# From The Internet of Things to SELF Internet of Things

Horia A. Maior[1] and Shrisha Rao[2]

***Abstract*— The Internet of Things is a term to describe networked objects not traditionally thought of as computers (e.g., cars, household appliances) which may nonetheless be connected using Internet protocols and technologies (TCP/IP, etc.). Such "things" may also be connected, for control or communication, to the traditional Internet, and may themselves also be equipped with sensors or actuators to interact with their environments. It is envisioned that an Internet of Things will be useful in particular in resource-constrained systems (e.g., with smart grids). The naive approach to an Internet of Things would use a central controller or master node of some sort to oversee the activities of all "things" in the Internet of Things. However, this has obvious drawbacks, not the least being scalability. It is therefore desirable that a "thing" govern its own actions while achieving global "self" properties (such as self-adaptivity, self-stabilization) in an Internet of Things that has to work with resource constraints (e.g., limited allowable peak electricity usage for a domestic or industrial system of many "things"). Such an Internet of Things with self-governing "things" would not require a central controller, and addition or removal of "things" would be far easier. This paper provides a theoretical model of such Self-Internet of Things, giving a set of principles and properties to the system in respect to resource-constrain and energy efficiency systems. This would involve a proposed type of behaviour for a single "thing" in such an Internet of Things, as well as the principles or protocols by which such "things" are connected with one another.**

**Keywords:** Internet of Things, Distributed Systems, Resource Allocation and Management.

## I. INTRODUCTION

The Internet of Things (IoT) has been a highly discussed topic in he past 10 years within many areas of research, in all government, academic and industrial contexts because of its great prospect. It dates back to 1990's, precisely with the start of Industrial automation systems [1]. Called "... the third wave of the world information industry after the computer and the Internet ..." [2], IoT is a term used to describe a network of "things"; smart objects (devices) located in the physical world, not traditionally thought of as computers (e.g., cars, household appliances). They are equipped with some sort of sensors, and are normally connected with the world surrounding them (e.g. human beings, other objects etc.), being able to exchange information with other objects using Internet protocols and technologies (TCP/IP, etc.).

### *A. Self-governing, decentralized, extensible IoT, connected to a shared, variable power supply. Each thing is an object.*

hehe ...

[1]Horia A. Maior, Mixed Reality Lab, Faculty of Computer Science, University of Nottingham, United Kingdom. psxhama@nottingham.ac.uk

[2]Shrisha Rao, International Institute of Information Technology - Bangalore, India. srao@iiitb.ac.in

### *B. Sharing needs and deadlines between objects, provide them with a general view of the system. Higher priority objects with small power consumption are more likely to get powered, achieving a happy IoT*

One interesting aspect of the IoT is it's future impact on our daily life. Zorzi et al [3] points out there is a first time opportunity to interact with the surrounding environments and to exchange information that previously was not available by simply looking at objects (devices). Apart from a direct interaction individual-machine (object), it will also have a indirect effect through object-to-object interaction. Objects were not used to communicate and be influenced by other objects. Prasad and Kumar [1] describes this particular aspect of IoT as the advance version of Machine to Machine (M2M) communication, where objects are exchanging information between them without a human intervention.

The application of IoT is vast, covering many areas of research. One of it's major application area is environment monitoring within Smart Cities (with Traffic Management research, Smart Parking etc.), Smart Environment (with Air Pollution research, Fire Detection research etc.) and so on. Liu and Zhou [2] finds the features of automatic and intelligent objects of IoT suitable for monitoring environment information.

IoT will be useful in particular in resource-constrained systems (e.g. with smart grids). The naive approach to an Internet of Things would use a central controller or master node of some sort to oversee the activities of all "things" in the Internet of Things. However, considering a network that includes billions of objects interconnected, this has obvious drawbacks, not the least being scalability. Moreover, being composed of a single controlling unit, such system can carry at most one activity at any single moment.

It is desirable that a "thing" govern its own actions while achieving global "self" properties (such as self-adaptivity, self-stabilization) in an Internet of Things that has to work with resource constraints (e.g., limited allowable peak electricity usage for a domestic or industrial system of many "things"). Liu and Zhou [2] describes such property as "autonomy Features" of objects, where objects have the ability to reason, negotiate, understand, adapt and learn from other objects or environments. Such an Internet of Things with self-governing "things" would not require a central controller, and addition or removal of "things" would be far easier.

The specifications of our IoT system described above can be represented as a distributed system, where a central component is nonexisting, instead, all objects communicate with every other object in the system in order to coordinate their actions. They also have a processing capability and self properties being responsible to control own actions.

**Maybe** - text will be written about Self-Stabilization.

## II. Related Work

Here comes some related work

- Maximum Demand
- Total Demand
- Demand Responsiveness
- Demand Balance

## III. Problem Statement and The Model

In Section I we discussed and presented our interest and in the same time the focus of this paper. The following section, Section II, describes the existing work within the area of IoT from various points of view, and mentions the applications such IoT within many areas of research. In this section we will describe an abstract model of IoT from a resource-constrained point of view, formally stating the problem we address. At the end of the section, we present and prove our result. The next section will present ...

The problem in hand includes some sort of IoT within a resource-constrained system. We are particularly interested in energy consumption, and the use of IoT to help achieve energy efficient systems. This implies that the objects of the IoT system are consumers of energy, and by exchanging information between objects, we can achieve more energy efficient systems. This can be explained best with an example. Imagine a smart house where your smart toaster communicates to all other smart objects in your home that he wants to consume power at 7 am. If the power budget would be limited, all other objects would consider the toaster schedule and will act accordingly. We already pointed out that the naive approach of having a centralized system where all objects are controlled in some way by a central controller or master node would have many limitations. We decided to go beyond this approach and represent the IoT as a distributed system, where each object in the system govern its own actions.

### A. Definition and Notations

Let there be numerous objects $o_i$ that consume some energy. The set objects, denoted as $N$, together with some sort of power supply constitute the whole system $G$.

$N = \{o_0, o_1, ..., o_{n-1}\}$, the set of all $n$ objects of the system $G$.

Each object consumes a non-negative amount of energy power resource. To simplify the system, let us consider this non-negative amount of power a constant for each object in the system (an object cannot change its power setting). Let $\mathbb{R}^+$ be the set of positive real numbers, then let the demand function, $f : N \to \mathbb{R}^+ | f(o_i) = r$, gives the power demand of each object $o_i$. This means object $o_i$ demands $r$ amount of power.

The Total Demand of system $G$ at any one time, as described by Rao [4], is the sum of all power demands for all objects of system $G$:

$$\sum_{i=0}^{n-1} f(o_i)$$

The power supply of system $G$, is the total amount of energy available to share between all the objects in the system at any time. The main property if the power supply is that it is variable in time. If we think of solar energy, during the daytime there is more energy to consume, whereas during the night, the solar energy is close to none. Let the set $T$ denote the set of all time instances, and $\mathbb{R}^+ \cup \{0\}$ be the set of positive real numbers including 0. Then, the function $\gamma : T \to \mathbb{R}^+ \cup \{0\}$ denotes the power supply function. With $\gamma(t) = w$, $w$ the [4] the maximum resource limit, in our case energy power, available in the system $G$ at a given time $t$.

So far we could have distinguished between objects of $G$ in terms of their power demands. The question here is, what if the power supply cannot satisfy with power resource all objects of in the system? Rao [4] describes that in practice, distributed processes (objects in our case) drawing some sort of resource (power resource in our case), are distinguished from one another in terms of some sort of priorities. This is because some objects are more important than others in terms of their functions and utilities, and if the power supply cannot satisfy all the objects, priority will distinguish between the the high priority to low priority objects.

Let $\delta : N \times T \to \mathbb{R}^+ \cup \{0\}$ be the priority function, with $\delta(o_i, t) = p$, where $p$ is the priority of object $o_i$ at time $t$. As described, priority function is time dependent, that means objects can change priority in time.

### B. IoT as a Distributed System

- objects are networked
- they communicate
- learn about the system
- Alg 1 (exploration alg)
- Alg 2 (addition of new objects to the network)
- Alg 3 (removing objects from the network)
- proofs

In a distributed IoT, every object in the system $G$ is networked in such a way that is able to exchange information with all other objects (for simplicity, we make abstraction of how they are physically networked). It is also assumed that every object has some sort of computing capability (we assume that computing capability has no cost what so ever), and it is in some way 'self-aware' of its current 'needs' (for example each object knows or is able to find out about its power settings). Because we are building a de-centralized system, the decision-making comes to the object itself (this means that each object in the system decides on its own when is a suitable time to consume power resource), therefore, every object is also interested in an overview of the whole system (for example each object needs to know the maximum power limit $\gamma(t) = w$ but also each object has to know about what is its priority relative to other's in the network).

When creating such a system, initially each object in the system is "exploring" the whole system in some way. During the exploration, objects need to find out (and memorize in some way) other objects with the same priority, and in the same time find out other priorities in the system. This way, every object will have an general overview of what is their own priority relative to other object in the system. The exploration is possible trough some sort of communication and exchange of information between objects. Making abstraction of how the communication is made, let $v$ and $u$ be two objects of the system $G$; $v, u \in \{N\}$. Let $_m sg(v, u, m)$ be a message, where $v$ is the sender of the message, $u$ is the delivery object of the message and $m$ is the message (the message can contain any kind of information coming from the sender). Following Peleg's approach [5], a message can be also broadcasted or "flooded" to/over a network (in our case system). The following is an adapted algorithm from [5], for flooding a message across all $n$ objects of a system from a root object $o_j$:

**Algorithm 1: Algorithm Flood**

Let a source $o_j$

**for** $i \leftarrow 0$ ***to*** $n-1$ **do**
  **if** $i! = j$ **then**
    Send $msg(o_j, o_i, m)$
  **end**
**end**
**for** $i \leftarrow 0$ ***to*** $n-1$ **do**
  **Upon** receiving a message $o_i$
  - Store the message
  - Compute the message
  - Send acknowledgement

**end**

Considering Algorithm Flood above, we design an exploration algorithm within our distributed IoT system. The algorithm is run by all abjects in the system, all trying to communicate to other objects information about their priority and their power weight. In the same time, they receive compute and store information from other objects, in the end, every object having an overview of their position in the system.

Let $o_i$ be an object of $G$; $o_i \in \{N\}$. As described above, each object need to store information about other priorities in the system and objects of the same priority in the system. Let each object $o_i$ have to arranged lists:

- let $P_i =$ be an arranged list where each object $o_i$ can store priorities of other objects in decreasing order; such that $P_i(0)$ is the highest priority in the set.
- let $Q_i$ be an arranged list of objects of the same priority as $o_i$ in increasing order; such that $Q_i(0)$ is the object with smallest demand.

Before joining the system and running the exploration algorithm, every object $o_i$ lists are: $P_i = \{\emptyset\}$ and $Q_i = \{\emptyset\} \cup \{\delta(o_i, t)\}$. Please consider Exploration Algorithm below

**Algorithm 2: Exploration Algorithm** run by any object joining the system $G$

Let $time = t_0$
Let object $o_i$ run an Flood Algorithm over $G$
send: $msg(o_i, join, \delta(o_i, t_0), f(o_i))$

**for** $j \leftarrow 0$ ***to*** $n-1$ **do**
  Let $o_j$, $(j! = i)$ receive the message
  **Upon** receiving:
  **if** $\delta(o_i, t_0) = \delta(o_j, t_0)$ **then**
  - **Insert** $o_i$ in $Q_j$ **in order** of $f(o_i)$
  - Send acknowledgement $(\delta(o_i, t_0) = \delta(o_j, t_0))$

  **end**
  **else**
    **if** $\delta(o_i, t_0) \notin P_j$ **then**
    - **Insert** $\delta(o_i, t_0)$ in $P_j$ **in order**
    - Send acknowledgement $\delta(o_i, t_0) < \delta(o_j, t_0)$ (if so)
    - Send acknowledgement $\delta(o_i, t_0) > \delta(o_j, t_0)$ (if so)

    **end**
  **end**
**end**

We are describing an extensible IoT system, therefore new objects can join and objects that no longer want to be part of the system can leave at any time. When joining the system, the new object shall run the exploration algorithm. This way, other objects will make note of the new object's details (like priority and power setting), but in the same time, the new object will create it's overview of the system trough message acknowledgements from the existing objects. When an object is leaving the system it shall inform the other objects about it's intentions.

### C. Prioritized objects and a happy IoT

So far we have described the objects of $G$, having a power setting and some sort of priority. We described a distributed $G$, where each object communicates with all other objects in the system, and explores the system and finds out about other objects. However, we did not yet mentioned how the system is powered from the power supply, and how objects receive power resource.

We discussed that objects can have different priorities, and that is because some objects are more important in a way than others. Therefore, higher priority objects are more likely to get power resource than lower-priority ones. We also discussed that objects can have different power demands (settings). That is, one object might need more power resource than other ones. Using the following proposition, the system satisfies with power resource as many high priority objects as possible.

**Proposition 1.** *In our Self-Internet of Things model, for any amount of power resource available at any time ($\gamma(t) = w$), the higher-priority object, having the smallest power demand, will be powered first.*

Summarizing the system $G$, there are $n$ objects $o_i \in N$,

**Algorithm 3: Leave Algorithm** run by every object leaving the system $G$

Let $time = t$
Let the object $o_i$ leave the system
Flood Algorithm over $G$
send: $msg(o_i, leave, \delta(o_i, t), Q_i)$

**for** $j \leftarrow 0$ ***to*** $n-1$ **do**
    Let $o_j$, $(j! = i)$ receive the message
    **Upon** receiving:
    **if** $\delta(o_i, t) = \delta(o_j, t)$ **then**
        - **Remove** $o_i$ from $Q_j$
        - Send acknowledgement
    **end**
    **else**
        **if** $Q_i - \{o_i\} = \emptyset$ **then**
            - **Remove** $\delta(o_i, t)$ from $P_j$
            - Send acknowledgement
        **end**
    **end**
**end**

with $0 <= i <= n-1$. Each object has a power demand $f(o_i)$, a priority $\delta(o_i, t)$, and an overview of other objects priorities in the system (in decreasing order in list $P_i$; such that $P_i(0)$ the highest priority) and objects with the same priority (in increasing order of power demands in $Q_i$; such that $Q_i(0)$ is the smallest demand object of $o_i's$ priority). The power resource is the function $\gamma(t) = w$, representing the power budget of the system $G$ at every time $t \in T$.

## IV. Results, Discussion and Further Work

### A. Set of principles for a self- Internet of Things

- principle one - principle two - principle three etc

## V. Conclusions

blah blah

## References


[1] S. S. Prasad and C. Kumar, "An energy efficient and reliable internet of things," in *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on*. IEEE, 2012, pp. 1–4.

[2] Y. Liu and G. Zhou, "Key technologies and applications of internet of things," in *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, 2012, pp. 197–200.

[3] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 44–51, 2010.

[4] S. Rao, "A foundation of demand-side resource management in distributed systems," in *Transactions on computational science VIII*. Springer, 2011, pp. 114–126.

[5] D. Peleg, *Distributed computing: a locality-sensitive approach*. SIAM, 2000, vol. 5.


**Algorithm 4: Power Algorithm**

Let Time = $t$
$w \leftarrow \gamma(t)$
Let object $o_i$ run this code

**while** *true* **do**
    **if** $\delta(o_i, t) > P_i(0)$ *and* $f(o_i) = Q_i(0)$ *and* $f(o_i) <= w$ **then**
        **Power** object $o_i$
        $w \leftarrow w - f(o_i)$
        **if** $Q_i - \{o_i\} = \{\emptyset\}$ **then**
            $nextPriority \leftarrow P_i(0)$
            **Send** $msg(nextPriority, w)$
        **end**
        **else**
            $nextObj \leftarrow Q_i(1)$
            **Send** $msg(nextObj, w)$
        **end**
    **end**
    **Upon** receiving one of the msg:
    **Receiving** $msg(nextObj, w)$
    **Receiving** $msg(nextPriority, w)$
    **if** $(o_i = nextObj$ *and* $f(o_i) <= w)$ *or* $(\delta(o_i, t) = nextPriority$ *and* $f(o_i) = Q_i(0)$ *and* $f(o_i) <= w)$ **then**
        **Power** object $o_i$
        $w \leftarrow w - f(o_i)$
        **if** $Q_i - \{o_i\} = \{\emptyset\}$ **then**
            $nextPriority \leftarrow P_i(j) <=> (P_i(j-1) > \delta(o_i, t) > P_i(j))$ **Send** $msg(nextPriority, w)$
        **end**
        **else**
            $nextObj \leftarrow Q_i(j) <=> (o_i = Q_i(j-1))$
            **Send** $msg(nextObj, w)$
        **end**
    **end**
**end**