Position paper submitted to Workshop
“Interconnecting Smart Objects with the Internet”

Authors:
Daniel Barisic (BOSCH Security Systems, daniel.barisic@bosch.com)
Anton Pfefferseder (BOSCH Security Systems, anton.pfefferseder@bosch.com)

Unified Device Networking Protocols for Smart Objects

Web of things is the future

In the coming years, the use of smart objects will drastically increase in different application areas like Smart grid, Home automation, Building automation and Health care. Driver for these activities will be the business value coming from cost reduction due to reduced development, installation and maintenance effort in comparison to the existing, proprietary solutions or niche standards of today. Furthermore, new added value solutions will be enabled by the convergence of sensors/actuators with IT systems as well as by sharing sensor information over multiple systems. The systems that are nowadays build as vertical solutions will morph to open, standardized, IP based solutions in the future.

Key enabler for this is the integration of IP capable transceivers directly into the smart objects. To this end, diverse transceivers are needed that address the specific requirements with regards to cost, performance and energy consumption of the different application areas. Some of the well known physical layers at the moment are 802.15.4, Wifi, PowerLine and G.Hn. In the future, this portfolio will continuously be augmented with other wireless and wired solutions to finally enable the integration of IP into all application areas.

Without a doubt, IP will take over the role as it did with today’s Internet, which is to act as a convergence layer that unifies the different communication solutions and therefore provides a homogenous infrastructure by which the future smart objects are accessible. This will make the use of smart objects even more attractive, as communication with them is greatly simplified and does not require detailed knowledge about the objects HW platform, SW stack or physical network connection. In addition, there will be no need for dedicated protocol translation gateways anymore, which introduce significant complexity and cost into the current smart object deployments.

Device Networking beyond IP

While IP will be a crucial component to enable the web of things, it is not the only necessary ingredient. When we look at existing systems that deal with smart objects today (e.g. BACnet, LON), we can identify a certain set of features that is supported by all these systems. These features include discovery mechanisms that allow identifying when a new device joins the system as well as mechanisms to search for connected devices with particular characteristics (like e.g. location, type of sensor, etc.). In addition, solutions are in place to dynamically inspect the capabilities of a device and the data it is able to provide (description). The notification of events or alarms is as well a typical feature that can be found in any existing system. Also, newer standards like the upcoming, IP based Zigbee Smart Energy Profile 2.0 include support for the mentioned features.

The reason for this is that all these features are motivated by the basic characteristics of smart object networks, which are the inherent flexibility in comparison to classical IT systems and the interconnection to the physical world via sensors and actuators. Thus, there is a need to support features like discovery, description and event notification independent from the domain, application or underlying communication technology.

In the classical smart object systems, completely separate communication stacks have been developed for different communication links, which therefore stimulated the development of individual device networking protocols (discovery, description and event notification). Once IP is used on all smart objects, the technical need for diverse device networking protocols is removed. From an application point of view, there will be no difference in discovering an IP enabled smart object that is using Wifi or an 802.15.4/6LoWPAN based smart object. Therefore, a set of IP based, device networking protocols can be developed that support the features of discovery, description and event notification independent of the individual physical communication links, applications or domains. It will be possible to reuse these protocols in different smart object networks similar to the way that IP, UDP or HTTP can be reused.

In case no such common device networking protocols are established, different interest groups will start to develop application specific solutions for these generic problems, like done for the SEP2.0. The result will be a set of protocol suits with similar functionalities but incompatible specifications. This will again hinder the interoperation, like with the classical automation systems, as it creates an unnecessary diversity and thereby complexity for the use of smart objects. On the contrary, the establishment of common device networking protocols will reduce the delimiting factor between smart objects of different domains to the type of data and the device profile they are supporting. Also, the wider adoption of common protocols will provide an ecosystem with drivers, libraries and tools that can be reused in the different domains and applications. Thereby development and maintenance efforts as well as costs are reduced.

Scalability is important

In order to come to a common set of device networking protocols that can be deployed in a wide range of diverse application scenarios, it is necessary to address the underlying diversity of requirements with regards to e.g. cost, efficiency and battery lifetime. While some applications aim to comply with highest restrictions on energy consumption and cost (e.g. 8-16bit µC with 802.15.4 radios), others prioritize ease of installation and user interaction, using more capable microcontrollers and radios. Thus, it is important that the device networking standards are able to scale gracefully to reflect the requirements of different domains. For example, a discovery protocol should support simple, efficient plug and play mechanisms for small systems as well as the usage of management instances like discovery proxies to handle discovery in large scale deployments. Also, an event notification protocol is needed that allows delivery of events in a timely, efficient fashion and which also supports the management of complex, filter based event registration and leasing mechanisms for more complex smart objects.

Existing standards like UPnP or WS-DD (DPWS) have already defined solutions for device networking. They use either HTTP or SOAP driven approaches to create the necessary features of discovery, description and event notification and also address requirements with regards to flexibility. However, the cost for the flexibility is increased traffic and larger messages, which makes these standards unattractive for applications with focus on low cost and energy consumption.

Efficiency is important

On the other side, the current developments in the IETF CORE working group bring forth new approaches with the focus on efficiency. Most notably the combination of COAP and the CORE link format provides a set of device networking protocols for discovery, description and event notification. However, due to the focus on the highly, resource constraint systems, the required flexibility is sometimes reduced for the sake of efficiency.

One example for this is that the COAP option fields, that are the equivalent to HTTP headers, are only referenced by an index. For predefined, commonly used option fields like Content-Type, indexing is preferable as it significantly reduces the required amount of data to be transmitted in comparison to the ASCII encoding. However, the usage of application specific option fields is hindered by the indexing approach. In case an application specific COAP option is introduced, the application will depend on the correct option registry to be available in all smart objects it wants to use. While synchronizing the registries inside a closed system is possible, it will lead to significant problems if multiple applications and smart objects want to interoperate. In that case, different applications may have defined option registries with conflicting entries and therefore cannot co-exist in the same system. Similar issues may also arise when a COAP based smart object is accessed via a HTTP-to-COAP proxy, where HTTP headers may be included that have no equivalent standard COAP option.

This example shows the pitfalls of trading flexibility for efficiency. Luckily, the described issue can be solved by introducing an option field that allows ASCII encoding of application specific options. This can even be done efficiently without increasing the COAP message size for all messages where no application specific option is present.

Achieving scalability and efficiency

We see the need for a common set of device networking protocols that are flexible and efficient at the same time. This will be the main enabler for a wide adoption of these protocols. The key approach to come to the desired solution is to be as efficiency as possible without limiting the flexibility of the solution.

One good example in this direction is the 6LoWPAN standard, which provides full IPv6 compliance for 802.15.4 based smart objects. 6LoWPAN is able to compresses IP and UDP information significantly for packets that have particular characteristics. This allows efficient support of IPv6 even with the limited MTU of 802.15.4. However, there are IP packets that cannot be compressed by 6LoWPAN and that will be highly inefficient when transported via 802.15.4. However, these packets are still supported by 6LoWPAN and are simply forwarded rather than dropped. As a result, 6LoWPAN provides efficient IP communication for some type of IP packets and at the same time preserved the flexibility of IP.

We believe that there is the need for similar approaches for the device networking protocols, where particular characteristics in communication are leveraged to create efficiency, but where a comprehensive and flexible feature set is still maintained. Multiple techniques can be identified that support flexibility and efficiency at the same time. For example, an efficient encoding of XML data (like with EXI), preserves the flexibility in the design of data types while reducing the message size with an efficient, binary encoding. Similarly, COAP provides a good optimization of data transport, as it supports unreliable and reliable communication, without the overhead of the TCP 3-way-handshake. Also, optimization on the software for smart objects can be achieved, when contract first development is followed. Even if comprehensive device networking protocols are defined, a single smart object may only implement a subset of these protocols. The software which is developed for the particular smart object can therefore be tailored to the particular parts of the protocols that are required for the device, which simplifies the software and reduces memory and energy requirements. The approach of gSOAP is a prominent example for that.

In summary, we see the need to establish a set of common device networking protocols (discovery, description, event notification) that can be applied independent of application, domain or communication link. At the moment we see solutions like DPWS that provide the necessary flexibility but lack efficiency and the IETF CORE efforts that provide highly efficient solutions which lack extensive flexibility. We propose a combination of these solutions which should include the “best of both worlds”. To this end, we propose a COAP based SOAP binding, which will allow efficient, compressed XML based communication and which leverages the flexibility and comprehensiveness of the web services standards. First tests show, that such a binding leads to high optimization of message size and energy consumption. In addition, the support of the protocol is possible on 16 bit µCs like the MSP430, enabling also the adoption on constraint smart objects.