Some Challenging Issues for Internet of Things Realization

ZORAN BOJKOVIC, BOJAN BAKMAZ, MIODRAG BAKMAZ
University of Belgrade

Abstract: - From a wireless communication perspective, the Internet of Things (IoT) paradigm is strongly related to the effective utilization of wireless sensor networks (WSNs) and radio frequency identification (RFID) systems. IoT will provide a wide range of smart applications and services like remote health care monitoring, intelligent transportation, smart distribution, home automation, systematic recycling, etc. Generally, IoT represents intelligent end-to-end systems that enable smart solutions and, as such, it covers a diverse range of technologies. This paper seeks to provide some challenging issues for IoT concept realization. It starts with perspective embedded Internet protocol stack presentation. After that, some features of routing over low-power and lossy networks are emphasized. Also, concept of IP-based WSN approach to the IoT together with mobility management, global time synchronization, and security issues, is analyzed.

Key-Words: Embedded Internet, IoT, mobility, routing, WSN, 6LoWPAN.

1 Introduction
The term Internet of Things (IoT) was coined more than a decade ago by the Auto-ID Labs, as the leading global network of academic research laboratories in the field of networked RFID [1], where in parallel the concept of ambient intelligence and ubiquitous computing has been developed. Since then, there have been considerable developments in both academia and industry. The developments have primarily been dedicated to applying RFID technology to the logistics value chain. The IoT can be considered as a convergence among a number of heterogeneous disciplines. This multidisciplinary domain covers a large number of topics from technical issues (routing protocols, semantic queries), to a mix of technical, social and business issues (security, privacy, usability).

Pleasant user experiences are planned in the workplace and public areas as well as in the home environment by embedding computational intelligence into the nearby environment and simplifying human interactions with everyday service. Overcoming the technical challenges and socio-economic barriers of widescale IoT deployment requires a practical evaluation of corresponding solutions using interdisciplinary, multitechnology, large-scale, and realistic testbeds. The testbeds aim to design and deploy experimental environments that will allow [2]:

- The technical evaluation of IoT solutions under realistic conditions,
- The assessment of the social acceptance of new IoT solutions,
- The quantification of service usability and performance with end users in the loop.

Sensor widespread deployment represents significant financial investment and technical achievement. The data they deliver is capable of supporting an almost unlimited set of high value proposition applications. However, a main problem hampering success is that these sensors are locked into unimodal closed systems. Unlocking valuable sensor data from closed systems is a great task. Access to sensors should be opened such that their data and services can be integrated with data and services available in other information systems [3].

When it comes to wireless sensor technology, a variety of wireless sensor network (WSN) approaches such as ZigBee, and others proprietary solutions have been proposed. If a trillion things are connected through a single open standard interface such as IP, they become transparent as general hosts and servers supporting seamless connectivity, unique addressability and rich applicability. The research in this field focuses on how to build a fundamental architecture that enables the IP to be used in WSN environment [4].

At present, there is a clear need to develop a reference architectural model that will allow interoperability among different systems. With respect to the technological roadblocks, there is a need for action in three areas [5]:

a) An architectural reference model for the interoperability of IoT systems, outlining principles and guidelines for the design of its protocols, interfaces, and algorithms;

b) Mechanisms for the efficient integration of the architecture into the service layer of the future Internet networking infrastructure;

c) A novel resolution infrastructure, allowing scalable lookup and discovery of IoT resources, entities of the real world, and their associations.

This work starts with perspective embedded Internet protocol stack presentation. After that, IP-based WSN approach to the IoT will be emphasized. The relationship with mobility management, global time synchronization, as well as security issues will be described, too. Further research topics conclude the work.

2 Perspective Embedded Internet Protocol Stack

The widespread sensor deployment represents significant investment and technical achievement. A crucial problem hampering success is that sensors are typically looked into unimodal closed systems. To exceed this problem, sensor connection to the Internet and publishing outputs in well understood machine-processible formats on the web is indispensible.

Internet connectivity not only requires network level integration, but also application level integration to enable structured access to sensor data. To enable sensor automatic reasoning, these sensors, their outputs, and their embedding into the real world must be described in a machine-readable format that is compatible with data formats used to describe existing world knowledge in the web. Not only must the syntax and semantics of such a description be defined, but efficient mechanisms to annotate newly deployed sensors with appropriate descriptions are required [3]. Finally, the users wish to search for real-world entities by their current state. Such search requests refer not only to the output of sensors, but also to further machine-readable information that is available elsewhere in the web. The search engine needs to integrate these different data sources in a seamless manner.

Integrating resource-constrained sensors into the Internet is difficult since traditionally deployed protocols such as HTTP, TCP, or even IP are too complex and resource-demanding. To achieve integration, simple alternatives are required that can easily be converted from/to Internet protocols. Enormous progress has been made at the IETF over the past few years in specifying new protocols that connect smart objects to IP networks. The IETF has formed three WGs that define an adaptation layer (IPv6 in low-power WPAN - 6LoWPAN) [6], routing over low-power and lossy networks (ROLL) [7], and a resource-oriented application protocol (Constrained RESTful Environments - CoRE) [8]. Stack for wireless embedded Internet based on protocols proposed by corresponding IETF WGs is presented on Figure 1.

![Fig. 1. Perspective protocol stack for wireless embedded Internet.](image)

2.1 Adaptation layer

The adoption of IP by wireless embedded devices is challenging due to several reasons [9]:

- Battery-powered wireless devices require low duty cycles, whereas IP is based on always connected devices.
- Multicast is not supported natively in IEEE 802.15.4 [10] but it is essential in many IPv6 operations.
- Sometimes it is difficult to route traffic in multi-hop wireless mesh networks to achieve the required coverage and cost efficiency.
- Low power wireless networks have low bandwidth (20-250 kbit/s) and frame size (IEEE802.15.4 packets are rather small, 127 bytes maximum at the physical layer, minus MAC/security and adaptation layer overhead). On the other hand, the minimum datagram size that all hosts must be prepared to accept, for IPv6 is 1280 bytes. IPv6 requires that every link in the Internet has a maximum transmission unit (MTU) of 1280 bytes or greater. On any link that cannot convey a 1280-byte packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.
- Standard protocols do not perform well in LoWPAN. For example, TCP performs very poorly in wireless networks due to its
inability to distinguish between packet losses due to congestion and those due to channel error.

Due to the fact that IPv6 represents backbone of NGN [11], the 6LoWPAN WG was chartered to standardize necessary adaptations of this network protocol, for systems that use the IEEE 802.15.4 PHY layer, and has defined how to carry IP datagrams over IEEE 802.15.4 links and perform necessary configuration functions to form and maintain an IPv6 subnet. 6LoWPAN represents a lightweight IPv6 adaptation layer allowing sensors to exchange IP packets [12], [13]. Core protocols for 6LoWPAN architecture have already been specified and some commercial products have been launched that implement this protocol suite.

2.2 Routing over low-power and lossy networks

The IETF has significant experience in IP routing and it has specified a number of routing protocols over the past two decades (RIP, OSPF, etc.). On the other hand, routing in networks made of smart objects has unique characteristics. These characteristics led to the formation of a new WG called ROLL whose objective is to specify a routing protocol for low power and lossy networks (LLNs), known as RPL [14]. LLNs are formed by smart objects with limited processing power, memory and energy. Unlike the MANET routing protocols, which perform well for adhoc networks, RPL is optimized for upstream and downstream routing (to/from a root node), a paradigm appropriate for networks connected to the Internet. This routing protocol is essential for the deployment of IoT, since it enables traffic forwarding between low-power devices and the Internet. It has been designed assuming that the LLN scan comprises up to thousands of nodes interconnected by unstable links. Furthermore, RPL has been designed to operate over a variety of link layers such as IEEE 802.15.4, and it is a typically distance vector IPv6 routing protocol.

A Directed Acyclic Graph (DAG) [14] is a directed graph having the property that all edges are oriented in such a way that no cycles exist. All edges are contained in paths oriented toward and terminating at one or more root nodes (traditionally called sinks in WSNs). RPL routes are optimized for traffic to or from one or more roots (sinks). As a result, RPL uses the DAG topology and is partitioned in to one or more Destination Oriented DAGs (DODAGs), one DODAG per sink. RPL specifies how to build the DODAG using an objective function. The objective function computes the optimal path according to certain routing metrics and constraints. In this way, DODAGs with different characteristics can be formed. For example, different DODAGs are constructed with the objective to (1) find the optimal path in terms of link throughput while avoiding battery-operated nodes or (2) find the optimal path in terms of latency while avoiding non-encrypted links. There can be several objective functions operating at the same node depending on the different path requirements of a given traffic. In this way, it is possible to have multiple DODAGs active at the same time to carry traffic with different requirements.

RPL specifies local and global repair mechanisms for recomputing routes when an inconsistency is detected or based on administrative decisions [15]. Local repair means detaching a node’s sub-DODAG by increasing its rank value. Once a root initiates a global repair event, all the nodes in the DODAG recompute their rank values and reconfigure their parent sets. An example topology of a RPL network together with non-storing and storing modes is presented in Fig. 2.

![Fig. 2. Example of RPL nodes that form a directed acyclic graph rooted at a destination node support multipoint-to-point traffic.](image)

Solid arrows represent each node’s preferred parent (determined from the node’s neighbors and their rank values) while dotted arrows point to the other nodes in the parent set. In order to support
routing to various destinations within the DODAG which are the root, RPL uses the Destination Advertisement Object (DAO) message. RPL supports scenarios where in-network nodes do not have enough memory to store routes to all possible destinations. In this case, the DAO messages, which contain information on the desired parent set of a destination node, are propagated up the DODAG until they reach the root. The root gathers DAOs from all nodes in the DODAG, and uses them to construct "down" routes to various destinations. Data to these advertised destinations is forwarded along a DODAG until it reaches the root, which then attaches a source routing header and sends it back down the DAG. Alternatively, nodes in the DODAG may store next-hops to downstream destinations. However, a key design simplification was not supporting "mixed-mode operation" in which storing and non-storing nodes coexist, since this was still considered a research issue. Thus, all nodes in a DODAG must either store or not store routes.

2.3 Application protocol

In 2010, the IETF started a new WG so-called CoRE [8] with the aim to extend the web architecture to even the most constrained networks and embedded devices. Today’s web protocols work well between servers and clients running on PCs and handheld devices. However, constrained LLNs often mean high packet loss (5-10%), frequent topology changes, low throughput (10-20 kbit/s), and useful payload sizes that are often less than 100 bytes. Embedded devices typically depend on cheap microcontrollers with processors running at several MHz and limited memory. In addition, the interaction patterns in M2M applications are different, often requiring multicast support, asynchronous transactions, and push rather than pull. The CoRE WG has been chartered to develop a new Web transfer protocol and appropriate security setups for these M2M applications over constrained networks and nodes [16].

The WG is at present completing work on the Constrained Application Protocol (CoAP) [17]. It provides a highlight alternative to HTTP using a binary representation and a subset of HTTP’s methods (GET, POST, etc.). In addition, CoAP provides some transport reliability using acknowledgments and retransmissions. For seamless integration, reverse proxies may convert 6LoWPAN to IPv6 and UDP/CoAP to TCP/HTTP so that sensor data can be accessed using these omnipresent protocols. The integration of sensors into the Internet using CoAP/HTTP already enables many applications in which developers query and process data provided by a known set of sensors. A machine understandable description of sensors and generated data are required. Semantic web technologies fulfill this requirement as they enable machines to understand, process and interlink data using structured descriptions of service [18]. Linked open data as the framework makes this integration both immediate and meaningful through the inclusion of semantic links into a resource’s machine-readable description.

The use of UDP as transport protocol and the reduction of the packet header size, significantly decrease the power consumption in IoT. In order to evaluate the CoAP performance improvement compared to the HTTP, a simple experiment can be performed. In [19] a series of web service requests first between a CoAP client/server and then between an HTTP client/server system are generated. CoAP system is based on previously described protocol stack. Table 1 illustrates the results of the comparison between CoAP and HTTP in terms of bytes transferred per transaction and power consumption. It should be noted that the results have been taken in steady state conditions.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bytes/transaction</th>
<th>Power consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP</td>
<td>1451</td>
<td>1.333 mW</td>
</tr>
<tr>
<td>CoAP</td>
<td>154</td>
<td>0.744 mW</td>
</tr>
</tbody>
</table>

An HTTP transaction has a number of bytes nearly 10 times larger than the CoAP transaction. This is a consequence of the significant header compression executed in CoAP. In fact, CoAP uses a short fixed length compact binary header of 4 bytes and a typical request has a total header of about 10-20 bytes. After being encapsulated in the UDP, 6LoWPAN and MAC layer headers, the CoAP packet can be transfer into a single MAC frame which has a size of 127 bytes. It is straightforward that the higher number of bytes transferred in an HTTP transaction implies a more intensive activity of the transceiver and CPU and consequently higher power consumption. The battery lifetime is unrealistically short in both cases as a consequence of the high number of client requests generated during the experiment. It is worth underlining that the results presented in this example do not exhaustively compare the two protocols. The simple experiment presented is only intended to illustrate how the UDP binding and the header compression introduced in CoAP improve the power consumption of IoT.
3 Wireless Sensor Networks and Internet of Things

There have been many research projects in the field of IP-based WSN approach to the IoT from a dedicated IP stack for low-processing-power microprocessors to real deployments. Since the development of a Micro IP (μIP), an open source TCP/IP stack capable of being used with tiny microcontrollers for smart sensors, a few approaches were carried out on top of TinyOS and ContikiOS.

There are a wide range of technologies that will be involved in building the IoT. The enhancement of the communications networks infrastructure, through heterogeneous technologies, is essential as well as adoption of IPv6 in order to provide a unique address to each thing connected to the network. The technologies that allow the location and identification of physical objects will also be basic in this context. WSNs are able to provide an autonomous and intelligent connection between the physical and virtual worlds. Focusing on this type of networks, a particular important challenge is the creation of an E2E secure channel between remote entities [20]. Therefore, it is necessary to allow the elements of a WSN, to connect other entities through the Internet. An increase in research efforts has lead to maturity in this field, yet there seems to exist some gaps to be filled.

As for protocols, they should be carefully designed in terms of the trade-off between interlayer independence and optimization. Some protocols may vertically locate through layers for optimization. Other protocols may stay beyond the adoption layer for protocol independence. The Sensor Networks for an All-IP World (SNAIL) [4] approach to the IoT protocols were designed to comply with IETF RFC 4944 [12]. The SNAIL adaptation layer includes all the necessary frame formats and operations, and is fully compliant with standards related to header compression, addressing, fragmentation and reassembly, etc. To complete the IPv6 adaptation, it employs important protocols that are not specified in the standard. To obtain IPv6 address in the start-up process a bootstrapping network protocol is proposed. In this protocol, auto-configuration is completed by combining a network prefix. This prefix is obtained from a neighbor discovery (ND) message from a response node with an interface identifier from an association process between two IEEE 802.15.4 nodes. The bootstrapping process, a joining node registers its information, which includes an IP address, 64-bit extended unique identifier (EUI-64), type of routing protocol, sensors, as well as service for further network management by the gateway. Also, the registration information is used to advertise the gateway’s liveness and the network prefix.

In order to serve rich web content and to reduce the overhead of sensor nodes with no sacrifice of interoperability, the Distributed Resource-Based Simple Web Service (DRESS-WS) is designed [4]. It uses HTTP over TCP and distributes traffic to presentation servers and sensor nodes. As for web content, there exists two types of web content: real data and presentation templates. The real data are dynamic, while the templates are static and shareable accounting for high traffic intensity. Only real data are served by nodes that host a web server, while the templates are served by presentation servers.

3.1 Mobility management

Mobility management is one of the most important research issues in 6LoWPAN based networks. Because, typical mobility protocols are generally targeted for global mobility, they introduce significant network overhead in terms of increasing delay, packet loss, and signaling when mobile nodes change their point of attachment very frequently within small geographical areas. Having in mind that fact, methods for reducing handover delay are therefore essential for the IoT. The mobility protocol should be supported to a lightweight fashion, because a thing’s mobility behavior directly inherits characteristics of portable devices.

SNAIL architecture uses a novel mobility management protocol called MARIO, which stands for Mobility Management Protocol to Support Intra-PAN and Inter-PAN Handover with Route Optimization for 6LoWPAN [21]. The design of MARIO is based on MIPv6 and a fast and seamless handover scheme.

In the case of the inter-MARIO handover procedure illustrated in Fig. 3, when a partner node detects MN’s movement, it sends preconfiguration message with MN information. The information is stored by foreign agents (FAs) due to resource limitations. The partner node also gives the MN information, such as channel information, about the neighbor PAN. When orphaned, the MN can use the channel information to selectively scan a channel. When the MN associates with the new PAN, the FA performs a surrogate binding update (BU) simultaneously with the MN’s IP operations, as in the case of the care-of address (CoA) generation. With this operation, the home agent (HA) creates a binding for the home-of address (HoA) of the MN to the FA. After the process of joining the new PAN
Fig. 3. Inter-MARIO handover procedure [21].

is completed, the MN sends a binding update to the FA with a binding for the MN’s HoA to the MN’s CoA. This binding operation brings the MN to the end of the handover procedure. Due to the handover preconfiguration and surrogate BU, MARIO can reduce the channel scan delay, the layer 2 association delay, and the binding message exchange delay. MARIO also provides an additional benefit to the solution of the route optimization problem.

This procedure does not provide an optimal route from the corresponding node to the MN, since packets addressed to the MN may be delivered to the HA and then it is forwarded to the new location of MN. To solve this problem, also known as triangle routing, the MIPv6 standard proposes a route optimization method which is the Return Routability (RR) procedure [22].

### 3.2 Global time synchronization

Network Time Protocol (NTP) and Simple NTP (SNTP) are the most widely used application level time protocols based on E2E communication. Also, they are unlikely to be used for resource-limited multihop-based networks because E2E communication for time correction causes substantial overhead. This overhead is due to the necessity for a number of periodic control messages to synchronize from each node.

Several time synchronization protocols and schemes are proposed for WSNs in the open literature [23]. These protocols only focus on clock synchronization, not global time synchronization. This limitation is due to the fact that the protocols are tightly coupled with energy conservation in the MAC layer. Thus, the synchronization process is often executed only between neighbors.

To ensure the consistency and precision of information, the IoT requires globally synchronized time. A multihop time synchronization protocol called 6LoWPAN Network Time Protocol (6LNTP) is proposed in [4]. This protocol has two phases (Fig. 4):

1. gateway synchronization phase, which involves the use of the NTP, and
2. LoWPAN synchronization phase which adjusts the time to the reference time from the gateway.

In the first phase, the 6LoWPAN gateway corrects the time by exchanging request and response packets with an NTP server. While in the second phase, the nodes synchronize their time with the gateway that synchronized with an NTP server.

### 3.3 Security issues

In order for WSN to become a part of the IoT it is necessary to consider from the adaptation of existing network standards to the creation of interoperable protocols and the development of supporting mechanisms for composable services. One of the challenges is security, mainly because it is not possible to directly apply existing Internet-
centric security mechanisms due to the nature of WSN. The relevant security challenges are related to the integration of WSN within the IoT. Although these challenges are tightly related to WSN, they can also be applicable to other relevant technologies of the IoT (e.g. embedded systems, mobile crowdsensing, etc.). Even if a WSN itself is protected with its own security mechanisms (e.g. using the link-layer security), the public nature of the Internet requires the existence of secure communication protocols for protecting the communications between two peers.

Sensor nodes can make use of the 6LoWPAN protocol to interact with IPv6-based networks. They are powerful enough to implement symmetric key cryptography standards such as AES-128. Due to the power constraints and limited computational capabilities of the nodes, at present there is no explicit support for the IPsec protocol suite in 6LoWPAN. As a consequence, it is necessary to study how other mechanisms can be used in order to create an E2E secure channel.

The creation of secure channels is just one of the steps in the creation of a securely integrated WSN [20]. In order to avoid unauthorized users to access to the functionality of the WSN, authentication and authorization mechanisms must be developed. Also, creation of suitable and scalable identification mechanisms that can provide unique and virtual identifiers to all the different network elements is important. Finally, the survivability problem of IP-based WSN must be considered.

Other important challenges in this particular field are the integration of security mechanisms and data privacy [24]. The security of the IoT from a global perspective, regarding information must be considered. Even if different wireless technologies are secure by their own, their integration will surely generate new security requirements that must be fulfilled. Also, it is necessary to analyze how the security mechanisms that protect one single technology, will be able to coexist and interact with each other.

### 4 Conclusion

The characteristics of the IoT bring new challenges in developing applications and make the existing challenges in the area of ubiquitous computing considerably tougher. Existing semantic sensor web technologies enable the integration of sensors into the Internet. As it was difficult to foresee the wealth of current web applications back when the web was created, we have to see how people will use the Semantic Web of Things. Also, at present it is hard to predict it as broadly adopted as the web is. Using sensor data is clearly beneficial, as then integration with knowledge from arbitrary services is possible. For example, sensor data can be linked to geographic data, user-generated data, scientific data, etc. A strong indicator of whether this line of development will be successful in the long run, is
provided also by the exponential growing amount of linked data.

An important and big step toward the IoT would be to facilitate suitable IP-based WSN technologies to support the network of things. An increase in research efforts has led to maturity in this field, yet there seems to be gaps to be filled because of the focus on how to adopt the IP to the space of things. Considering the impact of billions of new inter-networked devices, the emerging IETF protocol stack for IoT should be cornerstone for research in this field.

Although the IoT concept has been around for several years now, there are still many crucial issues that have not been solved, including heterogeneity, scalability, security, etc. The complexity of these technical issues, especially in view of the resource-constrained nature of many components and of the use of wireless communications, calls for a unified architectural view able to address them in a coherent fashion.

References:

[10] IEEE Std. 802.15.4-2006, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), Sep. 2006.