A Gateway Architecture for Interconnecting Smart Objects to the Internet

Authors:

Akbar Rahman (Akbar.Rahman@InterDigital.com), Dorothy Gellert (Dorothy.Gellert@InterDigital.com), Dale Seed (Dale.Seed@InterDigital.com)

Abstract

The majority of Smart Objects are constrained by some aspect of their performance (e.g. power, CPU, memory, etc.) and may not have the capability to directly connect to the Internet. A gateway (or proxy server) becomes a key requirement for interconnecting a local Smart Object subnet to the Internet. A Smart Object Gateway architecture allows for efficient service delivery between the Smart Object and an end point on the Internet such as an application server. This paper describes the key characteristics and benefits of this approach, leveraging the IETF low power and constrained node protocols, and the ETSI M2M model. Finally, an end-to-end Smart Object network implementation that interworks between different technologies and network domains is described.

Introduction

Gateways are commonly used today for many services in the Internet. Note the terms gateway and proxy are used interchangeably in this paper. The most common use of a gateway today is for World Wide Web (WWW) traffic and is called a web proxy. A web proxy is located between a web browser client and a web server. A web proxy can perform a wide variety of functions including caching, media type transformation, protocol reduction, and anonymity filtering [RFC2616].

In a similar fashion, the Smart Object Gateway performs the following key functions on behalf of Smart Objects:

- Network connectivity (i.e. interconnection to the Internet and other networks)
- Protocol translation (e.g. CoAP to/from HTTP)
- Caching (for incoming traffic to Smart Objects that are in a power save state)
- Multicast (i.e. fan out of incoming traffic to multiple Smart Objects)
- Service enablement (e.g. naming, registration, management)
- Security (e.g. boot strapping, and authentication proxying to allow Smart Objects to register to backhaul networks such as cellular)

Architecture

It is expected that Smart Objects are often grouped together in clusters of various sizes which may be considered subnets. Figure 1 illustrates an example of a typical Smart Object deployment in a home environment. Due to the constrained natured of the devices, it is expected that the Smart Objects may not have the capability (e.g. full IP protocol stack) to connect directly to the Internet.

Instead, the Smart Objects will first organize themselves to gain local connectivity. This may involve, for example, the formation of a Wireless Personal Area Network (WPAN), or use of wired connections (e.g. power line communication). A key part of this local connectivity is the connection to the Smart Object Gateway. The Smart Object Gateway provides the physical, protocol and service capability to interwork the local Smart Objects with the rest of the Internet. For example, the Smart Objects may have only WPAN physical connectivity. They rely on the gateway to provide a fiber optic, DSL, cellular or other physical connectivity to the Internet.

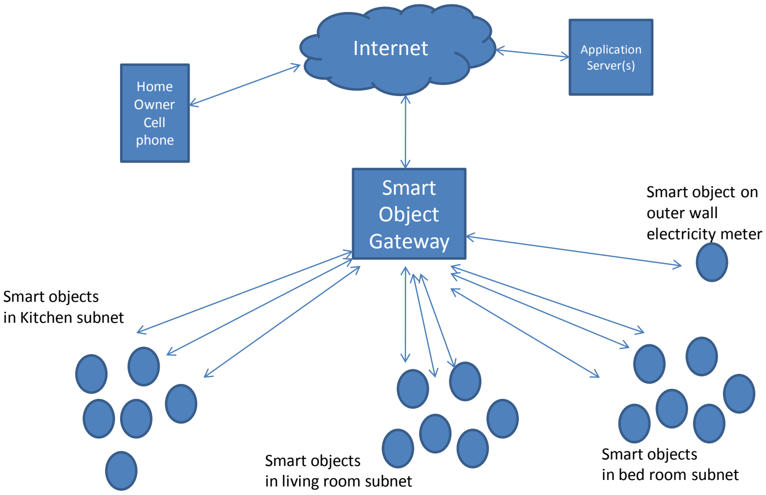
Internet
Application Server(s)
Home Owner Cell phone
Smart Object Gateway
Smart object on outer wall electricity meter
Smart objects in Kitchen subnet
Smart objects in living room subnet
Smart objects in bed room subnet

Figure 1 -
Smart Object Gateway Architecture in Home Environment

Network implementation

A Smart Object network was implemented as shown in Figure 2 based on the IETF {CoAP, ROLL, 6LoWPAN} protocols and the ETSI M2M model [ETSIM2M]. Note the interface identification (e.g. dla, mld) in the figure follows ETSI M2M nomenclature. Multiple access technologies are integrated into the network including 802.15.4 (Smart Object WPAN) and 3G cellular (backhaul connection to Internet). The Smart Object Gateway plays a key role in the network by connecting the Smart Objects to the Internet, and allowing the objects to participate in end-to-end services (e.g. management) with the remote server. The gateway also caches incoming traffic for the Smart Objects when they are asleep. The traffic is subsequently forwarded to the Smart Objects as they became active.

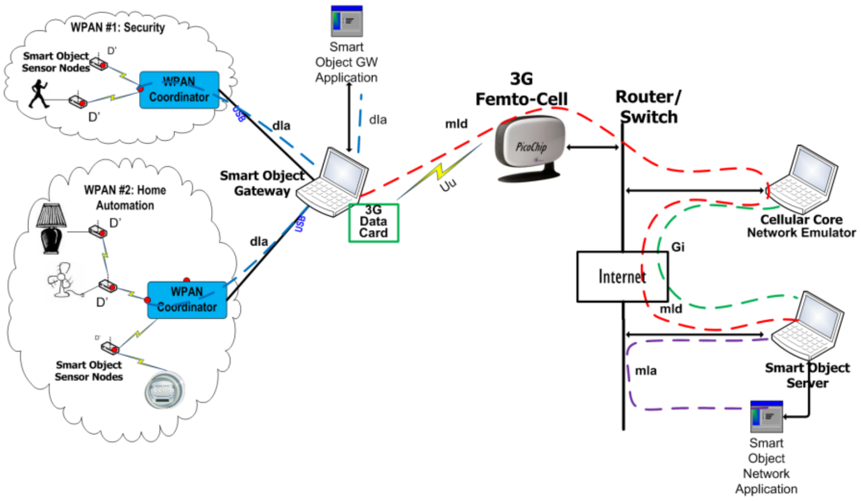

Figure 2 – End-to-End Smart Object Network Implementation

The protocol stack implementation is shown in Figure 3. End-to-end application layer services are supported. For example, the Smart Object initially registers with the network server. The Smart Objects are then managed remotely through the Internet from the server. An option is also available for the Smart Object Gateway to provide some or all of these services (in lieu of the remote server). The underlying protocol implementation includes IETF CoAP between the Smart Objects and the Smart Object Gateway. The Smart Object Gateway then performs CoAP-HTTP protocol conversion to communicate with the server in the Internet.

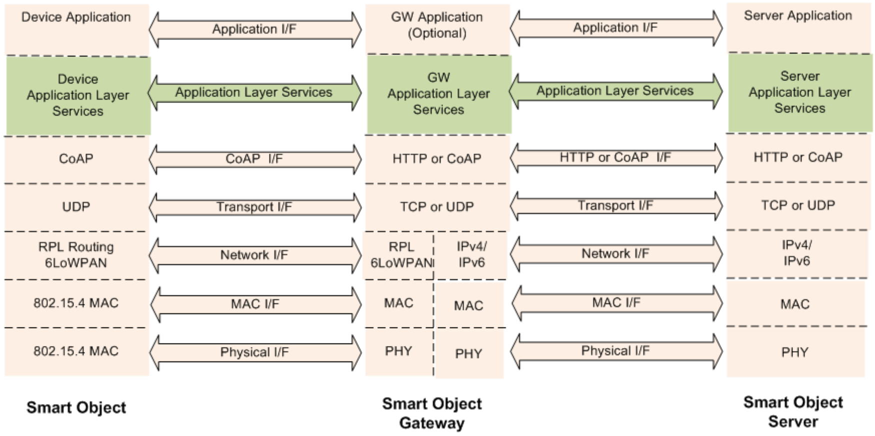

Figure 3 – Smart Object Network Protocol Stack Implementation

Conclusion

Smart Objects provide key functionality such as sensing and actuating, but may not have the capability to directly connect to the Internet due to their constrained nature. A Smart Object Gateway architecture allows the Smart Objects to work cohesively by interconnecting the objects to the Internet and enabling end-to-end services.

Future challenges to consider for Smart Objects include automated discovery and secure boot-strapping of Smart Objects onto operator and/or service provider networks. Other challenges are streamlining of existing management protocols for Smart Objects, and mobility of Smart Objects.

References

[RFC2616] Fielding, R., Gettys, J., Mogusl, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1", RFC 2616, June 1999.

[ETSIM2M] "Machine-to-Machine Communications (M2M); Functional Architecture", ETSI TS 102 690 (Ver. 0.9.6), December 2010.