Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 594 – 601

Procedia Computer Science

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

A DNS Architecture for the Internet of Things: A Case Study in Transport Logistics

Bill Karakostas

School of Informatics, City University London, UK

Abstract

This paper proposes a DNS architecture for the Internet of Things (IoT). Similarly to the existing DNS infrastructure on the Internet, the DNS for the IoT translates unique identifiers (URIs) of physical objects to concrete network addresses, from which information about such objects (e.g. status, location) can be extracted. We propose an experimental DNS infrastructure for IoT, in the domain of transport logistics. We have simulated the behaviour of the DNS infrastructure using a 3 tier hierarchy of domain name servers and a three level caching strategy. Preliminary results indicate that a DNS approach could be a feasible proposition to realise object tracking in IoT.

Keywords: DNS; Internet of Things; DNS caching strategy; global object identifier; transport logistics

1 Introduction

Internet and the Web have made it possible to search online information about people, places and events, using search engines and online directories. A natural extension of this is the ability to search for *any* physical object, no matter where that object is located, by using simple URLs such as
www.tracking.com/findmyobject?id=1234567

The main obstacle to attain such capability, is the lack of globally unique identifier for objects. Unlike people or places, objects we trade and use every day, rarely have a truly universal identifier. To be able to resolve a URL like the one used in the example above, we would need to (a) know the globally unique (or at least unique within the *tracking.com* domain) identity of the object we want to track and (b) find a server that can resolve the URL to a physical network address that possibly leads to a record containing information about the object we are inquiring about.

The Internet of Things (IoT) paradigm proposes to connect all physical objects in a global Internet based infrastructure to exchange information and communication. IoT aims to support intelligent identification, location, tracking, monitoring and management. IoT is based therefore, on the integration of several communications solutions, identification and tracking technologies, sensor and actuator networks, and distributed smart objects [2].

Within the IoT scope, several proposals for unique object identifiers have been made. *Ucode* for example is an identification number system that identifies objects and places in the real world uniquely [14]. Information can be associated with objects and places, and the associated information can be retrieved by using ucode.

doi:10.1016/j.procs.2013.06.079

Defining and assigning (locally) unique identifiers for man-made objects such as products, and object types has traditionally been the concern of the manufacturing and logistics industries. From the use of barcodes, to the more recent RFID technology, the industry has come up with several unique object identification schemes. Globally unique identifiers has been proposed for logistics applications by standardisation bodies such as GS1, based on classification schemes such as EAN, UCC, and JAN, to identity the type of products of individual vendors. However such codes are unique at the level of object type, not instance; for example, the same product code can be assigned to two packages of the same products. In contrast, *ucode* is a code to identify individual objects, unlike existing product code to display product types.

However, tracking logistics objects such as parcels requires identifiers that are unique (at least) within the scope of the logistics chain, i.e. across locations such as the factory, warehouse, distribution centre, and hence across participants' systems. Unique identifiers used for tracking purposes are routinely employed by logistics companies, as the example email notification below illustrates:

items have been despatched via our courier Hermes in parcel number 0359130888584

Although in the example above, parcel number *0359130888584* is not globally unique; it is unique within the context of the system used for tracking the parcel. This, however, is not quite IoT scale capability, as information about tracked objects is confined within a limited number of systems and cannot easily be shared on a broader basis, across the whole transportation chain.

For tracking information to be useful therefore, each logistics object must be ascribed a globally unique identity. Such identity must serve as a key for obtaining information about the object and must be preserved as the object moves across physical locations and hence computer networks. As argued above, the problem with current approaches and standards for logistics object identification and tracking is that, once such information is captured by the tracking systems of the logistics company, it is usually confined within single systems, or shared with only a small group of partners or customers.

To obtain true Internet scale object tracking capabilities in an IoT setting, there is a need to expose the object information to a public network that utilises the mechanism of domain name servers for establishing and maintaining the association between an object's identifier and its network address. In short, we need a Domain Name Server (DNS) architecture for IoT, which is exactly what is advocated by this paper.

To summarise, this paper proposes a hierarchical organisation of domain name servers, and a DNS query resolution mechanism for IoT, that can scale to Internet levels. More specifically, the paper:

- proposes an estimate of the size of an Internet-scale DNS network for IoT
- describes the implementation and simulation of a scaled down DNS network and its performance (in terms of DNS errors and cache failures) in tracking logistics objects
- finally, suggests factors, technical and organisational, that could influence the establishment of a global scale DNS for the Internet of things.

The rest of the paper is structured as follows.

Section 2 provides the background in the main subject areas investigated by this paper, i.e. the DNS system, unique object identification schemes, and the concept of object tracing and tracking in transport logistics. Section 3 first proposes a DNS architecture for IoT in transport logistics, and then provides an estimate for the scale of this endeavour, in terms of Domain Name Servers, numbers of objects, object tracking queries and network traffic volume for an IoT for transport logistics. Section 4 proposes a three tiered DNS architecture for the IoT, while Section 5 presents a simulation of a subset of the architecture, used to estimate its performance (in terms of DNS query hits and failures) under realistic conditions. Finally, Section 6 identifies areas for further research and standardisation required for the commercial adoption of the proposed approach.

2 Background Concepts and Definitions

As the approach proposed in this paper is based on two core technological areas (DNS and unique object identifiers) and an application domain (transport logistics), this section surveys the main concepts involved as well as existing research.

2.1 Unique Identification Schemes and Standards

Unique object identification schemes for industrial usage have been proposed by bodies such as GS1, an international non for profit standards organisation that defines and promotes standards to improve visibility and traceability across supply chains. One of the GS1 standards is the Serial Shipping Container Code (SSCC), a

unique identifier for individual logistic units. With the use of SSCC, logistics units can be tracked individually throughout the supply chain.

The Electronic Product Code (EPC) is a universal identifier based on Universal Resource Identifiers (URIs). EPC provides a unique identity for every physical object anywhere in the world, for all time. EPC structure is defined in the *EPCglobal Tag Data Standard*, which is an open standard [5].

The Object Name Service (ONS) [5] is an automated networking service similar to the Domain Name Service (DNS) discussed in the next section. When an interrogator reads the RFID tag of a product, the Electronic Product Code is passed to a middleware, which, in turn, queries an ONS on a local network or the Internet to find where information on the product is stored. The result of that query is a server where a file about that product is stored. Information about the product in the file can be forwarded to a company's inventory or supply chain applications.

2.2 Domain Name Service (DNS)

DNS is an Internet wide infrastructure level Internet service used for the discovery of information about a domain name and for mapping a host name to an IP address. The DNS has three major components [12] [13] :

- The *domain name space and resource records* , which are a tree structured name space and data (such as network addresses) associated with the names. The DNS for the Internet uses some of its domain names to identify hosts. A query based on a domain name can return Internet host addresses.
- *Name servers*, i.e. servers that hold information about the domain tree's structure and set information. Name servers know the parts of the domain tree for which they have complete information; a name server is said to be an *authority* for these parts of the name space.
- *Resolvers* that are client applications that extract information from name servers in response to client requests.

2.3 IoT and Transport Logistics

It has been argued that in the future, IoT will play an increasingly important role in transport logistics [16]. The subset of IoT that we consider in this paper is the universe of all man-made objects that are both transported in logistics, as well as used to transport other objects. This includes all products that are manufactured or processed in some way, and then moved from an origin to a destination, by a transport logistics chain. We call such products and transportation materials and equipment *logistics objects*. Many logistics objects can be combined in a *logistics unit load* which is a combination of units such as products (goods) packaged together in a carton, case or pallet that need to be managed through the transportation chain (Figure 1).

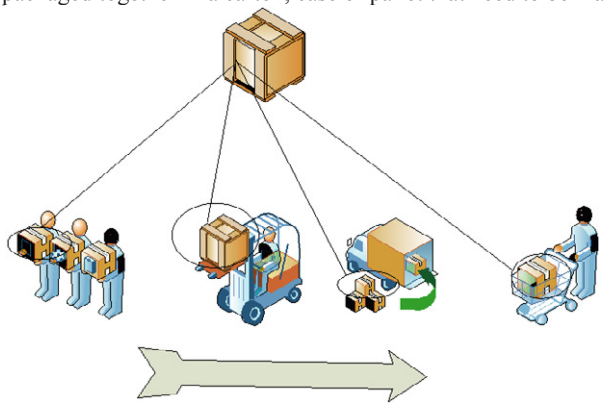

Fig. 1: Tracking a logistics object through the transportation chain.

Logistics equipment is another type of logistics object that is used for transporting other logistics objects, including other logistics equipment objects. For example, a shipping container is a logistics equipment object that can carry other logistics objects such as cartons or pallets.

Figure 1 shows several types of logistics packaged in boxes (logistic units), which are in turn packaged in pallets (other logistic units) transported by logistics equipment such as forklift trucks and delivery trucks.

Most ordinary logistics objects lack automated identification and communication capabilities, i.e. they are more likely to have a barcode rather than an RFID chip attached to them. These days, however, many types of logistic equipment have both automated identification (for example through active or passive RFID), as well as communication capabilities (for example over mobile telephony networks, using embedded radios) that allow them to communicate with other systems on closed networks or even over the Internet by using gateways acting as Internet proxies [10].

Network capable logistics objects are assigned network addresses by the network they belong to, where a network gateway acts as a proxy for translating between internal object addresses and network addresses. As logistics objects are naturally, mobile, they frequently change associations with gateways as they move across different networks, effectively changing their network address.

Essentially, tracking logistics objects is the problem of knowing of which network they are part of at any time, and thus, knowing the server that holds information about the logistics object. As the object searcher will often have just the object identifier as the only information on which to base the search upon, a DNS system for logistics objects is needed to make the discovery of an object's address at any time, possible, and through that to obtain information about the logistics object.

As we argue in the rest of this paper this network of DNS server has an architecture that is similar to the current DNS of the Internet. One of the differences between the DNS for logistics objects and the current DNS is that the authoritative servers of the former cannot be fixed, due to the mobile nature of the logistics objects. As it will be explained in subsequent sections, the time to live (TTL) of SRVs varies from a few hours to several days or even weeks for inter-continental journeys.

3 Organisation and structure of DNS Network for the logistics industry.

Similarly to the current Internet DNS, the DNS system for logistics objects needs to have a hierarchical organisation in order to cope with the size of the domain. Many arrangements for domain name assignments are possible, and need to be based both on technical as well as business considerations. We propose that such decisions are best taken by an international body. A possible DNS server hierarchy for logistics, could for example involve a top level (Level 0) DNS server managed by an international body like UN/CEFACT, that is also responsible for domain name harmonisation. Large logistics providers could be responsible for the DNS backbone infrastructure, by having Level 1 domain names allocated to them. Lower level subdomains could correspond to the business organisation of the logistics companies. For example, large hubs and distribution units could be assigned tier 2 subdomains, while smaller installations such as container yards, warehouses, etc., would get assigned progressively lower level subdomains.

According to this approach and in line with the IoT vision presented at the start of this paper, a query sent to the top level server could be formed as *TopDomainName/serviceofobjectrequired/?ObjectID=ID*
where *ID* is the unique identifier of the tracked logistics object.

This query would need to be resolved to a fully qualified URL such as
TopDomainName/Subdomain/sub-subdomain/.../serviceofobjectrequired/?ObjectID=ID
by forwarding it to authoritative servers for each sub-subdomain.

Correspondingly, Level 0 servers would handle name queries about Level 1 servers, while Level 1 servers would handle DNS queries that correspond to the tracking queries handled today by the web sites of the large logistics companies. Level 1 servers would then delegate the queries to the appropriate subnet of Level 2 servers, depending on the type of the logistics object been queried. Level 2 (and lower) servers would be responsible for resolving the query by tracing the object to the appropriate logistics location or equipment. Ultimately, the lowest tier servers would have to interact with the various gateways that act as proxies for the logistics objects. The gateway acts therefore as the authoritative servers for logistic objects DNS queries. They would supply the *service record* (similar to the SRV defined in [11]) for each object. SRVs are further discussed in following sections.

4 Size of an Internet Scale DNS Network

The design of a suitable DNS architecture for IoT must take into account the size of the real life network, in terms of aspects such as the expected number of DNS queries. While it is not easy to obtain a global, accurate figure for the total number of the logistics objects handled by the industry, an estimate can be extrapolated from data provided by large logistics companies. UPS (URL: http://www.ups.com) for example, is a large logistics service provider that receives 26.2m tracking requests daily, which equates to 1.1 million requests per hour or 300 requests per second.

As UPS represents roughly 50% of the US domestic market for small parcels, a total of in excess of 50m tracking requests per day (for small parcel tracking) could be expected, in the US alone. In the DNS architecture proposed here, that number would need to be multiplied by a factor of 5 to 10, due to the number of queries by DNS resolvers, including primary queries as well as follow ups because of timeouts, DNS errors etc. The total Internet traffic of 500 million queries per day (in the US alone, and for small parcel tracking only) is comparable, for example, to Google's average of 3 billion queries per day (according to http://searchengineland.com/by-the-numbers-twitter-vs-facebook-vs-google-buzz-36709).

In turn, this large number of queries would generate a significant amount of network traffic which (assuming an average 1kB per query/response), could potentially reach levels of several petabytes per day. That traffic estimate excludes the internal traffic generated by the synchronisation of DNS servers.

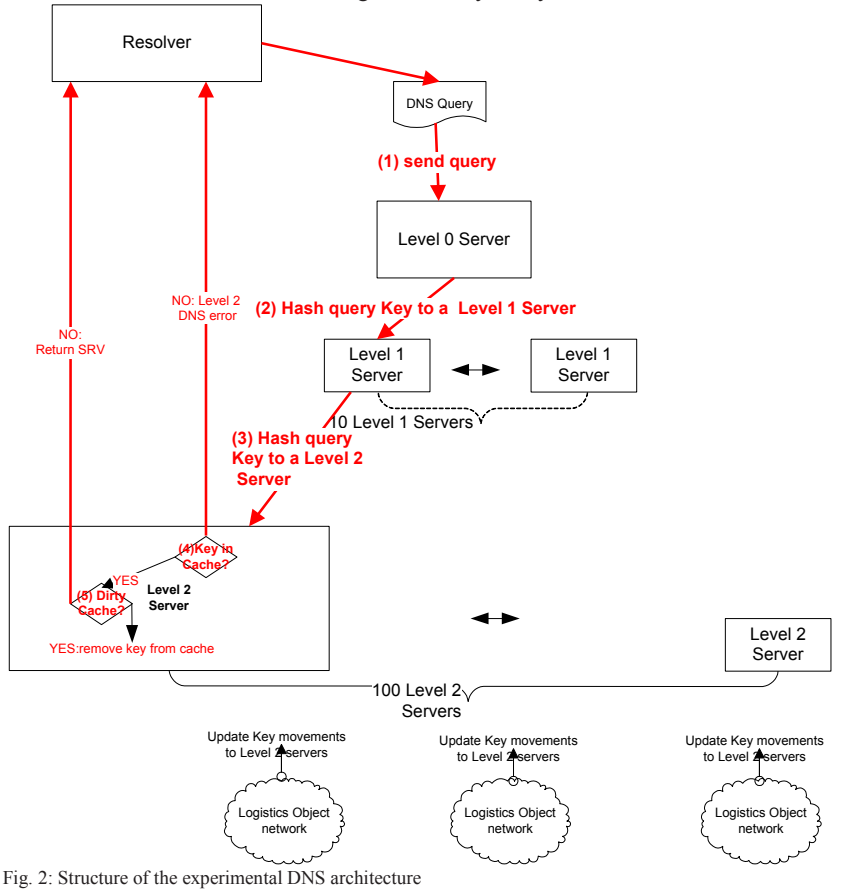

Fig. 2: Structure of the experimental DNS architecture

5 An Experimental DNS Architecture

This section describes an experimental implementation of the architecture shown in Figure 2, whose purpose had been to assess the feasibility of a tiered DNS approach for the domain of transport logistics. The aim was to simulate the behaviour of a small part of the proposed DNS (a subdomain), corresponding say to a business area of a large logistics company. Understanding the DNS behaviour of a sub-domain could then be used to extrapolate the behaviour of the whole DNS system.

For the purpose of our experimentation we adopted a three level DNS architecture, as shown in Figure 2, comprising:

- A top level server (Server Level 0), responsible for handling the DNS queries from resolvers. In real conditions this would be implemented as a cluster of (possibly geographically distributed) servers.
- Ten Level 1 authoritative servers, corresponding to major sub-subdomains within the main subdomain, i.e. to business areas/major business units within the transport logistics company that manages the top subdomain. For example, if the (hypothetical) managed Level 0 subdomain was *parcels.transport.com*, its ten (sub)subdomains could correspond to *express.parcels.transport.com*, *export.parcels.transport.com*, *domestic.parcels.transport.com* and so on.
- One hundred Level 2 authoritative servers, responsible for keeping track of the networks that host the logistics objects. Each of the Level 2 servers maintains the Service records (SRVs) of several logistics objects. An SRV record is a structure similar to that described in [11], containing information such as the TTL, hostname, port number etc., of the server the object is associated with, as well as the services provided by the object e.g. its physical location, temperature and other relevant information.

Level 2 servers cache SRV records according to their Time to live (TTL), where, as in the existing DNS, a TTL of 0 means that the record should not be cached. As a consequence of this caching policy, changes to DNS

records do not propagate throughout the network immediately, but require all caches to expire and refresh after the TTL.

TTL values of logistics objects records usually differ from those of more static resources on the Internet. A logistics object can move location (and hence address), with a frequency ranging from a few hours to a few days. A parcel distributed by a courier company in a city area network will usually spend only a few hours (or possibly less) inside the distribution centre and the delivery truck (two different networks), making its TTL value small. A deep sea container in contrast, could spend several weeks on the ship it is transported by, thus having a large TTL.

SRVs are cached by the Level 2 servers in order to avoid the latencies involved in contacting network gateways, and to improve the overall query response time. If an SRV is not in the cache of a Level 2 server (or if the cache is dirty), that server will have to wait for the gateway of the network that the logistics object has joined to push the SRV to it. Depending on the chosen policies, all Level 2 servers could share the SRVs, or alternatively each server would be responsible for managing a number of SRVs.

In our approach, Level 1 servers do not cache the unique ids of logistics objects; they cache instead the ids of Level 2 servers that they believe to have the SRV records for those objects. The reason for this decision is the need to reduce the number of DNS query redirections by using information about the domains' hierarchical nature. To reduce the number of DNS errors, search for logistics objects is directed to the servers where they are most likely to be at any time.

Hence, a Level1 server will cache the id of one or more Level 2 servers that track a logistics object. The server ids could actually be stored in the Level 1 server's SRV in a priority order. For example, a *roadtransport.transport.com* Level 1 server could store in the following order
transportequipment.road.com, warehouse.road.com , distributionhub.road.com
to specify the order in which Level 2 servers should be queried for a particular logistics object.

To simplify implementation, in this experimental setup, Level 1 servers cache only one Level 2 record id in an SRV record. That record gets updated when the Level 2 server returns a DNS error when it is queried by a Level 1 server for a particular object id.

We implemented the DNS architecture discussed above as a number of 111 concurrent processes written in the Erlang language and physically distributed over a TCP/IP LAN. We additionally implemented another process that simulates the resolvers that query the Level 0 server. We set the resolver process to send DNS queries for random object ids (keys) with a frequency of between one and fifty queries per second. We configured the resolver process so that it does not cache the query results.

We generated 1 million object ids (unique keys) and used them to create 1 million SRV records that were distributed equally (but randomly) amongst the 100 Level 2 servers. Thus, an SRV record at level 2 contains amongst other the key of the tracked object and a TTL value. We set the TTL value of each record to a random number between 0 (do not cache) and 7200 seconds (2 hours).

Next, we populated the caches of Level 1 servers with 100,000 records per server. A Level 1 DNS record is a tuple *{ObjectKey, Server2Id}* meaning that Server2id is the Level 2 server that tracks the object with Id, *ObjectKey*. Therefore, at the start of the simulation, all caches (at Levels1 and 2) contain complete and accurate information. However as logistics objects move along the transport chain, caches gradually become 'dirty' (cached information becomes inaccurate).

Five days (120 hours) of continuous simulation were employed to measure the impact of such object mobility on level 1 DNS errors as well as on cache hits and misses at Level 2.

According to the above setup, a Level 1 DNS error occurs when a Level 1 server holds an incorrect *{Key, Level2Server}* record, while a Level 2 DNS error occurs when the Level 1 Server queries a Level 2 server for a key whose TTL has expired.

Level 2 DNS errors occur because a logistics object leaves a network managed by a particular Level 2 server and moves to another network managed by another Level 2 server. In real life that could mean for example that an object is transferred to a different transport equipment (loaded on a truck) or storage location . We simulated that phenomenon by deploying another process that periodically updates randomly the level 2 SRVs by reassigning keys to different servers. When a key is moved to a new server, its TTL is reset to a new random value between 0 and 7200. We experimented with different frequencies of such key moves between 1 and 30 keys per second. We set the frequency of DNS queries to between 1 and 50 queries per second, as such range mimics the values a large logistics company would expect to receive for one of its main subdomains, under real conditions

Lastly, we implemented a hashing function that hashes keys send in a query by the resolver process to ids of Level 1 Servers. Mapping keys to Level 1 servers can in theory improve performance as an object is likely to spend most of its lifecycle within the authority of a top subdomain. For example, a logistics object of type 'parcel' is expected to spend most of its lifetime within subdomain parcel.transport.com.

We simulated the behaviour of our DNS network by configuring the SRV records with different TTL values, running the simulation and logging DNS errors, as well as cache hits and misses. The objective of this

experiment was to observe the performance of the three level caching strategy followed, for different frequencies of key migrations.

More specifically, we logged the Level 1 and level 2 DNS errors as well the number of level 2 cache hits over the total period of the simulation.

Figure 3 shows the performance of the DNS network over a period of 17,000 seconds, handling 50 DNS queries per second (50,000 queries per sampled interval) and with a key migration frequency of 30 keys per second, from a total population of 1 million keys.

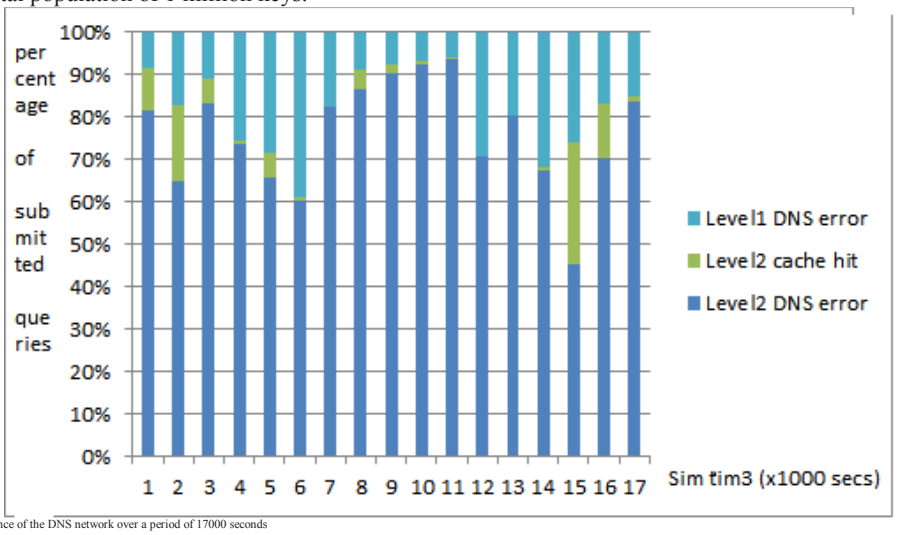

Fig. 3: Performance of the DNS network over a period of 17000 seconds

As it can be observed from Figure 3, more than 3 out of 4 of queries (76% on average) result in a Level 2 DNS error, hence in at least one query follow up. A relative lower average of 18.5% (less than 1 in 5 queries) for Level 1 DNS errors implies that the three level caching strategy is effective in reducing the burden on the top tier servers. In our approach, as in the existing DNS, updates are carried out in a push rather than a pull mode. Hence, It appears, that a tighter synchronisation between Level 2 servers and the logistics object network gateways, based on a pull mode of update, would be required to reduce the number of Level 2 DNS errors.

Effectively, this could mean that some of the network devices, protocols and functionality at the edges of the proposed DNS architecture would have to be redesigned to cope with the expected traffic.

6 Discussion and Conclusions

This paper proposed a mechanism, based on the existing DNS architecture, for scaling up the IoT capability to Internet levels. Currently, IoT is made up of a loose collection of disparate, purpose-built networks. As IoT evolves, these networks, and many others, will need to be connected [4]. As a consequence, an IoT of Internet scale will require globally unique object identifiers and mechanisms for mapping such identifiers to network addresses.

Existing object identification and naming services do not scale up, typically being limited to one standard or technology. ONS for example, only works with EPC, a unique number that is used to identify a specific item in the supply chain and that is stored on a RFID tag [3]. With a *ucode resolution server*, [14] retrieving information on objects and places is possible if their ucodes alone can be obtained even when you don't know anything (clues) about the object or place of the inquiry. However, ucode resolution requires the availability of an Information server and it is not obvious how this server network can scale up to Internet size.

Other architectures for IoT information storage and retrieval have been proposed, such as the one described in [8]. This approach is based on a hierarchical virtual storage overlay P2P network. This DHT network is designed for accurate object locating, and even allows imprecise (fuzzy) object locating. It must be noted though that this approach has not been implemented and thus its scalability potential is hard to estimate.

The concept of a DNS system for IoT has been proposed at a theoretical level [1], and prior to that was also advocated by some of the original inventors of DNS [4]. However such proposals have remained theoretical, and thus we argue that the first time the feasibility of such a DNS system has been tested in practice, is in the experimental implementation described in this paper.

More research and experimentation is required before the approach described here achieves commercial acceptance. Some of the characteristics of DNS might be inconsistent with IoT objectives. For example, a DNS

access to information is more critical than instantaneous updates or guarantees of consistency. This was reflected in the findings from our simulation. Our experiments did not measure the response time for DNS queries, as the results would be meaningless in this experimental setup. In real conditions, slower response rates essentially introduce latency which the design of resolver programs would need to take into account. Intelligent resolvers could be introduced, that establish the type of logistics object (for example, a small parcel on a domestic route, as opposed to a container on a transatlantic route), that is tracked and adapting the query strategy and/or caching method accordingly. Essentially such methods are knowledge based, and similar for example, to what is proposed in [15] where a semantic device bus and an ontology model of device services to overcome the heterogeneous device collaboration problem, are proposed.

On balance, however, results from our experiments indicate that a Domain Name Server system for the Internet of Things, can be largely based on the architecture and settings of the existing DNS. Nevertheless, due to the more volatile nature of logistics objects (compared to typical Internet resources) caching strategies would need to be modified and be multi-type.

In conclusion, a DNS implementation for the Internet of things like the one discussed here would benefit the logistics industry and the economy as a whole by making easier the integration of information across whole supply chains. While standardisation, legal and commercial problems would need to be overcome before it became a reality, as this paper has demonstrated, the idea of DNS for physical objects is, at least in principle, a feasible one, and can influence ongoing initiatives and reference architectures for IoT [6].

References

[1] Afilias (2008) *Finding your Way in the Internet of Things. An Afilias Whitepaper*. September 2008. Available from www.afilias.info

[2] Atzori, L., Iera, A, Morabito, G. *The* Internet of Things: A survey. *Computer Networks: The International Journal of Computer and Telecommunications Networking archive* Volume 54 Issue 15, October, 2010.

[3] EPCGlobal (2007) *Electronic Product Code (EPC): An Overview*. Available from http://www.gs1.org/docs/epcglobal/an_overview_of_EPC.pdf

[4] Evans D. *The Internet of Things. How the Next Evolution of the Internet Is Changing Everything*. April 2011 Cisco Internet Business Solutions Group (IBSG).

[5] GS1. *EPCglobal Object Name Service (ONS) 1.0.1* May 29, 2008 Version: 1.0.1

[6] Haller, S (ed). *IOT – i The Internet of Things Initiative D1.2 First Reference Model White Paper*, 2011. Available from http://www.iot-i.eu/public/public-deliverables/

[7] Liu, Y and Zhou G. Key Technologies and Applications of Internet of Things. *Fifth International Conference on Intelligent Computation Technology and Automation, Zhangjiajie, China, 2012.*

[8] Liu, W, Lihua Yin Weizhe Zhang_*and* Hongli Zhang. A general distributed object locating architecture in the Internet of Things. *16th International Conference on Parallel and Distributed Systems*, *Shanghai, China, 2010.*

[9] Mockapetris, P. *Letting DNS Loose*. Jan 02, 2004, available from http://www.circleid.com/posts/letting_dns_loose/

[10] Permala , A, Karri Rantasila and Eetu Pilli-Sihvola . RFID: From Closed Systems to Improving Visibility in the Manufacturing Supply Chain *IJAL*, 3:2, 2012, p. 14-24.

[11] RFC 781, *Internet Protocol - DARPA Internet Program Protocol Specification*, Information Sciences Institute, J. Postel (Ed.), The Internet Society (September 1981)

[12] RFC 1034, *Domain Names - Concepts and Facilities*, P. Mockapetris, The Internet Society (November 1987)

[13] RFC 1035, *Domain Names - Implementation and Specification*, P. Mockapetris, The Internet Society (November 1987).

[14] Sakamura, K. *Ubiquitous ID Technologies*. uID Center, 2011. Available from www.uidcenter.org/

[15] Yang, K, Shijian Li, Li Zhang, Gang Pan. Semantic Device Bus for Internet of Things. *EEE/IFIP International Conference on Embedded and Ubiquitous Computing, Hong King, China, 2010.*

[16] Yuqiang, C , Jianlan G and Xuanzi H. The research of Internet of things' supporting technologies which face the logistics industry, *2010 International Conference on Computational Intelligence and Security, Nanning, China, 2010.*