*For Fault Tolerance*" case discussed above. This script is executable and has all permissions. User Just need to run this script and Terminals Pop up. See Appendix 4 screenshots for obtained results.

(i) As terminals pop up the Gateway and Gateway Replica will be sending Heart beats to each other.

(ii)**Enter 1** in either Gateway or Gateway Replica **to crash** one of them. After that user will not see any heartbeats.

(iii)**Enter 0** in the previously crashed Gateway/Gateway Replica **to recover** the Gateway/Gateway Replica. User will see heart beats being exchange again.

#### **Performance Analysis:**

We have performed experiments on the performance by measuring the delay from between Gateway and the Sensors.

- → Push Performance delay test which Gateway Received push temperature from Motion sensor.
- $\rightarrow$  Pull Performance delay test which Gateway pull temperature from temperature sensor

Interval(average taken)	500ms	1000ms	2000ms	5000ms	10000ms
push(report) delay(ms)	3.454	2.898	3.572	3.452	3.126
pull(report) delay(ms)	4.224	3.554	4.232	4.522	4.134

Broadcast delay(ms)	3.756	2.967	3.421	3.653	2.983
Load balancing additional avg delay(ms)	0.457	0.348	0.569	0.317	0.613
Time saved by caching(ms) = (Database AccessTime) - (Cache Access Time)	2.568	3.185	2.157	2.893	3.027

 $\rightarrow$  From basic statistics of this table, the delay of push and pull are very small where as pull delay is a more than push.

- $\rightarrow$  An overhead of time due to load balancing is observed while assigning nodes to replicas.
- $\rightarrow$  Caching have helped in reducing access time than accessing the external file every time.

## Possible improvements and extensions to your program

(1) the consistency is not very strict with eventual consistency. If there are frequent multiple sensors write to same gateway and replica, there will be some lost, so we can sequential consistency to improve it to be sequential consistency

(2) the fault tolerant is based on heartbeat. It is difficult to differentiate the long delay and dead. We can improve to accumulate historical experience about the reponse time and approximate estimation response time to decide whether it is long delay or really dead.

## **Conclusion:**

 $\rightarrow$  Application can run in two modes one for testing one mode tests the implementation of load balancing, consistency and caching. Another for testing Fault tolerance.

 $\rightarrow$  Paxos conceptual design is illustrated.

 $\rightarrow$  Performance analysis of various cases were specified based on delay between various communicating nodes.

 $\rightarrow$  We have extensively tested all the Test cases as mentioned above to ensure the application is executing as expected.

**Appendix 1: Load Balance** 

Terminal	🗰 🖘 🖗 🖬 🖂	📋 (70%) ◀)) 4:18 AM 🛟
🛜 🙆 🖨 🗇 Terminal		s   🐉 Java
Gate Way Started	Gate Way Replica Started	
iplocalhostMultiThreadReceive begin here.	iplocalhostMultiThreadReceive begin here.	×
MultiThreadReceive java RMI registry created.	MultiThreadReceive java RMI registry created. querying data base	X X •
Enter Device ID	Enter Device ID	a a v
1 for Temperature	1 for Temperature	
2 for Motion	2 for Motion	m.GatewaySe
3 for Door	3 for Door	icastControlli
DataBase is Registered at Gateway           Door Sensor is registered at Gateway	came into take controlBULB1 came into take controlMOTION0	port : int
Temperature Sensor Registered at Gateway	Smart Bulb is registered at Gateway	pAddress : Str
	Motion Sensor is registered at Gateway	
	came into take control after assign5	ookup : String
	came into take control after assign2	oSend : Strii
	came into sensor came into sensor	ageory : Str
	came into sensor came into take controlOUTLET1	int
🕞 🐨 🐨 🐨 Terminal	Smart outlet is registered at Gateway	castContro
Read Gateway IP from Configuration File!	came into take control after assignó	() : void
iplocalhostBackend Data Base java RMI registry created.	came into Device	
Ready to Log data from other devices and sensors	🛛 😂 😑 Terminal	
	Read the IpAddress from the Configuration file Door Sensor java RMI registry created.	
	😣 🖨 💷 Terminal	
S 🖨 🛈 Terminal	Read Gateway IP Address from Configuration file!	
	Temp Sensor java RMI registry created. Report to Gateway or Gateway Replica enter Teperature	
Read Gateway IP from Configuration File! [plocalhostSmart Bulb java RMI registry created.	Report to Gateway of Gateway Repitta enter reperature	
Terminal		
HeaterImpl java RMI registry created.		
Need to Report the State Enter Y or N		
Gate Way is Assigned to you		



#### **Appendix 2: Consistency**

Terminal	🗱 🛜 🖬 🖾 🗑 (70%) 🕪 4:37 AM 🔱
🔀 🗧 🖉 🗇 🗊 Terminal	🛇 🗢 🗉 Terminal
Gate Way Replica Started a iplocalhostMultiThreadReceive begin here. a uplocalhostMultiThreadReceive java RMI registry created. a querying data base Enter Device ID 1 for Temperature 2 for Motion 3 for Door came into take controlDATABASE2 DataBase is Registered at Gateway	Cache Miss Unlucky Guy!!! The Current entry in Database isMachine1 2015-05-06 02:40:24.536 Sensor T emerature is12 The Current is4 Cache has space and updated 1 The Current pin Database isMachineMachine1 2015-05-06 02:40:24.536 Sensor T tewaySe emerature is12 querying data base Enter Device ID int
came into take control after assign7 came into Database came into take controlTEMPERATURE0 Registered at Gateway Replication rerminal	1 for Temperature ress:Str 2 for Motion p:String 3 for Door Cache hit Lucky Guy!!! temperature sensor's current temperature is : 12 temperature sensor's current temperature sensor's current temperature sensor's current temperatur
🗾 🚽 🛛 🕒 🐨 Terminal	😒 🗇 🗉 Terminal
<pre>iplocalhostMultiThreadReceive begin here.     #ultiThreadReceive java RHI registry created.     #ultiThreadReceive java RHI registry     came into take controlDATABASE2     Came into take control After assign1     came into take controlTEMPERATURE0     #Registered at Gateway Replication     came into sensor     Updated Tempearature State for consistency to12     Updated Tempearature State for consistency to13     Updated Tempearature State for consistency to35     Updated Tempearature State for consistency to35 </pre>	temperature sensor's current temperature is : 33 temperature sensor's current temperature is : 13 3 querying data base Enter Device ID 1 for Temperature 2 for Motion 3 for Door The Current entry in Database isMachineMachine3 2015-04-09 11:15:21.436 Sensor D Controlli 0 oor is Open Cache hit Lucky Guy!!! ismart outlet device's state is off Door's event reported= eventHeaterOffSensed temperature sensor's current temperature is : 35 temperature sensor's current temperature is : 44 poor's event reported= eventHeaterOffSensed Door's current temperature is : 44 smart outlet device's state is off Door's current temperature is : 44 smart outlet device's state is off Door's current reported= state is off Door's current reperature is : 44 smart outlet device's state is off Door's current reperature is : 44 smart outlet device's state is off Door's current reperature is : 44 stContro
Read the IpAddress from the Configuration file Notion Sensor java RMI registry created.	13 reported Report to Gateway or Gateway Replica enter Teperature 35 reported Report to Gateway or Gateway Replica enter Teperature 44 reported Report to Gateway or Gateway Replica enter Teperature
HeaterImpl java RMI registry created. Need to Report the State Enter Y or N Need to Report the State Enter Y or N	Read the IpAddress from the Configuration file Door Sensor java RMI registry created.
Read Gateway IP from Configuration File! iplocalhostSmart Bulb java RMI registry created.	Read Gateway IP from Configuration File! iplocalhostBackend Data Base java RMI registry created. Ready to Log data from other devices and sensors Gate Way Replication is Assigned to you

## Appendix 3: Caching

Terminal	🗱 🔶 🖬 🖂 🗑 (70%) 40) 4:33 AM 🔱
🗾 🗧 🛛 🖨 🗉 Terminal	😣 🔿 🗊 Terminal
Gate Way Replica Started Gate Way Replica Started In iplocalhostMultiThreadReceive begin here. MultiThreadReceive java RMI registry created. MultiThreadReceive ja	1 for Temperature 2 for Motion 3 for Door Cache Miss Unlucky Guy!!! The Current entry in Database isMachine2 otion exists 2015-04-09 11:35:31.324 Sensor M
2 for Motion 3 for Door Came into take controlDATABASE2 DataBase is Registered at Gateway came into take control after assign7 came into Database	The Current is4 atewaySe Cache has space and updated tControlli 2 The Current entry in Database isMachineMachine2 2015-04-09 11:35:31.324 Sensor MEint otion exists dress:Str guerving data base
Came into take controlTEMPERATURE0 Registered at Gateway Replication Came into take control after assign1 Came into take control after assign1 Came into take control after assign1	Enter Device ID     :up:String       1 for Temperature     send:String       2 for Motion     geory:String       3 for Door     th       Cache hit Lucky Guy!!!     astContro
Motion Sensor java RMI registry created.	HeaterImpl java RMI registry created. Need to Report the State Enter Y or N
NOT TErminal	S 🖨 🙃 Terminal
Read Gateway IP Address from Configuration file! Temp Sensor java RMI registry created. Report to Gateway or Gateway Replica enter Teperature Gate Way is Assigned to you	Read the IpAddress from the Configuration file Door Sensor Java RMI registry created.
🖉 🖨 🗊 Terminal	🕃 🔿 🗉 Terminal
Read Gateway IP from Configuration File! tplocalhostSmart Bulb java RMI registry created.	Read Gateway IP from Configuration File! iplocalhostBackend Data Base java RMI registry created. Ready to Log data from other devices and sensors Gate Way Replication is Assigned to you

Terminal	🗱 🤝 🖬 🔽 🗑 (70%) 🜒 4:34 AM 🕸
💦 📑 🛛 🔿 🗊 Terminal	😣 🕒 🗉 Terminal
Gate Way Replica Started	1 for Temperature
iplocalhostMultiThreadReceive begin here.	2 for Motion 🗖
MultiThreadReceive java RMI registry created.	3 for Door
Less Marying data base	Cache Miss Unlucky Guy!!!
Enter Device ID	The Current entry in Database isMachine1 2015-05-06 02:40:24.536 Sensor T emerature is12
2 for Motion	The Cussent is4
3 for Door	Cache has space and updated
TTT ▶ came into take controlDATABASE2	1 Controlu
DataBase is Registered at Gateway	The Current entry in Database isMachineMachine1 2015-05-06 02:40:24.536 Sensor T <mark>:int</mark>
came into take control after assign7	emerature 1:512 dress : Str
Come into Database	querying data base
Registered at Gateway Replication	1 for Temperature Send: Strin
came into take control after assign1	2 for Motion geory: Str
A G G G Terminal	3 for Door
Read the IpAddress from the Configuration file	Cache hit Lucky Guy!!!
Motion Sensor java RMI registry created.	
	🛇 🖨 🗉 Terminal
	HeaterImpl java RMI registry created.
	Need to Report the State Enter Y or N
	S 🖨 🐵 Terminal
C C C Terminal	Read the IpAddress from the Configuration file
Read Gateway IP Address from Configuration file!	Door Sensor java RMI registry created.
Temp Sensor java RMI registry created. Report to Gateway or Gateway Replica enter Teperature	
Gate Way is Assigned to you	
	🕲 🖨 🗊 Terminal
CO Terminal	Read Gateway IP from Configuration File!
Read Gateway IP from Configuration File!	iplocalhostBackend Data Base java RMI registry created.
iplocalhostSmart Bulb java RMI registry created.	Ready to Log data from other devices and sensors
	Gate Way Replication is Assigned to you

Illustration 3: Caching Implementation Cache Miss and Cache Hit

#### **Appendix 3: Fault Tolerance**

Terminal	🗱 🤶 🖬 🖂 🗑 (70%) 🕸	4:02 AM 🔱
Z S S S S S S S S S S S S S S S S S S S	😣 🔿 🗊 Terminal	= =
🔍receive gateway's HeartBeat Message	id = 1	
Plareceive gateway's HeartBeat Message	ipAddress = localhost	
Goreceive gateway's HeartBeat Message	id = 2	
receive gateway's HeartBeat Message	ipAddress = localhost	
f1receive gateway's HeartBeat Message	Detect GateWay Replica is crashed, needs to take over gateway replica	
	id = 1	
Enter 1 to cause this replica fault/crash(not sending hearbeat); En	ter 0 to reccipAddress = localhost	
	id = 2	
receive gateway's HeartBeat Message	ipAddress = localhost	
🔁 🗖 receive gateway's HeartBeat Message	receive gateway replica's HeartBeat Message	
	receive gateway replica's HeartBeat Message	
Enter 1 to cause this replica fault/crash(not sending hearbeat); En		
	receive gateway replica's HeartBeat Message	
receive gateway's HeartBeat Message	receive gateway replica's HeartBeat Message	
receive gateway's HeartBeat Message	pace/controlcoposcoponen	
📃 🛛 🖯 🙂 Terminal		
Read Gateway IP Address from Configuration file!	HeaterImpl java RMI registry created.	
🔍 🖉 Temp Sensor java RMI registry created.	Need to Report the State Enter Y or N	
Report to Gateway or Gateway Replica enter Teperature	<u>G</u> ate Way is Assigned to you	
Notified GateWay Replica Crashed		
Notified GateWav Replica Crashed	Read Gateway IP from Configuration File!	
📃 🔋 Terminal	iplocalhostSmart Bulb java RMI registry created.	
Gateway IP from Configuration File!	Notified GateWay Crashed	
calhostBackend Data Base java RMI registry created.	Notified GateWay Crashed	
geody to Log data from other devices and sensors	Notified GateWay Crashed	
	Notified GateWay Crashed	
	Read the IpAddress from the Configuration file	
	Motion Sensor java RMI registry created.	
	Gate Way is Assigned to you	
	Gate Way is Assigned to you	
	Notified GateWay Replica Crashed	

Illustration 4: Gateway Failure Detection

Terminal	🗱 😓 🕲 (70%) 💷 4:01 AM 🔅
🚬 🗸 🛛 😑 🗉 Terminal	S S Terminal III III III III III III III III III I
Detect GateWay is crashed, needs to take over gateWay Plid = 3 ipAddress = localhost came into door	receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message Oreceive gateway replica's HeartBeat Message
<pre>ipAddress = localhost recetive gateway's HeartBeat Message recetive gateway's HeartBeat Message recetive gateway's HeartBeat Message recetive gateway's HeartBeat Message Grecetive gateway's HeartBeat Message Grecetive</pre>	Enter 1 to cause this gateway fault/crash(not sending hearbeat); Enter 0 to reco ver receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message
Frecetive gateway's HeartBeat Message recetive gateway's HeartBeat Message recetive gateway's HeartBeat Message	receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message
Report to Gateway or Gateway Replica enter Teperature	HeaterInpl java RMI registry created. Need to Report the State Enter Y or N Gate Way is Assigned to you
Contended of the second	Cateway IP from Configuration File! iplocalhostSmart Bulb java RMI registry created. Notified Gateway Crashed Notified Gateway Crashed Notified Gateway Crashed Notified Gateway Crashed Notified Gateway Crashed Notified Gateway Crashed
<b>1</b>	ugu Douligation Egilure Detection

Illustration 5: Gateway Replication Failure Detection



#### Illustration 6: Gateway Replication Recovery

Terminal	👗 🔶 👜 🖂 🗑 (70%) «× 4:00 AM 🔱
💌 🖌 🛇 🖨 🗉 Terminal	😵 🖱 🗉 Terminal 🛛 👘
<ul> <li>Construction</li> <li>Const</li></ul>	receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message Ireceive gateway replica's HeartBeat Message Enter 1 to cause this gateway fault/crash(not sending hearbeat); Enter 0 to recover receive gateway replica's HeartBeat Message receive gateway replica's HeartBeat Message
tpAddress = localhost	
So Terminal	
Read Gateway IP Address from Configuration file! Report to Gateway or Gateway Replica enter Teperature Configuration File! Configuration File! CocalhostBackend Data Base java RMI registry created. CocalhostBackend other devices and sensors	HeaterImpl java RMI registry created. Need to Report the State Enter Y or N Gate Way is Assigned to you
	Read the IpAddress from the Configuration file Motion Sensor java RMI registry created. Gate Way is Assigned to you Gate Way is Assigned to you

Illustration 7: Heart Beat Messages