Interoperability Concerns in the Internet of Things

Jari Arkko
jari.arkko@ericsson.com
Ericsson Research, Jorvas, Finland

Position paper for the IAB Smart Objects Workshop, Prague, March 25th, 2011

Abstract

This paper discusses the interoperability challenges in networks dedicated to the observation and management of objects in the physical world, the "Internet of Things". Technical constraints, chosen implementation techniques, and user requirements make interoperability hard in these networks. This paper reviews the reasons for the challenges and suggests some approaches to mitigate them.

Introduction

A key feature of the Internet is that different devices can work together: any browser works with any web server, almost all content is viewable by all devices, any device can plug into a home router, different networks can exchange routing information with each other, and so on. As the Internet has evolved, *interoperability* has always been a major concern, in terms of protocol design and extensibility, building products that in practice work well together with other devices, and setting standards.

In general, today's Internet builds on a key set of protocols that work extremely well between different types of devices and in varying types of networks: IP itself, TCP, DHCP, DNS, HTTP, TLS, HTML, XML, and so on. But in many cases there are still some components in the protocol stack that are proprietary, application-specific, available for limited platforms, or come from a single source. For instance, specific link layer technologies, content formats (Flash), or applications (Skype). This shows that it is important to balance the need for an interoperable Internet with the need to allow commercial innovation. Still, it is expected that over time, generally interesting components become available for all devices that need them.

Nevertheless, today's Internet is primarily characterized by applications with a human in the loop. A successful Internet application is one where the desired human experience is achieved. For instance, the desired visual effect is correctly rendered on screen. This makes interoperability a bit easier, as the humans are responsible for processing the "semantic" part of the communications. Today's Internet also consists of a relatively homogeneous set of devices. While there are differences between a smartphone, a laptop, and a high-end server, for instance, they are all still high power computing devices.

The emerging Internet of Things is expected to have new technical solutions, different types of devices, new requirements, and new usage patterns that will change the Internet. Some of the new requirements and expected usage patterns will cause interoperability challenges. For instance, there will be

- a capability mismatch between traditional Internet hosts and small devices,
- widely differing communication and processing bandwidths in different devices,
- needs for interoperability at a semantic level,
- different internetworking protocol choices (legacy vs. IP vs. IPv6), and
- solutions that are suitable for only some networks.

The rest of this paper discusses these different areas of concern. We conclude with some recommendations on how these concerns can be alleviated.

Capabilities

The desire to build large numbers of small, battery-operated, and inexpensive devices drives the need for simple solutions. Often these devices are not easily software upgradable, and their protocol and application suite is limited. Some of the typical limitations include:

- MTU limitations,
- simplified web protocols (COAP/UDP instead of HTTP/TCP),
- single-stack instead of dual-stack,
- limited or no support for security that would be suitable for operation over the Internet,
- limited communications bandwidth,
- limited processing power for large numbers of messages,
- sleep schedule that does not allow for communication at all times,
- and so on.

These limitations would have no effect if the device only communicated to other similar devices, but they do have an effect when attempting to provide *Internet-wide interoperability* to such devices. For instance, clients that today employ HTTP would be unable to communicate with such a device. We believe that Internet-wide interoperability is required, as the system of connected devices usually consists of sensors, actuators, user interfaces, servers, and other components. Many of these components are expected to be devices in the traditional Internet. For instance, it is likely that computers and smartphones are used as the user interface for controlling many Internet of Things applications.

It is important to note that some of the capacity requirements would preclude direct communication to an Internet of Things device even if implemented exactly the same protocol stack as other devices in the Internet. For instance, a sensor whose value is interesting to a large audience may not be able to accommodate all requests.

Semantic Interoperability

Most Internet applications designed for humans often require only transport of data from one place to another, and an accurate rending of that data on the screen. It is not necessary to process or understand the data in any semantic manner.

Much of the current focus in the Internet of Things is also on the lower parts of the stack: designing the wireless networks, running IPv6 over them, getting routing to work, and using UDP/TCP and COAP/HTTP.

It is important to realize that this is **not** enough for true interoperability. For instance, it would not be enough for a light switch from one vendor to control lights from another. For true interoperability we need *semantic interoperability*, the ability of the devices to understand what the data they communicate **means**. Most often this would imply standardizing not just the protocols and data formats, but also the meaning of the data, e.g., that “1” in a particular field means that the light should be switched on. Standardizing the meanings is difficult and time consuming, however. It has to be done on a per-application basis and with application specific expertise.

There are of course different ways of achieving semantic interoperability. This does not always involve standards. Devices could accept program code that performs the required actions. For instance, a light switch might accept a program fragment from a light bulb to run the user interface necessary to control the light. This is similar to how Flash-based applications can support new video codecs without requiring

support from the browser or any Internet-wide agreement about the new coding format. It remains to be seen if programmable control models become popular in the Internet of Things.

Nevertheless, there should be some way for the light switch and the light bulb to agree how the lights are turned on. This is not to say that there is no benefit from an Internet of Things without it. There will always be a need for some proprietary or leading edge, non-standard communications. And even if none of the application layer communications would interoperate with each other, we would still have a common backbone for the Internet of things, consisting of the IP layer, routing, COAP/HTTP proxies, and so on. We call this the *Internet of Things transport network*. This would be tremendously valuable. But it would not enable an Internet of Things where any light works with any switch or any energy meter works with any provider's server.

Authorized Interoperability

There are of course a number of security related challenges in the Internet of Things as well. Many of these fall into the capabilities category. Small devices may not have a suite of security tools that are commonly used in the Internet today.

But there is another, more fundamental issue. It is not enough that two endpoints support the same security mechanisms. The communicating parties also have to share some type of relationship that allows them to *authenticate* each other and *authorize* whatever actions are taking place. There are many ways to implement this, for instance with shared secrets, trusted third parties, or certificate infrastructures. It is relatively straightforward to set this up in small networks or within a single organization. Setting this up in a larger scale or in situations that require multiple participating organizations is going to be harder. It remains to be seen to what extent Internet of Things applications require such more complex setups, but there seems to be at least some situations where ability to setup security relationships between multiple organizations would be useful. For instance, home owners, manufacturers, and electricity utility companies might all want to control a particular home appliance.

Network-Specific Solutions

The Internet of Things is pushing the limits of technology in many areas. As we approach those limits we need to apply optimizations and design techniques to make our technical solutions feasible. But at the same time this may make our solutions less general than we would wish. For instance, the RPL routing protocol has two modes optimized for different types of networks. Those modes are necessary, because without them its not possible to support some important applications. However, the modes are incompatible and highly optimized implementations are unlikely to support both. As a result, interoperability is not assured merely through the use of the same protocol. Note that while we use the two modes from RPL as an example, many similar issues exist elsewhere as well (different metrics in RPL, different header compression types in 6LOWPAN, XML vs. JSON vs. binary XML in sensors that use COAP, and so on).

Conclusions

Moving legacy and application specific solutions to IP is a necessary and useful step. However, it is only the first step in ensuring a truly useful Internet of Things where different objects seamlessly communicate with each other. Some of the key areas where further work is needed include:

- standardization of application specific messages and semantics,
- standardization of suitable data formats,
- overcoming capability related interoperability issues,
- the use of gateways and proxies as opposed to mere end-to-end communication,
- development of middleware to handle distribution, storage, and conversion of data flows
- general tools to interconnect to social media, messaging, and other systems
- solving the common authentication and authorization infrastructure problem,
- further work on ensuring that each individual protocol specification is interoperable in all situations.

Looking back at the development of the Internet, one of the lessons that we can draw from it is to ensure that we have sufficiently general mechanisms that address most needs. Highly optimized and specialized solutions have rarely succeeded. Robustness and generality are often more important than mere performance.