Interoperable Services on Constrained Devices in the Internet of Things

Hauke Petersen, Emmanuel Baccelli, Matthias Wählisch

Interoperable Services on Constrained Devices in the Internet of Things

Hauke Petersen[1], Emmanuel Baccelli[2], and Matthias Wählisch[1]

[1]Freie Universität Berlin, Germany
[2]INRIA, France

***Abstract*—The Internet of Things (IoT) promises billions of constrained devices connected to the Internet in the near-future. The efficient integration of these massively deployments into service-oriented architectures requires light-weight and highly automated mechanisms for device configuration and setup on all layers of the network stack. In this paper we highlight how the set of existing solutions for configuration and service management (i.e.: discovery, description, invocation) are insufficiently analyzed in the context of the IoT on one hand, and on the other hand do not fit all the specific requirements and attributes of constrained devices, in terms of memory usage and power consumption in particular.**

I. Introduction

THE Internet of Things (IoT) is currently emerging: the next billions of machines [1] connected to the global IP network are expected to consist in a variety of heterogeneous devices, ranging from wireless sensors to smart home appliances and many other types of machines that were typically not connected so far. These developments are expected to profoundly transform our environment, which will be heavily influenced by the new cyber-physical reality that is going result from automated interaction between these devices – with or without humans in the loop. In effect, our interface to the Internet may soon no longer predominantly be our traditional keyboards, mouse and/or screens, but rather a multitude of heterogeneous, interconnected smart objects.

II. Constrained Devices

The vast majority of devices that will constitue the IoT are expected to be severely constrained in terms of memory, CPU as well as power capacities (e.g., running on tiny batteries). To that effect, the term *constrained devices* [2] was recently introduced to define a category of connected devices with stringent resource restrictions compared to common desktop computers, such as (i) significantly reduced power consumption (mWatt vs. Watt), (ii) much less computation power (MegaFLOPS vs. TeraFLOPS), or (iii) orders of magnitude less memory (KiloBytes vs. GigaBytes). Furthermore, constrained devices are typically based on micro-controllers that provide only a limited set of functionalities e.g., they are typically not equipped with memory management units (MMU), which de facto rules out using operating systems such as Linux on such devices.

Remark 1 — About software running on constrained devices. The IoT is is currently held back by fragmentation due to a plethora of too rudimentary, and too hardware-specific software platforms, employed on constrained devices to accommodate network stacks and applications. Only recently is progress being made in this domain with the emergence of new operating systems (for instance RIOT [3], or Contiki [4]) which aims to provide an open source, modern, generic software platform upon which one can conveniently build IoT application software. It is therefore likely that a couple of OS will become dominant and thus defragment the IoT in the near future.

Remark 2 — About the future of aforementioned constraints. One might initially hope that, due to Moore's Law (or something similar), these constraints will soon disappear and are thus mostly irrelevant. However, experience over the last decade shows that such an evolution is in fact not taking place. For example, the memory limitations of recent tiny IoT devices [5] are roughly the same as the limitations of wireless sensors introduced 10-15 years ago [6]. This may be explained by the fact that advances in semiconductor technology enable not only more powerful devices, but also lead to significant cost reductions for constrained devices. As typical IoT deployments will consist in large number of devices (hundreds to millions), extremely low costs will play a key role and will thus likely lead to a future where the aforementioned constraints remain relevant.

III. Wireless Hyper-connectivity

When talking about the IoT, we are of course also talking about physical connectivity between smart objects and network protocols running on these devices. Ten to fifteen years ago, the Internet consisted mainly of servers, work-stations and personal computers that were connected predominantly through a fixed, wired network infrastructure. In the early 2000s the Internet started to evolve towards relying more and more on wireless communications to access the wired network infrastructure, via link-layer technologies such as IEEE 802.11 or cellular technologies such as UMTS and more recently LTE. Now, the Internet edge is pushing towards *wireless hyper-connectivity*, whereby the density of wirelessly interconnected devices increases dramatically and most devices have multiple wireless interfaces.

Remark 3 — About IoT communication technologies. One can expect that multiple link layer wireless technologies are here to stay. However, approaches are emerging at the

network layer, able to simultaneously harness and interconnect heterogeneous (existing and future) link layer technologies that are relevant for the IoT: a good example of such an approach, based on open standards, is IPv6 combined with optimizations at various layers e.g., UDP/RPL/IPv6/6LoWPAN. We expect that such an inclusive, layered approach will defragment the IoT in the near future.

Remark 4 — About IoT network architecture.
The availability of densely deployed, multi-radio interface devices and goals such as local interaction between devices to enable resilient, ubiquitous environment automation will likely lead to a network architecture that will both leverage not only traditional, infrastructure-based network paradigms, but also spontaneous wireless network paradigms [7]. Spontaneous wireless networking allows devices to dynamically self-organize the relaying of data towards destination – even without the help of infrastructure and pre-provisioned access points. This enriched network architecture will fuel a new world of distributed IoT processes and applications, that can seamlessly interconnect with one another and/or with the cloud.

IV. Open Standards: Building Blocks for Applications and Services in the IoT

The Internet of Things introduces two challenges. First, a significant of number (billions) of additional devices will be connected to the Internet, most deployments consisting in large batches of devices. Second, the majority of these devices will be using predominantly machine-to-machine (M2M) communication, i.e., such devices will only present external interfaces that are not primarily designed for human interaction. Thus, for both scalability reasons and interface design reasons, approaches which reduce management and configuration tasks to a minimum are necessary. Hence, autoconfiguration mechanisms are needed on *all layers of the network stack*.

While autoconfiguration mechanisms already exist for other types of machines connected to the Internet, there are however several issues to be addressed for IoT devices autoconfiguration. On one hand, many standard autoconfiguration solutions still require manual interaction to some extent. On the other hand, standard autoconfiguration mechanisms where not designed to fit the extreme memory and energy constraints that software running on IoT devices must comply with. Furthermore, existing autoconfiguration solutions focus on specific services.

In the following, for reasons including scalability and convenience, we thus argue not only for completely automated configuration solutions for IoT devices, but also for additional mechanisms enabling automated generic service deployment in this context.

A. Beyond Autoconfiguration: Service-driven Architecture

To cope with the increasing complexity of distributed systems the concept of service-oriented design has emerged [8], [9]. The key idea behind this is a system abstraction that enables functional system design by focusing on the actual functionality and end-user benefit rather than on underlying technology. The building blocks of such systems are considered as services, such as data providers (e.g., sensors).

At the service level, neither the actual physical characteristics of the device providing a service, nor the nature of the underlying network need to be dealt with. This approach thus enables the development of services without prior knowledge of device-specific implementation details, as such development can focus on the service(s) the device will offer. The concept of service-driven architecture suits well with the requirements of IoT deployment, by drastically reducing the complexity of dealing with large numbers of heterogeneous devices. However, to this day, building blocks and tools to implement this concept for IoT and constrained devices are not available .

The deployment and maintenance of IoT devices consist in mainly three different aspects, which are (i) configuration and management of the network layer, (ii) configuration and management of the security mechanisms, and (iii) configuration and management of the services of a device. For the first two aspects there are already solutions available that at least partly provide means for automated or effort-less configuration. For example, IPv6 stateless autoconfiguration [10] and 6LoWPAN neighbor discovery enhancements [11] enable network configuration mostly without manual interaction. DTLS [12] enables security mechanisms autoconfiguration configuration In the following, we will not discuss the completeness of these solutions or their adequacy with IoT scenarios and constrained devices, but rather focus on the third aspect mentioned above: configuration and management of the services of an IoT device.

To the best of our knowledge, autoconfiguration, discovery and usage of high-level services is to date only partly solved. Most of the protocols employed in this domain (such as DNS-SD [13] and SSDP [14]) aim only for service advertisement and discovery. Using the services then still require the implementation of the actual service protocols. However, while this approach may have worked so far in the Internet, IoT devices would greatly benefit from a coherent and standardized service implementation model, which could both (i) decrease the requirements with respect to local resources e.g., memory and (ii) better fit the M2M paradigm.

B. Towards Service-based Design for IoT Devices

The lack of a coherent set of open building blocks inhibits application development based on a service-based design approach. We argue that this gap should be filled with solutions that enable applications to use automatically and coherently services offered by interconnected service providers (some of which being IoT devices). In the following, we briefly discuss first steps towards this direction.

1) Virtual Machine & Interpreted Language Aspects: A possible approach to close this gap is to tie concepts of VM-based or interpreted programming languages to the concepts of SOAP or remote procedure calls (RPC). In conjunction with automated code generation based on service description languages, such an approach could enable easy adoption of services. While such concepts work well in specific scenarios,

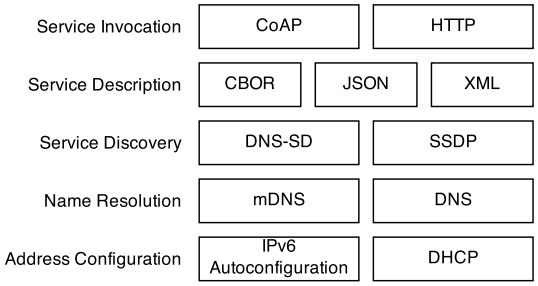

Fig. 1. Available protocols for different layers of service based applications.

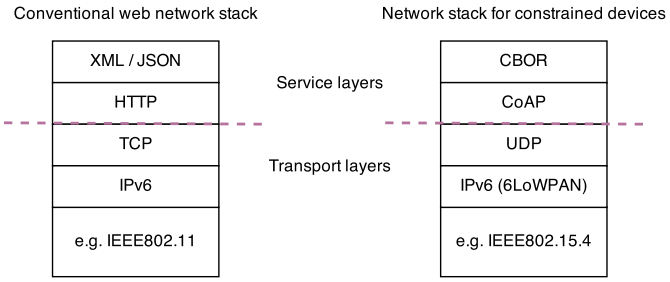

Fig. 2. Comparison of protocols in the conventional web network stack and the envisioned network stack for constrained devices.

there are drawbacks, especially when applied on constrained devices. For machines with very limited amount of memory it is not feasible to introduce additional software layers (e.g., virtual machines or heavy-weight middlewares) on top of the operating system (which must itself be as lightweight as possible). Such approaches thus cannot be deployed on very constrained devices, which is a significant issue since seamless interoperability requires the widest possible applicability of the solution, across all IoT devices. In the near-future, a thorough assessment is thus needed concerning how efficient such approaches can be in practice with respect to minimum memory footprint.

2) Application Layer Communication Protocols Aspects: Figure 1 shows a set of standard application layer protocols that are frequently-used to implement services. However, when it comes to constrained devices, the applicability of these protocols is limited. For instance, data encoding schemes and transport protocols that are based on human readable encoded data are verbose and thus not efficient enough when memory and energy resources are limited. Recently, binary-based protocols such as CoAP and CBOR have been introduced to better fit M2M and IoT requirements on constrained devices.

Moreover, the chattiness of these protocols is yet to be tested in the context of the required energy efficiency imposed by IoT constraints. In the near-future, a more comprehensive study of these protocols working together as a protocol stack will have to be conducted. Performance evaluation is required with respect to memory consumption, and more importantly, with respect to energy consumption. In particular, energy consumption of these solutions has only been studied in isolated and simplified setups, while low power networks have precise requirements in terms of reduced network activity, maximized sleep times, and low memory footprints.

V. Perpectives

The Internet of Things is already here from the hardware perspective. Massive deployment of constrained devices are already taking place or are planned in the near-future. The sheer number of new devices expected to be connected to the Internet calls for (i) efficient and automated configuration and management capabilities, on all layers of the network stack, and (ii) new application development paradigms such as service-base design that can reduce the complexity of gluing high-level applications to the actual devices that are connected to physical world. However, constrained devices differ from other networking devices in terms of processing, memory and power capacities. Moreover, constrained devices will be used prevalently in M2M scenarios. Existing protocols and mechanisms for device configuration are mainly developed for conventional Internet devices which do not have the aforementioned, stringent constraints, and generally do not possess the agility to adapt for this new heterogeneity. Future work thus has to be undertaken to (i) analyze existing configuration protocols with focus on power consumption and memory requirements and (ii) develop a unified software stack that enables the coherent integration of constrained devices into service-driven environments.

References

[1] Ericsson, "More than 50 billion devices," Ericsson White Paper, Tech. Rep., 2011.
[2] C. Bormann, M. Ersue, and A. Keranen, "Terminology for Constrained-Node Networks," IETF, RFC 7228, May 2014.
[3] "RIOT - The friendly OS for the IoT." [Online]. Available: https://www.riot-os.org
[4] "Contiki." [Online]. Available: http://www.contiki-os.org
[5] "ARM Cortex-M0." [Online]. Available: http://www.arm.com/products/processors/cortex-m/cortex-m0.php
[6] "Texas Instruments MSP430." [Online]. Available: http://www.ti.com/lsds/ti/microcontroller/16-bit_msp430/overview.page
[7] J. Cordero, J. Yi, T. Clausen, and E. Baccelli, "Enabling Multihop Communication in Spontaneous Wireless Networks," in *ACM SIGCOMM eBook on "Recent Advances in Networking", Volume 1, Chapter 9, pp. 413-457*. ACM, 2013.
[8] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services," *Services Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 223–235, 2010.
[9] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *Internet of Things (IOT), 2010*. IEEE, 2010, pp. 1–8.
[10] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," IETF, RFC 4862, September 2007.
[11] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," IETF, RFC 6775, November 2012.
[12] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2," IETF, RFC 6347, January 2012.
[13] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," IETF, RFC 6763, February 2013.
[14] Y. Goland et al., "Simple Service Discovery Protocol/1.0 Operating without an Arbiter," https://tools.ietf.org/html/draft-cai-ssdp-v1, IETF, Internet Draft, 1999.