International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-2, December 2019

# SRAA: A Load Distribution Policy to manage the Resources Dynamically in a Smart City

**Priya Matta, Bhasker Pant, Sachin Sharma**

***Abstract—There are a variety of buzz words linked to advanced technology and technological developments in the modern era, out of which the world is primarily geared towards the Internet of Things. It is a network of things and devices capable of communicating with each other as well as connecting to the Internet, these devices are also embedded with the capabilities of electronics and computing. IoT is a smart city once incorporated with a city. A Smart City's architecture and growth has its own challenges. In addition to the other technological and social challenges, resource management is one of the most critical obstacles in achieving effective and seamless smart city execution. Load balancing is one of asset management's major components. Our research addressed the idea of IoT and a smart society with regard to a smart city, inspiration and challenges. The work proposed focuses on balancing the load in a smart city. Our main contribution is an algorithm for balancing the load in a smart city between different resource providers. SRAA has two variants in the proposed algorithm, which are explained. The algorithm modifications are performed using two case studies. The algorithm provided is suitable for a dynamic environment.***

***Keywords—Internet of Things, smart city, resources management, load balancing, SRAA.***

## I. INTRODUCTION

The development and incredible proliferation of technology and technological activities is moving the world into an Internet-of-Things (IoT) framework that is impressive and exceptional. In 1999, while collaborating with Auto-ID Laboratories, Kevin Ashton, a British businessman, coined the term IoT. The Internet of Things (IoT) spreads its networks throughout the globe, across all continents, across all physical entities in the world, out of a variety of renowned paradigms. The Internet of Things (IoT) is a wide range of tools, software and applications for use. It cannot be clarified or encapsulated in a concept that is simple and special.

**Revised Manuscript Received on December 30, 2019.**
* Correspondence Author

**Priya Matta*,** Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. Email: mattapriya21@gmail.com

**Bhasker Pant,** Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. Email: pantbhaskar2@gmail.com

**Sachin Sharma,** Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, Uttarakhand, India. Email: sxsharma88@gmail.com



In one way or the other, many academics and researchers identified IoT. There is no specific fully approved and settled definition of IoT. According to [1], "IoT represents a holistic environment that interconnects a large number of heterogeneous physical objects and things such as equipment, systems, livestock, cars, farms, factories etc. to the Internet in order to improve the efficiency of applications such as logistics, manufacturing, agriculture, urban computing, home automation, ambient assisted living and variety.

According to [2], "the popular IoT system involves three major technologies: embedded systems, middleware and cloud services, where embedded systems provide front-end devices with information, middleware links front-end heterogeneous embedded device systems to the cloud, and finally, the cloud offers comprehensive storage, processing and management mechanisms." A full-scale, comprehensive IoT platform has three major capabilities, namely: application enabling, data storage, and connectivity management, according to [3]. The IoT parameters are composed of 4A's and 4S's, according to [4]. The 4A's are Context and Position Awareness; Analyze and Take Action; Automate and Autonomous; and Anticipate, according to them. Similarly, Simplicity, Security, Smart, and Scalable reflect the 4S.

An effective description of IoT is the use of interconnected smart devices, available in the physical world, to practice some already existing processes, to improve some existing processes, or to facilitate some previously unavailable naïve ideas. Such tools and their supporting hardware are known as the IoT environment's major resources. Such services are required and accessed in the society by different end users and given by various providers of resources. When different users make resource requests, an effective load for resource providers is created. This load has to be distributed and balanced to take full advantage of the system.

There are five sections in this journal. The definition of smart city is discussed in section 2, after introducing the introduction to IoT in section 1, followed by inspiration and challenges in the same section in smart city. The complete system design, data structure, notation, policies and the proposed algorithm, namely Sender-Receiver Agreed Algorithm (SRAA), were elaborated in Section 3. The SRAA algorithm has two variants. Two case studies in section 4 demonstrate all variants of the SRAA algorithm. Section 5 finishes our research and addresses the context of the future.

## II. SMART CITY

### A. What is a Smart City?

If we think about innovation, technology improvisation, the Internet of Things (IoT) is the most common word we refer to.

Likewise, Smart Living is the term we should rely on whenever we think about technological progress along with social progress. Smart living is about a smart eco system, a smart house or a smart community to live in.

The concept of Smart City was created by residents of a city, its infrastructure, and the presence of smart connectivity via ICT. A smart city's main integrals are smart things, their networking, their ability to communicate and exchange data with each other. A smart city's most awaited feature is smart home.

Other than that, a few more names were added to the list: smart business, smart manufacturing, smart shopping, smart traffic, smart transportation, smart protection and smart healthcare. Smart wearables are again a smart life's trendy facet. For their effective life and efficient execution, all of these ideas depend on smart things or smart devices. Miniaturization of devices and exponential growth of wireless connectivity are two of the technology's main enablers behind the IoT theory. Because of the presence of IoT, physical objects and devices are turned into smart objects and smart devices. These intelligent devices bridge the gap between all physical devices. Since there is no single understood point of view to describe the smart city definition, in their literature reviews, numerous scholars have reported several different dimensions [5,6,7]. Some of a smart city's concepts emphasize ICT implementation while others emphasize the services being provided. According to [5], "a city with the smart industry" can be called a smart city. Smart industries are defined here as the industries activated by ICT, where ICT is deployed in their production process. Schaffers et al. [8] described "smart city as an ecosystem of open and user-driven technology to test and validate potential Internet-enabled services."

According to Balasubramanian and Cellatoglu [9], "Smart City deals with the technical enhancement of the living environment in order to provide help and improve the quality of life of its inhabitants." According to Dameri [10], "A smart city is a well-defined geographical area in which high-tech technologies such as ICT, logistics, energy production and so on work together to generate benefits for citizens in terms of well-being, inclusion and engagement, environmental quality, smart growth; it is governed by a well-defined pool of subjects, capable of setting out rules and policies for city government and development. "Simply put, a city that promotes the presence of smart objects and their intercommunication in order to carry out various decisive acts is known as a smart city.

### B. Motivation behind the Research

A number of factors depend on the appeal to the idea of a smart city following the notion of a virtual city. Therefore, these factors can be categorized as political, social and technological.

Economic factor acts as a significant step towards a smart city's materialization. A town that came into sight as a smart town certainly has a higher profile than a normal town. An enhanced city profile is a key indicator of the city's economic growth and is thus a major contributor to the nation's economic growth. This includes cost-effective management of the city's services [11]. Societal variable addresses better living standards. Based on Matta P. et. al [12] the IoT definitely assures the improvement of people's living standards as well as the tremendous improvement of IoT capabilities offered by the Internet.

There are different requirements for people living in the city, including static requirements and dynamic requirements. Fixed requirements may include regular gas supply, electricity, parking facilities, while some examples of dynamic requirements are traffic analysis, shortest path search, sudden and unexpected medical requirements.

The transformation of a city into a smart city makes it easy and timely for people to take advantage of the daily needs, critical and basic services and complex requirements.

A technical aspect is the last but most important factor. Technological factors serve as critical inspiration and help for a city as a smart city to grow. This consists basically of three sub-categories.

- Device portability
- Wired or wireless connectivity availability.
- Digital devices cost-effective.

These three factors of motivation are the key enablers of the technology behind the Smart City concept.

### C. Challenges

There are a wide range of challenges faced during the development of an IoT device, according to many researchers [13], [14], [15], [16]. When operating on an IoT project, a wide range of technological and social problems are experienced.

Once a city is considered to be a smart city, there are many obstacles to its transformation into a smart city. Most researchers and experts often address these issues from time to time. These challenges include the challenges of technology, security, infrastructure and finance, policies and governance, resource management, distribution of loads and balance. Some social problems are also faced when creating a smart city, such as lack of trust shown by people, lack of clarity about the advantages of a smart city.

Governance is a central obstacle of any smart city project, according to [5], [17], [18], [19], [20]. According to them, some major changes to the government model are needed. Some researchers [18], [21] focused on security and privacy. Security threats, data leakage and "right to privacy" are the unavoidable issues as we push towards the smart city notion. Therefore, since the devices are connected to the internet and many other devices, safety is a very important challenge [22].

According to HP, "about 70% of smart city IoT systems were at risk of attack due to enough vulnerabilities such as inadequate authorization, inadequate code security, and poor encrypted communication protocols"[23].

Challenges to a smart city can be classified into three main groups: infrastructure, security, privacy, and operating costs. 24]. That's wrong. Price, when planning a smart city, is a very important aspect. Development costs and operating costs are two elements of cost. The first component, i.e. development cost, is incurred only once in the entire lifetime, while the second component, i.e. operating cost, is equivalent to maintenance cost, which is essential to maintaining and preserving the smart city's impression [24]. Operational costs can be minimized by effectively managing the load, rather than increasing the number of resources beyond the limits. Load balancing is another compulsory challenge, according to [21].

In order to ensure efficiency and service quality, the load must be balanced, especially during peak hours [22]. This challenge can be defined as support for data traffic, storage and even application layer. According to Matta P. et al [25], "In order to exploit IoT's maximum capacity and benefits, all IoT-related assets must be handled efficiently and well-planned."

Once a Smart City is built and IoT is introduced, its complete, planned and faultless implementation poses many challenges. In all new research fields, resource management has stayed at the forefront, so in a smart city. Two essential components of resource management are load transfer and load balancing. Our paper focuses primarily on the problem of load balancing.

## III. PROPOSED ALGORITHM

Basically, our algorithm is designed to balance the load between different nodes. The service providers are the nodes here. Growing service provider provides one or many information requesters with some services. Like in a smart city, some companies can provide a service, for example. Airtel, Idea, Vodaphone etc. offers a very valuable asset "network bandwidth." When load is allocated to resource providers and their demands are met, a new resource demand can be dynamically generated. If a resource provider receives new request, it may get overloaded, or if a resource is issued to a resource provider, it may get overloaded. A proper load balance is therefore needed. In the proposed Sender-Receiver Agreed Algorithm (SRAA), this load balance is expected. Through mutual consent of overloaded and underloaded resource providers, the load of one resource provider can be shifted to some other resource provider in this algorithm. Suggest minimum value maintains this load balance.

The model of the process, data structure, algorithm notes are shown below.

System Model

- There are ‘n’ number of resource providers.
- There are ‘m’ number of resource types.
- There are ‘p’ number of resource requesters.
- A resource type can be provided by any ‘k’ number out of ‘n’ providers, where $k \leq n$.
- A resource provider can be in either of these three states:
  - Idle: If a resource provide is not actively participating in the system intentionally or unintentionally. In case the resource provider is not properly functioning, it is known as unintentional idle
  - Heavily Loaded: If the number of tasks (resource service) is more than a predefined threshold value, it is a heavily loaded provider or a source node. Such a node designates itself a SENDER.
  - Lightly Loaded: If the number of tasks (resource service) is less than a predefined threshold value, it is a lightly loaded provider or a sink node. Such a node designates itself a RECEIVER.
- Request generated by a resource requester is in a format of a three-column tuple, R ($RR_{id}$, $R_{id}$, $T_r$)

where $U_{id}$ is the unique ID of an end user, $R_{id}$ is the ID of the resource, $T_q$ is the timestamp of generated query and $T_r$ is the timestamp of request made by end user.

### A. Data Structure Used in the Algorithm

- Resource Provider Matrix (RPM): Resource Provider Matrix (RPM) is a ‘N’ X ‘M+1’ matrix maintained by the system, that keeps the track of all the requests being serviced by the all available resource providers, where ‘N’ is the number of resource providers and ‘M’ is the number of resource types.

RPM[i][j] = k, it means ‘k’ number of resource type $RT_j$ have been allocated to some requesters by resource provider RPi, where $1 \leq j \leq m$.

RPM[i][m+1] = k, it means ‘k’ is the total load of resource provider $RP_i$, as RPM[i][m+1] is the total sum of all the requests to the Resource Provider RPi, it can be well understood by following equation

$$RPM[i][m+1] = \sum_{j=1}^{j=m} RPij$$

- Resource Provider Table (RPT): Resource Provider Table (RPT) is a table with a three-tuple record, of the form ($RP_{id}$, Status, Load).

Where:

$RP_{id}$ depicts Resource Provider ID,

Status contains a value

-1: if Resource provider is Lightly loaded, or RECEIVER

0: if Resource provider is Idle

1: if Resource provider is Heavily loaded, or SENDER

Load: depicts the total number of resources allocated by the corresponding resource provider + number of requested resources.

### B. Notations Used in the Algorithm

- Resource Request Message: Request generated by the requestee is in a format of a three-column tuple, ($RR_{id}$, $RT_{id}$, $RP_{id}$)

where $RR_{id}$ is the unique ID of the resource requester

$RT_{id}$ is the ID of the resource type,

$RP_{id}$ is the ID of the resource provider.

- Resource Release Message: Whenever a resource is released, resource provider generates a Resource Release Message in a format of a three-column tuple, ($RR_{id}$, $RT_{id}$, $RP_{id}$)

where $RR_{id}$ is the unique ID of the resource requester

$RT_{id}$ is the ID of the resource type,

$RP_{id}$ is the ID of the resource provider.

### C. Policies Used in the Algorithm

There are basically three policies used during complete execution of algorithm. These are as follows:

- Transfer Policy: In every IoT system, a *threshold* will be calculated, or must be predefined on the basis of the configurations of providers and the type of services they are providing. Every provider whose total load is more than or equal to the threshold value, is designated as heavily loaded provider, or the *source node*.

Similarly, provider whose total load is less than the threshold value, is designated as lightly loaded provider, or the *sink node*.

- Selection Policy: Selection of *source node* will be accomplished automatically in Request Initiated Version of SAAR Algorithm, as the source node itself will be the initiator of algorithm.

While selection of *sink node* can be done in two ways, namely:

- First selection
- Best Selection

In first selection, the very first provider that is able to get the request transferred to itself, is selected as *sink node*.

In Best selection, all the possible providers are first listed. After listing them, all the listed providers are arranged according to their individual loads in an ascending order. The provider with the smallest load is selected as final *sink* node.

Both policies have their own pros and cons. In First selection, the consumption of time is quite lesser than that in Best selection. Similarly, Best selection is better in terms of scalability.

- Update Policy: Whenever a request is transferred from one resource provider to another, i.e. from source node to sink node, the status and load of both the nodes must be updated accordingly.

*a. Algorithm*

This algorithm can be initiated in two different ways, therefore, there are two different versions of SSAA:

**Request Initiated:**

1) Begin
2) Resource requester generates a request for specific resource to a resource provider. Request generated by the requestee is a Resource request message, in a format of a three-column tuple, ($RR_{id}$, $RT_{id}$, $RP_{id}$).
3) After analysing this request, RMT[i][j] is incremented in RPM, where 'i' is the ID of Resource Provider and 'j' is the ID of Resource Type, by performing the following
RPM[i][j] = RPM[i][j] + 1;
4) On updating this value, RPM[i][m+1], will be incremented by 1, as we know
RPM[i][m+1] = $\sum_{j=1}^{j=m} RPij$
5) After calculating total load of a resource provider and comparing it with threshold value, the RPT is updated as follows:
If (RPM[i][m+1] ≥ Threshold)
then,
STATUS [$RP_{id}$]= 1; //(declared Heavily Loaded)
LOAD[$RP_{id}$] = RPM[i][m+1];
Else if (RPM[i][m+1] = 0)
then
STATUS [$RP_{id}$]= 0;
LOAD[$RP_{id}$] = 0;
Else
STATUS [$RP_{id}$]= -1; //(declared Lightly Loaded),
LOAD[$RP_{id}$] RPM[i][m+1];
Endif
6) If the status of a Resource provider is either IDLE or Lightly Loaded, allocate the resource, as there is no requirement of load balancing, otherwise perform next step.
7) If the status of a Resource provider is 'Heavily Loaded' (i.e. 1), find a resource provider with status 'Lightly Loaded' (i.e. -1), by performing the following steps:
For (l=1; l ≤ m; l++)
{
Set a variable FOUND=0;
If (STATUS[l]== -1)
Assign variable Found=1;
Break;
}
8) If FOUND =1, transfer the request to Resource provider with ID=l by performing following steps
RPM[l][j] = RPM[l][j] + 1;
RPM[i][j] = RPM[i][j] – 1;
Go to step 2
Otherwise, wait till some resource is released.
9) End

**Release Initiated:**

1) Begin
2) Resource provider broadcasts a resource release message, in a format of a three-column tuple, ($RR_{id}$, $RT_{id}$, $RP_{id}$).
3) After analysing this release, RMT[i][j] is decremented in RPM, where 'i' is the ID of Resource Provider and 'j' is the ID of Resource Type, by performing the following
RPM[i][j] = RPM[i][j] – 1;
4) On updating this value, RPM[i][m+1], will be decremented by 1, as we know
RPM[i][m+1] = $\sum_{j=1}^{j=m} RPij$
5) After calculating new load of a resource provider and comparing it with threshold value, the RPT is updated as follows:
If (RPM[i][m+1] ≥ Threshold)
then,
STATUS [$RP_{id}$]= 1; //(declared Heavily Loaded),
LOAD[$RP_{id}$] = RPM[i][m+1]
Else if (RPM[i][m+1] = 0)
then,
STATUS [$RP_{id}$]= 0;
LOAD [$RP_{id}$] = 0;
Else
STATUS [$RP_{id}$]= -1; //(declared Lightly Loaded),
LOAD [$RP_{id}$] = RPM[i][m+1] ;
Endif
6) If the status of a Resource provider is either 'Idle' or 'Heavily Loaded', go to step 9.
7) If the status of a Resource provider is 'Lightly Loaded' (i.e. -1), find a resource provider with status Heavily loaded (i.e. 1), by performing the following steps:
For (l=1; l ≤ m; l++)
{
Set a variable FOUND=0;
If (STATUS[l]== 1)
then
assign variable FOUND=1;
Break;
}
8) If FOUND =1, transfer any one request from Resource provider with ID=l by performing following steps
RPM[l][j] = RPM[l][j] – 1;
RPM[i][j] = RPM[i][j] + 1;
Go to step 2.
Otherwise, wait till some resource is requested.
9) End

## IV. CASE STUDY

There are two variations of the proposed algorithm, namely Request Initiated and Release Initiated. To demonstrate their individual execution, two case studies are presented below.

*Case Study 1: To demonstrate the Request Initiated Algorithm*

Assuming a scenario, where there are 7 resource providers, say $RP_1$, $RP_2$, $RP_3$, $RP_4$, $RP_5$, $RP_6$, $RP_7$.There are 5 Resource Types, say $RT_1$, $RT_2$, $RT_3$, $RT_4$, $RT_5$. The predefined threshold value is assumed to be 20.

Then a sample RPM and RPT can be shown as below:

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 1 | 3 | 20 |
| $RP_2$ | 4 | 5 | 2 | 2 | 1 | 14 |
| $RP_3$ | 5 | 7 | 4 | 3 | 0 | 19 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 6 | 6 | 22 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Initial Resource Provider Matrix

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | 1 | 20 |
| $RP_2$ | -1 | 14 |
| $RP_3$ | -1 | 19 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 22 |
| $RP_7$ | -1 | 16 |

Initial Resource Provider Table

Let us assume that a new request is generated by Resource Requester $RR_{11}$, in the form of Resource Request Message ($RR_{11}$, $RT_2$, $RP_3$). After analyzing this request, and incrementing the request, updated RPM will be as follows:

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 1 | 3 | 20 |
| $RP_2$ | 4 | 5 | 2 | 2 | 1 | 14 |
| $RP_3$ | 5 | 8 | 4 | 3 | 0 | 20 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 6 | 6 | 22 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Resource Provider Matrix After Resource Request

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | 1 | 20 |
| $RP_2$ | -1 | 14 |
| $RP_3$ | 1 | 20 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 22 |
| $RP_7$ | -1 | 16 |

Resource Provider Table After Resource Request

After incrementing the request, the RPM [3][6] is equal to the threshold value, i.e. 20, therefore $RP_3$ becomes a heavily loaded provider, and designated as source node. Now some of the other resource provider is searched from RPT, and RP2 is found to be a lightly loaded provider or a sink node. Therefore, a load will be transferred from $RP_3$(source node) to $RP_2$(sink node). After transferring this load, the new RPM and RPT will be as follows:

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 1 | 3 | 20 |
| $RP_2$ | 4 | 5 | 3 | 2 | 1 | 15 |
| $RP_3$ | 5 | 8 | 3 | 3 | 0 | 19 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 6 | 6 | 22 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Resource Provider Matrix after Load Balancing

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | 1 | 20 |
| $RP_2$ | -1 | 15 |
| $RP_3$ | -1 | 19 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 22 |
| $RP_7$ | -1 | 16 |

Resource Provider Table after Load Balancing

*CASE STUDY 2: To demonstrate the Release Initiated Algorithm*

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 1 | 3 | 20 |
| $RP_2$ | 4 | 5 | 2 | 2 | 1 | 14 |
| $RP_3$ | 5 | 7 | 4 | 3 | 0 | 19 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 6 | 6 | 22 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Initial Resource Provider Matrix

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | 1 | 20 |
| $RP_2$ | -1 | 14 |
| $RP_3$ | -1 | 19 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 22 |
| $RP_7$ | -1 | 16 |

Initial Resource Provider Table

A resource is released from $RR_9$, so $RR_9$ broadcasted a message in the form of Resource Request Message ($RR_9$, $RT_4$, $RP_1$). After analyzing this release, and decrementing the allocated resource, updated RPM will be as follows:

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 0 | 3 | 19 |
| $RP_2$ | 4 | 5 | 2 | 2 | 1 | 14 |
| $RP_3$ | 5 | 8 | 3 | 3 | 0 | 19 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 6 | 6 | 22 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Resource Provider Matrix After Resource Release

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | -1 | 19 |
| $RP_2$ | -1 | 14 |
| $RP_3$ | -1 | 19 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 22 |
| $RP_7$ | -1 | 16 |

Resource Provider Table After Resource Release

After decrementing the values due to this release, the RPM [1][6] is now less than the threshold value, i.e. 20, therefore $RP_1$ becomes a lightly loaded provider, and designated as sink node. Now some of the other resource provider is searched from RPT, and RP6 is found to be a heavily loaded provider or a sink node. Therefore, a load will be transferred from $RP_6$(source node) to $RP_1$(sink node). After transferring this load, the new RPM and RPT will be as follows:

| | $RT_1$ | $RT_2$ | $RT_3$ | $RT_4$ | $RT_5$ | $\sum_{i=1}^{5} RTi$ |
|---|---|---|---|---|---|---|
| $RP_1$ | 5 | 4 | 7 | 1 | 3 | 20 |
| $RP_2$ | 4 | 5 | 3 | 2 | 1 | 15 |
| $RP_3$ | 5 | 8 | 3 | 3 | 0 | 19 |
| $RP_4$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $RP_5$ | 5 | 5 | 5 | 1 | 1 | 17 |
| $RP_6$ | 2 | 5 | 3 | 5 | 6 | 21 |
| $RP_7$ | 1 | 5 | 2 | 7 | 1 | 16 |

Resource Provider Matrix after Load Balancing

| $RP_{id}$ | Status | Load |
|---|---|---|
| $RP_1$ | 1 | 20 |
| $RP_2$ | -1 | 15 |
| $RP_3$ | -1 | 19 |
| $RP_4$ | 0 | 0 |
| $RP_5$ | -1 | 17 |
| $RP_6$ | 1 | 21 |
| $RP_7$ | -1 | 16 |

Resource Provider Table after Load Balancing

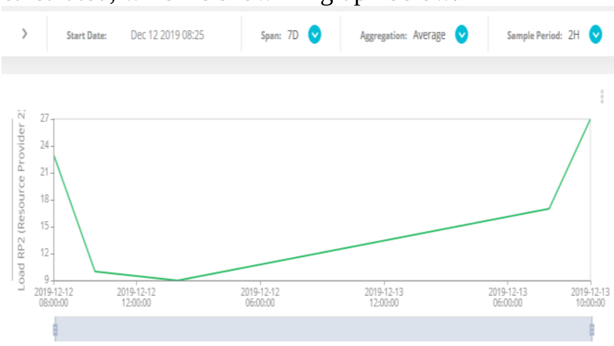
Both case studies were designed to demonstrate the implementation and two implementations of the proposed algorithm. In both cases, load is balanced by moving, with prior agreement, some incoming load from a heavily loaded provider (source node) to a lightly loaded provider (sink node), as well as mutual consent to a specific threshold value.

We could work in the future on a decisive factor, "how to pick the load to be transferred from source node to sink node." Another question that can be discussed is the fact that "when the algorithm is started."

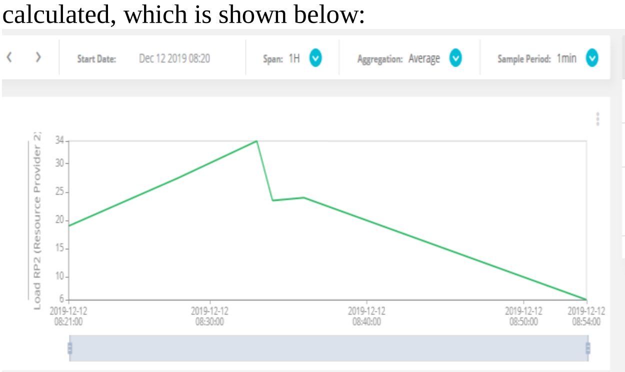
## V. RESULT ANALYSIS

The abovementioned algorithm when executed on a simulator gave some satisfactory results. Initially three resource providers were considered as RP1, RP2, and RP3. Their individual loads were fed using a slider widget. Initially Load at RP1, RP2 and RP3 are 30, 10 and 25 respectively. The load inputs for all the three resource providers can be shown as below:

Having the above loads, the average response time is calculated, which is shown in graph below:




After implementing SRAA, the final loads of RP1 RP2 and RP3 are2 4, 16 and 25 respectively, the response time is calculated, which is shown below:




This decrease in response time after the implementation of SRAA proves the success of SRAA in load distribution among various IoT nodes.

## VI. CONCLUSION

One can face many obstacles when building a smart city. Such challenges include technical, social, and finance as well as challenges in governance. Once a city is developed as a smart city, retaining its smartness while maintaining the quality of service is the main challenge.

This quality of service can be calculated with a decreased number of resources as an improved throughput. Therefore, when running a smart city, load management as well as load balancing is an unavoidable obstacle. Our research focused on a smart city's load balancing. Our main contribution is an algorithm for balancing the load in a smart city between different resource providers. The proposed SRAA (Sender-Receiver Agreed Algorithm) algorithm is based on the mutual consent of both heavily charged nodes and lightly charged nodes.

SRAA has two variations: Initiated Request and Initiated Release. With the support of all matrix and tables, both are explained separately. The algorithm modifications were also conducted with the aid of two case studies. The algorithm that is provided is suitable for a dynamic environment. We must focus on a decisive factor in the future, "How to select the load from the source node to the sink node." Another issue that can be addressed is the fact that "when the algorithm is launched."

## REFERENCES


1. Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems, 29(7), 1645-1660.
2. Chang, C., Srirama, S. N., & Buyya, R. (2019). Internet of things (IoT) and new computing paradigms. Fog and Edge Computing: Principles and Paradigms, 1-23.
3. Mijić, D., & Varga, E. (2018, August). Unified IoT Platform Architecture Platforms as Major IoT Building Blocks. In 2018 International Conference on Computing and Network Communications (CoCoNet) (pp. 6-13). IEEE.
4. Xu, B., Da Xu, L., Cai, H., Xie, C., Hu, J., & Bu, F. (2014). Ubiquitous data accessing method in IoT-based information system for emergency medical services. IEEE Transactions on Industrial informatics, 10(2), 1578-1586.
5. Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovic, N., & Meijers, E. (2007). Smart cities Ranking of European medium-sized cities, Centre of Regional Science, Vienna UT, October 2007.
6. Van Soom, E. (2009). Measuring levels of supply and demand for e-services and e-government: a toolkit for cities. Smart cities research brief; 3.
7. Lombardi, P., Giordano, S., Farouh, H., & Wael, Y. (2011, June). An analytic network model for Smart cities. In Proceedings of the 11th International Symposium on the AHP, June (pp. 15-18).
8. Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., and Oliveira, A., "Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation," Future Internet Lect. Notes Computer Science, vol. 6656, pp. 431-446, 2011.
9. Balasubramanian, K., & Cellatoglu, A. (2008). Improvements in home automation strategies for designing apparatus for efficient smart home. IEEE transactions on Consumer Electronics, 54(4), 1681-1687.
10. R.P. Dameri, "Searching for smart city definition: a comprehensive proposal," International Journal of Computers & Technology, vol. 11, no. 5, pp. 2544-2551,2013.
11. Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. IEEE Internet of Things journal, 1(4), 349-359.
12. Matta, P., Pant, B., & Arora, M. (2017, May). All you want to know about Internet of Things (IoT). In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1306-1311). IEEE.
13. Taivalsaari, A., & Mikkonen, T. (2017). A roadmap to the programmable world: software challenges in the IoT era. IEEE Software, 34(1), 72-80.
14. Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. Business Horizons, 58(4), 431-440.
15. Monzon, A. (2015, May). Smart cities concept and challenges: Bases for the assessment of smart city projects. In 2015 international conference on smart cities and green ICT systems (SMARTGREENS) (pp. 1-11). IEEE.
16. Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., & Shen, X. S. (2017). Security and privacy in smart city applications: Challenges and solutions. IEEE Communications Magazine, 55(1), 122-129.
17. Höjer, Mattias, and Josefin Wangel. "Smart sustainable cities: definition and challenges." In ICT innovations for sustainability, pp. 333-349. Springer, Cham, 2015.
18. Belissent, J. (2011). The Core of a Smart City Must Be Smart Governance. Forrester Research. Inc., Cambridge.
19. Elmaghraby, A. S., & Losavio, M. M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy. Journal of advanced research, 5(4), 491-497.
20. Mehmood, Y., Ahmad, F., Yaqoob, I., Adnane, A., Imran, M., & Guizani, S. (2017). Internet-of-things-based smart cities: Recent advances and challenges. IEEE Communications Magazine, 55(9), 16-24.
21. Botta, A., De Donato, W., Persico, V., & Pescapé, A. (2016). Integration of cloud computing and internet of things: a survey. Future generation computer systems, 56, 684-700.
22. Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J. R., Mellouli, S., Nahon, K., ... & Scholl, H. J. (2012, January). Understanding smart cities: An integrative framework. In 2012 45th Hawaii international conference on system sciences (pp. 2289-2297). IEEE.
23. Mohanty, S. P., Choppali, U., & Kougianos, E. (2016). Everything you wanted to know about smart cities: The internet of things is the backbone. IEEE Consumer Electronics Magazine, 5(3), 60-70.
24. Djahel, S., Doolan, R., Muntean, G. M., & Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. IEEE Communications Surveys & Tutorials, 17(1), 125-151.
25. Matta, P., & Pant, B. (2019). Internet-of-Things: Genesis, Challenges and Applications. Journal of Engineering Science and Technology, 14(3), 1717-1750.


## AUTHORS PROFILE

**Priya Matta** received her graduate degree from HNB Garhwal University, Post graduate from IGNOU and GEU, India in 2000, 2003 and 2010. She is currently pursuing Ph.D. in computer science and engineering with Graphic Era University, India. Her current research interests include Internet-of-Things and Information Security. She has 12 publications in top quality international conferences and journals. She is a member of several professional bodies, including CSTA (Association for Computing Machinery), IAENG and IEEE. She is a recipient of the Eminent Research Award (Computer Science & Engineering Engineering) of the Graphic Era University two times.

**Dr. Bhasker Pant** completed his graduate from M.B. Khalsa College, Indore, post graduate from Barkatullaha University, Bhopal, and doctoral studies at Maulana Azad National Institute of Technology, Bhopal, India in 2000, 2003 and 2012 respectively. He has done significant research work on Data Mining, Machine Learning, and Bioinformatics. His research interest includes Internet-of-Things and soft-computing too. He has 7 doctoral students and supervised more than 20 master's students. He has delivered invited talks, guest lectures and presented his research results in various counties like China, Singapore and India. He has more than 40 publications in top quality international conferences and journals. He is holding senior level academic positions the university time to time. He is currently the Dean (Research and Development) in Graphic Era University. He is a recipient of the Eminent Research Award (Computer Science & Engineering Engineering) of the Graphic Era University.

**Dr. Sachin Sharma,** Associate Dean, International Affairs and Associate Professor, Department of Computer Science and Engineering at Graphic Era Deemed to be University, Dehradun, UK, India. He is also Co-founder and Chief Technology officer (CTO) of IntelliNexus LLC, Arkansas, USA based company. He also worked as a Senior Systems Engineer at Belkin International, Inc., Irvine, California, USA for two years. He received his Philosophy of Doctorate (Ph.D.) degree in Engineering Science and Systems specialization in Systems Engineering from University of Arkansas at Little Rock, USA with 4.0 out 4.0 GPA and M.S. degree in Systems engineering from University of Arkansas at Little Rock with 4.0 out 4.0 GPA and He received his B.Tech. degree from SRM University, Chennai including two years at University of Arkansas at Little Rock, USA as an International Exchange Student. His research interests include wireless communication networks, IoT, Vehicular ad hoc networking and network security.