USING DNS FOR GLOBAL DISCOVERY OF ENVIRONMENTAL SERVICES

Andreas Kamilaris and Andreas Pitsillides
Networks Research Laboratory
Department of Computer Science, University of Cyprus, Nicosia, Cyprus

Keywords: Domain Name System, Service Discovery, Web of Things, DNS-SD, REST, Environmental Services.

Abstract: Embedded sensor devices are getting pervasive in our everyday lives. Due to the penetration of the Internet in the real world, sensor devices become globally addressable, while the Web of Things envisions Web presence to these devices and their services. Future abundance of Web-enabled sensors would imply the need for efficient techniques for globally discovering them. To solve this problem using the existing Internet infrastructure, we explore the exploitation of the Domain Name System (DNS) as a scalable, ubiquitous, real-time directory mechanism for embedded devices. We investigate some practical issues that could arise from such an attempt and we discuss the potential of this approach.

1 INTRODUCTION

Embedded computing is becoming an integral part of our lives. It is expected that millions of tiny sensors are measuring with high precision environmental conditions such as wind and temperature or physical events such as pressure and motion.

Lately, the Internet is penetrating slowly-slowly into the real world of physical objects. The introduction of IPv6 and the efforts for porting the IP stack on embedded devices (Dunkels et al., 2004), enable the vision of a global Internet of Things (IoT) (Gershenfeld et al., 2004).

While the IoT brings interoperability between heterogeneous devices at the network layer, there is still a need for consensus at the application layer. To address this need, the Web of Things (WoT) (Wilde, 2007), (Guinard et al., 2010) reuses well-accepted and understood Web standards to interconnect IP-enabled resource-constrained devices. Thus, embedded sensors can become fully integrated to the Web, directly by embedding Web servers on them (Yazar and Dunkels, 2009) or indirectly by means of gateways (Kamilaris et al., 2011).

In the future, Web-enabled sensor devices will constitute the large majority of the Web population. These devices will be deployed around the world to measure with high accuracy the physical environment, exposing their functionalities as open Web APIs. These Web APIs would probably conform to the REST architectural style (Fielding, 2000), (Richardson and Ruby, 2007), which is a core element of the WoT.

indent In general, there do not exist yet standardized, scalable and flexible ways to globally discover these embedded devices, based on their characteristics and capabilities. Dyser (Ostermaier et al., 2010) and Snoogle (Wang et al., 2008) are early efforts towards real-time discovery of physical entities, however, they either require additional Web infrastructure or they do not scale for the World Wide Web. On the other hand, microformats[1] suggest ways for making HTTP data available for indexing and searching, but their utilization for sensor devices would augment our dependency to commercial search engines.

We believe that service discovery of embedded sensor devices needs to be ubiquitous to the users of the Web. The proposed solution must comply with existing Internet standards and should not require major changes to the existing technical equipment and protocols. Users should be able to discover environmental services simply by typing related keywords in their favorite Web browser. In this way, discovery of physical devices may be similar to the way we discover Web sites through search engines.

In this paper, we propose to exploit the existing Internet infrastructure to achieve real-time discovery of embedded devices and environmental services. We investigate the utilization of the Domain Name System (DNS) as a scalable, pervasive, global meta-data

[1] http://microformats.org/

Kamilaris A. and Pitsillides A..
USING DNS FOR GLOBAL DISCOVERY OF ENVIRONMENTAL SERVICES.
DOI: 10.5220/0003953302800284
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 280-284
ISBN: 978-989-8565-08-2

repository for embedded devices, and its extension for supporting location-based discovery of Web-enabled physical entities.

Applying an existing technology for discovery of environmental services, especially one that has been ubiquitous for decades, offers many advantages, including well-defined support, easy configuration, experienced developers and users, as well as availability of open-source implementations.

The general idea of exploiting DNS for service discovery is not new. DNS-based Service Discovery (DNS-SD) (Cheshire and Krochmal, 2011a) proposes using standard DNS programming interfaces, servers and packet formats to browse a network for services. Similarly, Multicast DNS (mDNS) (Cheshire and Krochmal, 2011b) provides the ability to perform DNS-like operations on a local network in the absence of any conventional unicast DNS server.

Even though both protocols have been originally designed for device/service discovery in local networks, DNS-SD has been extended to provide wide-area service discovery. However, this functionality for service advertising is domain-centric, meaning that users are only able to be informed about services offered in some particular domain. In contrast, our proposal is service-centric, envisioning to enable global, real-time, location-based discovery of pervasive services, offered by Web-enabled sensor devices.

In the rest of the paper we examine the feasibility of this idea, addressing a number of practical issues. Our approach conforms to the DNS-SD protocol, where possible.

2 ENVIRONMENTAL SERVICE DISCOVERY THROUGH DNS

DNS is a hierarchical, distributed naming system for computers. Its main functionality is the translation of domain names meaningful to humans into IP addresses meaningful to machines. DNS distributes the responsibility of assigning domain names and mapping those names to IP addresses by specifying *authoritative name servers* for each domain. Authoritative name servers are responsible for their particular domains, and in turn can assign other authoritative name servers for their sub-domains. This mechanism allowed the DNS to be distributed and fault-tolerant.

Domain names consist of *labels* that are concatenated and delimited by dots, such as *www.webist.org*. The right-most label indicates the top-level domain, which is *org* in the previous example.

In the following subsections, we examine how to change the organization of DNS to support location-based, real-time discovery of environmental services.

2.1 Device Registration

We propose the inclusion of a new top-level domain at the DNS, which is the *env* domain. This domain is intended to support all embedded devices and environmental services which are enabled to the Web and registered to the DNS.

Specialized authoritative name servers may be responsible for this domain. These .env domain name servers shall allow real-time registration of physical devices and their (RESTful) Web services through Dynamic DNS (DDNS) (Rekhter et al., 1997).

In general, a particular service[2] instance can be described in DNS using SRV (Gulbrandsen et al., 2000) and TXT (Mockapetris, 1987) records. A SRV record has a name of the form:

$$< Instance > . < Service > . < Domain > \quad (1)$$

and gives the target host and port where the service instance can be reached. The DNS TXT record of the same name gives additional information about this instance, in a structured form using key/value pairs

Whenever a Web-enabled sensor device becomes available on the Web, it would create a request to the .env DNS server, asking for registration. In this request, the device must specify its name, location and the services it offers. The .env server could offer a Web API, allowing sensor devices to POST their discovery details in HTTP requests. In case all information is provided, the .env DNS server would acknowledge the request, assigning a fully qualified domain name to the device, registering it in a SRV record.

Hence, each sensor device will be assigned a unique hostname that has the following format:

$$devicename.\_http.\_tcp.service.location.env \quad (2)$$

where *devicename* is the user-friendly name of the sensor device (<Instance> portion of the SRV record), *service* is the actual Web service offered by it and *location* is the absolute location of the device. Service and location define the <Domain> portion, which becomes *service.location.env*. The <Service> section is either *_http._tcp* or *_http._udp*.

A sensor device may offer various environmental services. Thus, multiple hostnames will be created for this device, each for a different offered service. Since the WoT proposes a resource-oriented architecture (Richardson and Ruby, 2007), services must

[2] To avoid confusion, we note that by the notion of a "service", the DNS refers to the Internet protocols used to interact with some domain. In the WoT, services are meant as the Web-based functionalities offered by Web-enabled sensor devices.

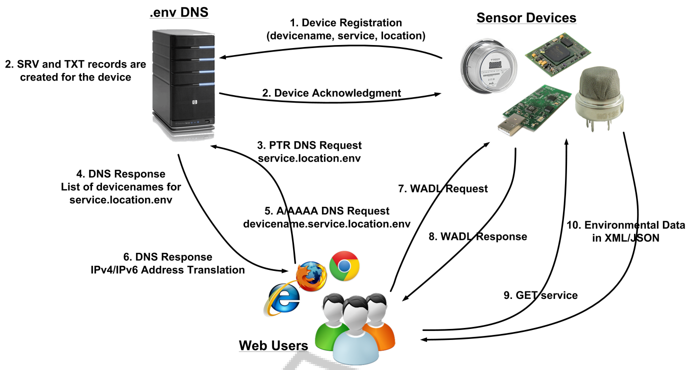

Figure 1: Environmental Service Discovery Procedure using DNS.

be described with uniform resource identifiers (URIs) (see Section 3). This avoids ambiguities when users construct environmental URLs in their Web browsers.

2.2 Service Discovery

Service discovery begins when a user types in his Web browser a URL ending with the .env label. Since the user may not be able to construct the equation shown in (2), he could ask about all sensor devices offering *service* and deployed in *location*. This would be achieved by using the PTR DNS record type (Mockapetris, 1987), as shown in Table 1, by specifying a query of the form <Service>.<Domain>. In this case, DNS would respond to the PTR lookup with a list of relevant <Instance> records.

In case there exist numerous relevant records, a technique such as round-robin DNS could be employed. Round-robin DNS is a technique for achieving load distribution and load balancing by responding to requests from various client computers according to an appropriate statistical model.

After the user receives the list of relevant instances, he can use any of them to construct a query similar to the equation shown in (2). By typing this URL in his Web browser, the DNS will translate it to the actual IPv4/IPv6 address of the corresponding sensor device. This translation is realized by using the A and AAAA DNS record types, as shown in Table 1. The user can also use the TXT DNS record type to receive general characteristics/capabilities of some sensor device, in case they are available.

When the IP address is resolved and the request is forwarded to the appropriate sensor device, the device needs to respond with a description of its functionality. This is necessary in order to understand the semantics of the device (e.g. indoor/outdoor, degree of accuracy, measurement unit) and how to interact with it to get informed about the environmental conditions. Thus, a description language must be defined, which declares the device semantics, but also explains the interaction possibilities with it.

Example description languages that could be adopted are Extended Environments Markup Language (EEML)[3] and SensorML[4]. These are languages that describe the capabilities of sensor devices, but do not describe effectively the interaction with them.

Therefore, we propose to employ Web Application Description Language (WADL)[5], which is an XML-based language that provides a machine-readable description of HTTP-based applications. It can be considered as the RESTful equivalent of Web Services Description Language (WSDL)[6], which is a standard for describing SOAP-based Web services. WADL is intended for applications based on the Web architecture, and it is meant as a platform- and language-independent way of describing services.

After the user receives the WADL description, he can construct the appropriate request, query the sensor device and get informed about the environmental conditions in the selected location, in a well-known format such as XML or JSON. The whole discovery procedure can be observed in Figure 1.

Environmental service discovery may involve not only humans but also machines, for automated M2M communication. In this case, machines could discover useful services on demand, and take decisions automatically based on current environmental conditions.

2.3 Freshness of Information

The idea of online directories for Web services (e.g. UDDI[7] for WS-*) never worked, mainly because of information inconsistency and unavailability. Through DNS, these inconsistencies can be avoided.

[3]http://www.eeml.org/
[4]http://www.opengeospatial.org/standards/sensorml
[5]https://wadl.dev.java.net/
[6]http://www.w3.org/TR/wsdl
[7]http://uddi.org/pubs/uddi_v3.htm

Table 1: DNS Record Translation.

No	DNS Record	Explanation	Example Record/URL
1	SRV	Records the service instance of a device offering *service* at *location*	mydevice._http._tcp.pollution.porto.env
2	PTR	Lists which devices support *service* at *location*	_http._tcp.pollution.porto.env
3	A	Translates an environmental URL to a 32-bit IPv4 address	mydevice.pollution.porto.env
4	AAAA	Translates an environmental URL to a 128-bit IPv6 address	mydevice.pollution.porto.env
5	TXT	Meta-data about the selected sensor device	mydevice._http._tcp.pollution.porto.env

In general, the DDNS service allows the DDNS server to allocate a static hostname to a physical device, and whenever the device is allocated a new IP address, this is communicated to the DDNS provider by software running on the device.

Furthermore, Dynamic DNS Update Leases (Cheshire et al., 2006) constitutes a method of extending DDNS to contain an update lease life, allowing a DNS server to perform DNS dynamic updates with an attached lease time, that are automatically deleted unless renewed before the lease expires.

Hence, the name server ”forgets” after some time interval the registration of the sensor devices. Devices may be forced to state frequently their operability to the DNS server, by re-registering, e.g. every some hours. In this way, dynamic IP address assignment to devices could be supported and also device unavailability could be identified in a relatively small delay.

3 PRACTICAL ISSUES

Our approach creates numerous practical issues that must be effectively addressed. The general operation of the DNS is expected to be affected, mainly in terms of management, security and increased traffic.

Concerning management issues, since the DNS follows a hierarchical structure, the addition of the .env top-level domain is not expected to be a complicated task. Availability of devices and services may be assured by DDNS update leases.

Partial reliability could be ensured by checking the IP addresses of the sensor devices, if they fall in the locations claimed by them during the registration process. More reliability and trust would be achieved if some user-based feedback system was applied for sensor devices and their environmental services, similar to the way eBay works for rating its users.

User-based rating could be realized if sensor devices had a social presence on the Web. Towards this direction, Evrythng[8] company aims to create Web-based social profiles for physical devices. It would then be easy for users to rate them, according to their quality of service. Such an initiative would also provide more advanced privacy, allowing the owners of the devices to share them only with their family, friends or everyone. In general, the social Web could be harnessed for authenticating and authorizing users to interact with environmental services.

Our approach requires unique devicename assignment to sensor devices that have same services and location. Upon a conflict, the DNS server should automatically select a new name for the device, typically by appending a digit at the end of its name. Since the .env DNS server would constitute a distributed repository, the devicename assignment should be visible to all .env name servers. Due to the massive amount of Web-enabled devices in the near future, this could be the cause of high load to DNS. However, this issue may be mitigated by area- or location-based assignment of devices to the .env DNS servers.

Another important issue concerns naming of services and locations. For example, locations are named differently in each language (e.g. ”Lisbon” in English vs ”Lisboa” in Portuguese). To avoid these ambiguities and guarantee uniqueness, electronic directory services such as X.500[9] could be used, where a distributed database would contain unique translations for services and locations.

Since the DNS system supports a built-in caching mechanism, this mechanism could be exploited to support caching of sensory data. During their registration, sensor devices would include their latest measurements, stored in TXT resource records. These measurements could then be forwarded to users who query the .env DNS server for a relevant service, while they are still fresh[10]. Nonetheless, current DNS infrastructure can not tolerate such high loads.

Finally, the whole system would be optimized by forcing the .env DNS server to return directly the IP addresses of relevant sensor devices and not their hostnames. However, this may not be practical since the IP addresses of the devices may change frequently while their hostnames remain the same. Furthermore, some users might prefer to use some particular sensor devices they found more reliable than others. By exchanging only IP addresses, this connection between Web clients and sensor devices can not be maintained.

[8] http://evrythng.net/

[9] http://www.x500standard.com/

[10] Defining the freshness of measurements varies between devices and services.

4 DISCUSSION

The procedure of discovering environmental services can be automated by selecting some sensor device from the list returned by the .env DNS server, parsing its WADL file and constructing HTTP requests.

This process can even be personalized, by selecting only devices that meet particular user preferences. For example, some users may wish to interact solely with devices having positive online feedback or those belonging to well-known authorities such as governmental organizations. User characteristics could be extracted from their online social networking status.

Since the whole idea is participatory-based, only people who are willing to share their sensor devices with the online community would do so. We believe that in the future a culture could be created around the concept of sharing environmental services with the rest of the world.

Even though our proposal targets environmental services, it could be well generalized to support any kind of physical devices and/or pervasive services. To achieve this, standardized "domain vocabularies" need to be created, for facilitating the construction of queries by end users. For example, a user that wishes to park his car in Barcelona could just need to type in his Web browser *parking.cars.Barcelona*.

Finally, defining extended environmental ontologies would encourage automatic information retrieval, generalized inferences and advanced Web mashup development very easily. For example, when the temperature in Porto is obtained, then the general temperature of Portugal can be automatically inferred.

5 CONCLUSIONS

Embedded computing is becoming ubiquitous in our everyday lives, being slowly-slowly blended with the Web. Discovering in real-time miniaturized Web-enabled sensor devices, deployed around the world, is a problem that needs to be effectively solved.

In this paper, we investigated theoretically how the DNS could be extended to support global discovery of environmental services. Our small research indicates that it may be feasible to achieve automatic, real-time discovery of sensor devices, by means of specialized domain name servers. Definitely, our proposal constitutes only a novel idea and it would require a great willingness from Web engineers to be realized.

It would be interesting to study ways for extending DNS-based discovery of any Web resource offering a RESTful Web API. Security is another crucial factor not discussed that requires extensive research.

For future work, we plan to experiment with BIND[11], which is an open-source DNS software for Linux, implementing a .env DNS server in a local environment, demonstrating the potential of this approach. In parallel, we will develop a Mozilla Firefox extension that automatically discovers sensor devices just by typing relevant URLs, constructs their hostnames and interacts with them.

REFERENCES

Cheshire, S. and Krochmal, M. (2011a). DNS-Based Service Discovery. IETF Draft, draft-cheshire-dnsext-dns-sd.txt.

Cheshire, S. and Krochmal, M. (2011b). Multicast DNS. IETF Draft, draft-cheshire-dnsext-multicastdns.txt.

Cheshire, S., Krochmal, M., and Sekar, K. (2006). Dynamic DNS Update Leases. IETF Draft, draft-sekar-dns-ul-01.txt.

Dunkels, A., Voigt, T., and Alonso, J. (2004). Making TCP/IP Viable for Wireless Sensor Networks. In *Proc. EWSN 2004*, Berlin, Germany.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, California.

Gershenfeld, N., Krikorian, R., and Cohen, D. (2004). The Internet of Things. *Scientific American*, 291(4):76–81.

Guinard, D., Trifa, V., and Wilde, E. (2010). Architecting a Mashable Open World Wide Web of Things. Technical Report No. 663, ETH Zurich.

Gulbrandsen, A., Vixie, P., and Esibov, L. (2000). A DNS RR for specifying the location of services. RFC 2782.

Kamilaris, A., Trifa, V., and Pitsillides, A. (2011). The Smart Home meets the Web of Things. *IJAHUC Journal*, 7(3):145–154.

Mockapetris, P. (1987). Domain Names - Implementation and Specifications. STD 13, RFC 1035.

Ostermaier, B., Roemer, K., Mattern, F., Fahrmair, M., and Kellerer, W. (2010). A Real-Time Search Engine for the Web of Things. In *Proc. of the Internet of Things 2010 Conference*, Tokyo, Japan.

Rekhter, Y., Thomson, S., Bound, J., and Vixie, P. (1997). Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136.

Richardson, L. and Ruby, S. (2007). *RESTful Web Services*. O'Reilly.

Wang, H., Tan, C., and Li, Q. (2008). Snoogle: A Search Engine for the Physical World. In *IEEE INFOCOM 2008*, pages 1382 –1390, Phoenix, AZ, USA.

Wilde, E. (2007). Putting things to REST. Technical Report UCB iSchool Report 2007-015, UC Berkeley.

Yazar, D. and Dunkels, A. (2009). Efficient Application Integration in IP-based Sensor Networks. In *First ACM Workshop On Embedded Sensing Systems For Energy-Efficiency In Buildings (BuildSys)*, California, USA.

[11] http://www.isc.org/software/bind