Unify to Bridge Gaps: Bringing XMPP into the Internet of Things

Michael Kirsche and Ronny Klauck
Computer Networks and Communication Systems Group
Brandenburg University of Technology Cottbus, Germany
eMail: {michael.kirsche, ronny.klauck}@tu-cottbus.de

***Abstract*—The Internet of Things vision states that sensors and actuators shall be integrated into the global Internet to facilitate an interaction with and integration of the physical environment. The development of enabling technologies like uIPv6 and 6LoWPAN provide the basic requirements for this interconnection. However, a seamless Internet-connection and interconnection between sensors and actuators can still only be provided with the help of protocols that use gateways, intermediate proxies, and protocol translators. We propose a solution to unify the world of sensors and actuators with the Internet through the use of the Extensible Messaging and Presence Protocol (XMPP) while omitting application protocol gateways and protocol translators at the same time. This article describes our ideas to boost the Internet of Things vision by using XMPP. We present our current work in progress and an outlook into our future working directions in this field.**

***Keywords*-XMPP; IoT; Seamless Interconnection.**

I. Introduction

The Internet has evolved from a small group of interconnected workstations to the world-embracing network of autonomous systems that we know and use today. The next logical step in its evolution is the interconnection with and integration of the real world, i.e. the physical environment we live in. This basic paradigm lies behind the Internet of Things (IoT) vision [1]. The IoT vision states that everyday objects and sensors as well as actuators shall be integrated into the global Internet, thus facilitating an interaction with the physical world. This concept of objects *(things)* being interconnected *(via ad hoc)* as well as being Internet-connected *(via infrastructure)* goes far beyond simple home automation applications like remotely-controlled temperature thermostats. It is rather a fundamental part of the pervasive and ubiquitous networking and computing concepts, as strategic projects like IBM's *A Smarter Planet* [2] show. A number of new applications will arise from this, ranging from simple warehouse management applications and new approaches for object awareness to self-reliant environmental monitoring and data collection without the need for human interaction. Benefits could be a broader horizon for our understanding and interaction with the physical world (through intelligent monitoring) as well as a reduction of waste and garbage (with clever resource management) or even just an increase in comfort (through smarter homes or computer-based aid in everyday tasks).

Still, most of these ideas and applications are glorified concepts and prospects of the impact and the possibilities of the Internet of Things. Pragmatically seen the current state of IoT development has not advanced that far. Researchers tried to bring the IoT vision closer to reality over the years by using a wide array of technologies, Wireless Sensor Networks (WSN) being one of them. Although WSN are not the sole ambassador of IoT enabling technologies, they tend to be one of the most promising ones, since they assist in monitoring and interacting with the physical environment. An essential research topic in this context is how to integrate sensor networks into the Internet. An important step here was the introduction of 6LoWPAN [3]. 6LoWPAN enables the use of IPv6 as a common protocol beyond network and technology borders. Another step was the development of uIPv6 [4], the world's smallest IPv6 stack. This combination of uIPv6 and 6LoWPAN enables interoperability between IPv6-powered sensors and IPv6 devices in the Internet; a basic premise for the IoT vision of interconnected objects.

While 6LoWPAN and uIPv6 enable communication on the lower layers, diverging application layer protocols still interrupt seamless interconnection between things. Protocols like the Constrained Application Protocol (CoAP) [5] or the Message Queuing Telemetry Transport (MQTT) protocol [6] were proposed to provide application layer solutions for addressing and managing resources in resource constrained networks. Both are counterparts to the Hypertext Transfer Protocol (HTTP), the de facto standard for request-response access to resources in the Internet. A seamless interconnection between Internet-based systems and sensor networks using CoAP or MQTT can only be managed over gateways or proxies. Using application protocol gateways is clearly a limiting factor, since they introduce additional complexity in terms of protocol mapping. They also break the end-to-end principle from a security [7] and protocol [8] point of view.

Another drawback is that CoAP is based on the request-response scheme. A topic-based publish-subscribe communication scheme is more suited for resource- and energy-constrained IoT applications, where things interact with each other based on their own context. We favour a solution where we register for events published autonomously whenever something happens instead of regularly requesting status updates, even in idle times. MQTT does provide such a publish-subscribe variation [9], with the disadvantage of

lacking support in the established Internet. Instead of creating another new protocol with additional gateways, intermediate proxies and lacking support, we want to facilitate one slim and trimmed protocol for both worlds that is established and proven to work. Our favoured choice is the Extensible Messaging and Presence Protocol (XMPP) [10], as it is already established, standardized, and freely available. We want to unify the Internet and the world of sensor networks by using XMPP as a basic service provisioning protocol. Service provisioning here means providing common services (e.g. unique addressing, message exchange, etc.), similar to the ones provided by HTTP/URI, but available and specifically adjusted for both worlds (Internet as well as IoT). We acknowledge the fact that protocols with slim footprints are necessary to support resource constrained devices, which are used in the IoT, but we want to refrain from using separated protocols to avoid the specified drawbacks. Instead of providing an overly complex all-in-one solution, we favour a building blocks concept, based on a slim yet powerful and configurable protocol, i.e. XMPP.

We want to contribute a downsized and trimmed version of XMPP to bring the IoT vision closer to reality. A comparable idea stands behind 6LoWPAN, where the powerful yet bloated IPv6 is reduced and compressed, hence facilitating IPv6 as the common protocol for the Internet layer instead of the previously necessary solutions with protocol gateways and bridges between WSNs and other networks. However, protocol unification should not stop at the Internet or transport layer. Web applications, for example, showed that a common and widely used basic service underlay boosts the deployment and pervasiveness of applications. We want to exploit the advantages of a common service provisioning protocol used both in the Internet and the IoT. Diversity at the application side should be supported by providing unified solutions through a single protocol instead of requiring several specifically tailored protocols.

This paper hence summarizes our ideas of promoting XMPP for IoT scenarios, with a description of approaches and advantages in Section II, and a roadmap of current and future work together with concluding remarks in Section III.

II. How XMPP Can Boost The IoT Vision

An idealistic view on realizing pervasive networking is given in [11]: "In an ideal world, we would have only one network management protocol for monitoring, alarming, configuration, and exchanging policy information, independently of the type of network (e.g. Smart Grid, IoT, wireless access or core network)." With XMPP as the underlying communication protocol for IoT we can get closer to this ideal, because application layer gateways can be omitted. The main goal for bringing XMPP into the IoT vision is to simplify the interconnection of devices [8] and to support Human-to-Machine (H2M) as well as Machine-to-Machine (M2M) communication between various device classes.

Initial steps of using XMPP for the IoT were the projects *uXMPP* [12] and *XMPPClient for mbed* [13], which provided rudimentary but lightweight XMPP implementations. These experiments showed that XMPP can be minimized to run on resource constrained devices while still being able to send messages and presence information to an XMPP client running on a PC, thus demonstrating interoperability. *OpenSpime* [14] is another approach that defines a set of XMPP Extension Protocols (XEP), including digital signatures, encryption, authority claiming, data reporting, and seeking for XMPP entities in a network. OpenSpime currently exists as a Python-based prototype, too heavy to run on resource constrained devices. These preliminary experiments provided initial insights but no conclusive scientific results. Other recent research proposes software architectures for smart environments to realize ubiquitous access to sensor data. Drawbacks of these ideas are the need for gateways interconnecting different networks [15] or the need for a middleware [16]. Disadvantages of using protocol gateways are that clients have to access sensor nodes over different techniques [15] and that they can cause operational problems while translations are often not 100% successful [17]. If a middleware is involved, all appliances depend on it during their interaction, whereas different versions of the middleware can disturb the cooperation. Sensor Web Enablement (SWE) [18], standardized by the Open Geospatial Consortium (OGC), represents another generic framework for a platform- and protocol-independent way of sensor interaction. SWE specifies a so-called *Sensor Web*, a complex middleware for sensor network management. Example deployments show [18, Sec. 4.1.] that complex scenarios can be realized, although large infrastructure support is required. We want to omit the need for intermediate systems and refrain from using complex middlewares. Through this work, we want to restart the discussion of using a slim and trimmed subset of XMPP for the Internet of Things, without the need for large infrastructures or protocol translators.

The advantage of XMPP as the default communication protocol for IoT is that an established and standardized protocol designed for real-time data streams can be used without the need for a middleware or protocol gateways. XMPP offers a rich variety of open source software for servers, clients, and libraries supporting several operating systems, ranging from desktop computers to mobile entities, thus easily connecting various devices and reducing developing and testing costs. Furthermore, the XMPP Standards Foundation offers a continuous maintenance for the XMPP protocol family, allowing system designers to benefit from the aspects of sustainability and expandability through protocol extensions (XEP). XEPs extend XMPP with additional capabilities so that it can simply be adopted to almost every scenario or environment. The expandability through XEPs is an important benefit of XMPP when compared to CoAP, which is explicitly designed for WSNs [5, Section 1] with a limited

set of methods, e.g.: `GET`, `POST`, `PUT`, `DELETE` [5, Section 5.8]. CoAP also depends on gateways to facilitate an Internet connection through HTTP [5, Section 8.1].

XMPP implements the publish-subscribe paradigm, a highly scalable bandwidth and energy efficient event distribution system [19] where only changes in sensed data are transmitted to registered receivers by decoupling the involved communication devices from each other. XMPP does not solely rely on its event-driven mechanism, it also supports a request-response scheme to ask explicitly for remote services or data. CoAP uses the Representational State Transfer (REST) architecture for asynchronous message exchange by actively requesting new data from involved devices. This data-polling can be stressful for a network when measurements are stable, because requests and responses still have to be exchanged without providing new information, thus wasting bandwidth and energy. Moreover, CoAP does not yet provide an adequate solution for end-to-end security in an IP-based IoT [7], while data access through XMPP is covered by today’s security standards (TLS/SSL). XMPP hence enables a secure, bijective and transparent access to various devices in different networks.

Our vision is the ubiquitous collaboration of XMPP-driven sensor nodes with standard applications and devices, thus facilitating the integration of WSNs into the Internet. As a result, each sensor node runs an XMPP software client on top of a IPv6/6LoWPAN implementation, through which it can publish measured data or receive control commands. Figure 1 depicts the appropriate architectural protocol stack.

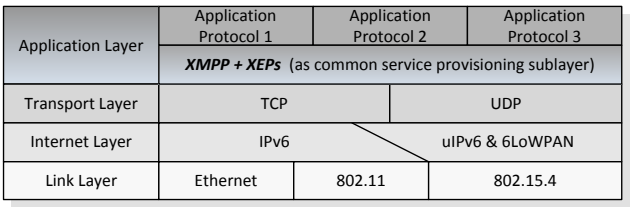

Figure 1. Protocol Stack with XMPP Service Provisioning Sublayer

A list of feasible services provided by the XMPP sublayer is exhaustive and application-dependent. One of the most common services is the neighbour entity detection, which we plan to provide with Multicast DNS (mDNS) [20] and DNS Service Discovery (DNS-SD). Each entity can thus detect other entities inside the network that support XMPP. Extending the network with additional sensor nodes is simplified due to the fact that all devices are able to explore their network vicinity for available XMPP services and servers and automatically start registering with them. Afterwards, each sensor device can publish its sensed data into the network or the Internet, depending on its connectivity status.

A noteworthy and important aspect of XMPP is the possibility to interact with a server infrastructure (XMPP Core) as well as the support for ad hoc/P2P communication. We plan to use *XEP-0174 Serverless Messaging* to exchange data in a P2P manor between entities in cases where no XMPP server is detected. All entities in an ad hoc network can hence share published services and benefit from each other. In scenarios where only one node has direct Internet access, it can offer an Internet sharing service and forward measured sensor data of the ad hoc group into the Internet. For both ways results can be displayed and monitored with XMPP chat applications, already available for most operating systems.

III. A Roadmap for an XMPP-powered IoT

This section summarizes necessary steps to fulfil the idea of bringing XMPP into the Internet of Things. In the first stage, we plan to analyse which minimal memory footprint is reachable for a minimized XMPP client that supports tiny embedded hardware platforms equipped with IPv6 stacks. The XMPP implementation needs to be extremely lightweight, because resource constrained embedded systems have small memory sizes and slow microcontrollers. We plan to use the uXMPP Contiki project as a starting point for research and acceptability testing, because it already implements rudimentary XMPP core functionalities, like presence and simple message exchange. Contiki [21] is a highly portable open source operating system for memory-constrained embedded systems and wireless sensor networks that already supports 6LoWPAN and IPv6. Our approach will be a reduction and optimization of XMPP together with mDNS on the top of an IPv6 stack for the use with constrained devices like WSNs, as an enabling technology for the XMPP-powered IoT vision. Therefore the most essential functions for typical IoT appliances and H2M/M2M communication need to be prioritised, mapped to existing or possibly new XEPs, implemented, and tested with a minimized XMPP client. First experiments showed that not all needed XEPs will fit into the limited memory of resource constrained devices (typically 128 KB of ROM), because of their initial design target (desktop systems with nearly no limit of bandwidth and hardware resources). Redesigning XEPs and splitting up functionalities might therefore be necessary. Following the building blocks concept, each function can be realized as an independent module and can be chosen as an optional feature to complete the XMPP Core functions as needed during compile time of the minimized XMPP client. Challenges are the implementation of XEPs as tiny modules with reduced code size and an economic use of message exchange (e.g. 127 Bytes max. packet size for IEEE 802.15.4) while extracting protocol behaviour unnecessary for constrained devices. Overall, the minimized XMPP client must still be standard-compliant while maintaining a low memory and bandwidth profile. This will enable an appliance and sensor specific protocol support, depending on the actual duties and scenarios while retaining the strong point of XMPP: its expandability though XEPs.

Furthermore, we plan to simulate mDNS in the second stage, to validate its scaling in large sensor networks, because of its necessity for the bootstrapping of the XMPP network. mDNS can resolve domain names without the help of any server. It is used by DNS Service Discovery (DNS-SD) to locate or to announce services of entities in a network. Through simulations we hope to identify bottlenecks and possible optimizations of existing implementations.

The third stage incorporates a closer comparison of WSNs and our XMPP-driven network to further analyse possible advantages of publish-subscribe based XMPP networks over polling and routing based wireless sensor networks.

Since energy is a major concern for constrained networks, we plan to measure the influence of the size and number of XMPP messages in comparison with other approaches (e.g. CoAP). To facilitate comparisons over a longer time scale, we plan to contribute by integrating necessary protocols into simulation environments, thus simplifying analysis and comparison. We try to depart from the typical approach here. Normally, protocols or phenomena are modelled from scratch for use in simulation environments, hence creating an abstract version of the real protocol or physical phenomena. To omit simplifications and abstractions at this point, we plan to use either low-level cross-layer simulators or to integrate existing protocol implementations seamlessly into higher-level simulators. The latter option would enable simulations on various scales, ranging from simulating single devices with a high accuracy (comparable to low-level simulators due to the use of real-life protocol implementations) to simulations of complete networks or autonomous systems. Both options (low-level cross-layer vs. higher-level simulation with integrated real-life protocols) need to be examined further to facilitate a decision between them.

Finally, we can summarize that our vision breaks with the current way of looking into the topic of integrating WSNs into the Internet of Things. We intend to boost the IoT vision by using XMPP to overcome the current intermission due to the need for gateways and workaround solutions. We presented our basic concept for an XMPP-powered IoT vision together with a short roadmap to highlight our future work and planned contributions.

References

[1] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, July 2009, http://www.rfidjournal.com/article/view/4986.

[2] IBM, "Smarter Planet," www.ibm.com/smarterplanet, 2011.

[3] Z. Shelby and C. Bormann, *6LoWPAN - The Wireless Embedded Internet*. John Wiley and Sons Ltd, 2009.

[4] M. Durvy, J. Abeille, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, and A. Dunkels, "Making Sensor Networks IPv6 Ready," *Proc. of the 6th ACM Conference on Networked Embedded Sensor Systems (SenSys 2008), Poster session*, November 2008.

[5] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP)," IETF, http://tools.ietf.org/html/draft-ietf-core-coap-08, Internet-Draft, Nov. 2011.

[6] IBM, "MQ Telemetry Transport," http://mqtt.org/, 2011.

[7] M. Brachmann, O. Garcia-Morchon, and M. Kirsche, "Security for Practical CoAP Applications: Issues and Solution Approaches," *Proc. of the 10th GI/ITG KuVS Fachgespraech Sensornetze (FGSN11)*, September 2011.

[8] F. Mattern and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," in *From Active Data Management to Event-Based Systems and More*, ser. Lecture Notes in Computer Science, K. Sachs, I. Petrov, and P. Guerrero, Eds. Springer, 2010, vol. 6462, pp. 242–259.

[9] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S A Publish/Subscribe Protocol for Wireless Sensor Networks," *3rd Int. Conference on Communication Systems Software and Middleware (COMSWARE 2008)*, pp. 791–798, 2008.

[10] P. Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core," IETF, Request for Comment 6120, Mar. 2011. [Online]. Available: http://www.ietf.org/rfc/rfc6120.txt

[11] M. Ersue and J. Korhonen, "Position Paper on Interconnecting Smart Objects with the Internet," [Online] http://www.iab.org/wp-content/IAB-uploads/2011/03/Ersue.pdf, February 2011.

[12] A. Hornsby and E. Bail, "uXMPP: Lightweight Implementation for Low Power Operating System Contiki," *Int. Conference on Ultra Modern Telecommunications (ICUMT)*, 2009.

[13] M. Schulz, "mbed Cookbook - XMPPClient," Jan 2011. [Online]. Available: http://mbed.org/cookbook/XMPPClient

[14] R. Ostinelli, "Internet Of Things: Vision, Prerequisites and OpenSpime," 2009. [Online]. Available: http://www.slideshare.net/ostinelli/my-pres-1518858

[15] M. Isomura, C. Decker, and M. Beigl, "Generic Communication Structure to Integrate Widely Distributed Wireless Sensor Nodes by P2P Technology," *Proc. of the 7th Int. Conference on Ubiquitous Computing*, Sep. 2005.

[16] J. Kusznir and D. J. Cook, "Designing Lightweight Software Architectures for Smart Environments," *Proc. of the 6th Int. Conference on Intelligent Environments (IE 2010)*, 2010.

[17] J. Schoenwaelder, T. Tsou, and B. Sarikaya, "Protocol Profiles for Constrained Devices," [Online] http://www.iab.org/wp-content/IAB-uploads/2011/03/Schoenwaelder.pdf, 2011.

[18] A. Broering, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, and R. Lemmens, "New Generation Sensor Web Enablement," *Sensors*, vol. 11, no. 3, pp. 2652–2699, 2011.

[19] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys (CSUR)*, vol. 35, no. 2, pp. 114–131, June 2003.

[20] S. Cheshire and M. Krochmal, "Multicast DNS," IETF, Internet-Draft, Feb. 2011.

[21] "The Contiki OS," [Online] http://www.contiki-os.org/, 2011.