Software Updates for Wireless Connected Lighting Systems: requirements, challenges and recommendations

Luca Zappaterra, Esko Dijk
Philips Lighting Research - Eindhoven (NL)
Email: luca.zappaterra@philips.com

***Abstract*—Wireless Connected Lighting Systems (WCLSs) are wireless mesh networks to provide lighting leveraging a large number of distributed embedded devices with LED lighting, sensing, processing and communication capabilities. In a WCLS, software updates (which include updates to operating system code, libraries and software components) are delivered to the nodes to patch security vulnerabilities, improve software stability and enhance/add features. These updates are typically carried out Over-the-Air (OTA), leveraging the existing wireless communication infrastructure. In this position paper the software updates requirements for WCLSs are firstly introduced, emphasizing the importance of delivering updates in an automated, consistent and efficient manner, while minimizing the interference with regular lighting operations. Then, available solutions and existing challenges occurring when distributing software updates in a WCLS are presented. Finally, recommendations are provided to assist the future design of optimized software updates solutions for WCLS and possibly other similar large scale wireless networks of lightweight embedded devices.**

I. Introduction

The ongoing transition from legacy lighting technologies such as incandescent, fluorescent and discharge light sources to solutions based on Light-Emitting Diode (LED) technology is offering the unique possibility to replace existing light-points with advanced digital LED light-points equipped with sensing, computational and wireless communication components. Thus, all light-points effectively become digital smart devices capable of capturing, processing and transmitting data in near real-time. These can be integrated with a management platform to form a Wireless Connected Lighting System (WCLS) which can provide incremental value beyond standard lighting solutions including automatic, personalized and localized lighting control, together with energy savings. The ubiquity of lighting infrastructures will soon permit WCLSs to be implemented everywhere in a cost-effective manner, leveraging the ongoing transition to advanced LED lighting solutions. An example of WCLS operating in an indoor space is illustrated in Figure 1. The system consists of a large number of connected light-points, sensors and switches which are quite often identical embedded devices with same firmware for the same type of device. These devices together form a local wireless mesh network (e.g. leveraging ZigBee or equivalent technologies) to exchange data among themselves, possibly using a gateway to interface with the Internet. A server may also be present in the building for local management purposes.

Software updates are often needed to be delivered to the embedded devices in a WCLS for different purposes which include: fixing of security holes and privacy issues, improvement of software stability and enhancement/addition of product features. For lightweight platforms such as the embedded devices considered in a WCLS, software updates consist of an actual replacement of the existing software including operating system code, software components and libraries. The authenticity of a software update data must be verified before installing it into the devices. A system reboot is then required after the installation.

In this context, the specific features of a WCLS may impact the requirements for software updates in two ways:

- Software updates should not worsen the lighting experience provided by a WCLS, as humans are very sensitive to the variations of light and any degradation of the lighting functionalities would strongly affect the user experience.
- Software updates need to be performed in a consistent, automated and efficient way. In fact, WCLSs are characterized by hundreds or even thousands of lighting devices deployed in a limited geographic areas such as buildings or public spaces, which require consistent software to be installed across devices. The process cannot rely on human intervention because of the large number of devices and the difficulty to physically reach them.

II. Existing solutions and challenges

Software updates in WCLS need to be carried out Over-The-Air (OTA), leveraging the wireless technology available for communication to securely deliver the software update without physically accessing each lighting device in the network.

The communication technology solutions implemented in a WCLS need to support OTA software updates, while meeting the specific requirements of lighting applications identified above. Here, wireless mesh network solutions based on Standard IEEE 802.15.4 technology are considered, due to its suitability and widespread adoption in the field of lighting applications.

The ZigBee technology suite (which leverages IEEE 802.15.4 PHY and MAC technology) provides standard application layer support to deliver firmware image updates to

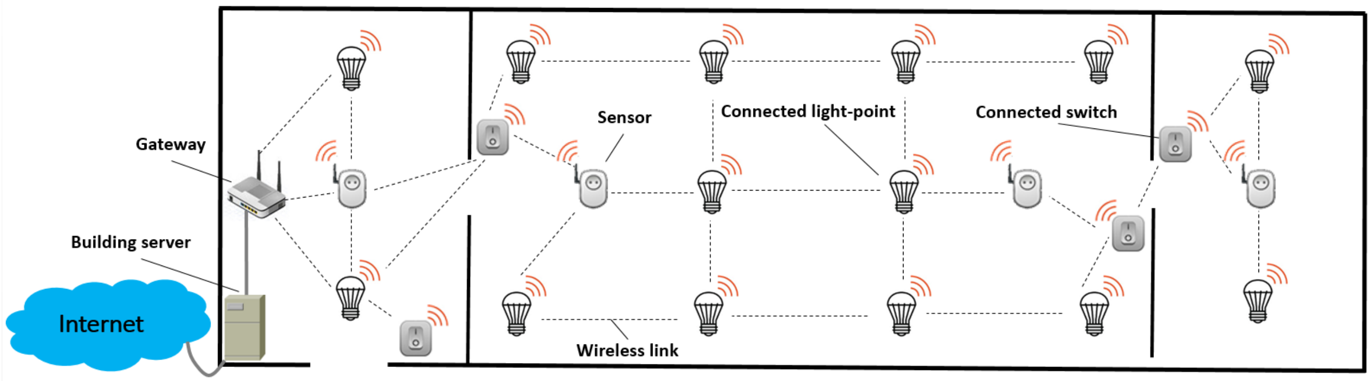

Fig. 1: An example of Wireless Connected Lighting System (WCLS) for indoor space.

devices in the ZigBee network, as described in the Over-The-Air Upgrade Cluster specification [1], part of the ZigBee Smart Energy Profile. The specification details standard procedures to securely provide an OTA software update from a server device implementing the updating service to a client device receiving the update. Architecture and implementation details to carry out software updates in a ZigBee network are not specified and left up to the implementers. As a consequence, proprietary solutions have emerged to implement OTA software updates in multiple nodes of a mesh network based on 802.15.4 technology. An example of implementation has the network gateway sending either unicast, multicast or broadcast messages to update individual nodes, group of nodes or all the network at the same time respectively, as described in [2]. Updates for nodes which are not directly connected to the gateway are forwarded by intermediate nodes. Other solutions [3] support the update of nodes in both direct link and multi-hop configurations, but only updating one device at a time. The whole process of updating the software for all the nodes in a large wireless mesh network can take several minutes or even hours as the gateway will need to query the network for nodes ready to be updated, deliver the data content to the devices and finally receive the confirmation of successful update by the target nodes.

Existing solutions for OTA software updates may not meet the requirements of WCLSs, namely no degradation of the lighting experience and consistent, automated and efficient distribution of software. For example, updating the software of each lighting node at a time via unicast may not be efficient since the process would need to be repeated for all the nodes in the network, possibly incurring large delays which can compromise the security and the lighting experience of a WCLS. In fact, the gateway connected to the Internet can only supply a data rate of less than 250 Kb/s to the wireless mesh network, as limited by the 802.15.4 PHY. Therefore the update of a large number of nodes can incur very large delays if done via unicast. Since it is expected that software updates in a WCLS will be carried out frequently both to fix software bugs and to provide new/improved features for customers (as seen happening for example in the "apps" and operating systems in the mobile market), then if critical software updates are not delivered promptly to all nodes, the security of the WCLS can be compromised. At the same time, if a software update to improve a lighting feature is not delivered in a consistent and timely matter to all the light-points of a WCLS, the lighting experience may be impacted.

Data broadcasting represents a more efficient solution to update the software of a large number of lighting nodes in a WCLS. Nonetheless, the widespread ZigBee network broadcasting protocol may incur inefficiencies when implemented in a dense mesh network such as a WCLS. In fact, when a broadcast frame is transmitted from a ZigBee device, all neighbors within communication range will re-broadcast the frame, eventually generating excessive traffic which can saturate the network. This behavior can result in inconsistent software updates across the nodes of a WCLS (when broadcasts collide with each other) as well as detrimental impact to the regular lighting communications (when broadcasts collide with existing lighting traffic). A more efficient solution to distribute data to all nodes in a wireless mesh network is detailed in the Multicast Protocol for Low power and Lossy Networks (MPL) specification [4]. MPL has been incorporated to support broadcast and multicast data traffic (including software updates) in the new Thread technology [5], which defines protocols to operate a secure IPv6 wireless mesh network on top of existing Standard IEEE 802.15.4, optimized for home automation and control applications such as lighting. MPL is based on the Trickle algorithm [6], which is specifically designed for propagating and maintaining consistent software updates in large wireless mesh networks. The Trickle algorithm implements a "polite gossiping" strategy, which limits the amount of broadcast data forwarded by a node when sufficient copies of the same data have already been received by the device. This solution allows to implement efficient and reliable broadcasts in a dense multi-hop network by achieving: 1) constant broadcast packet transmission per area, independently of nodes' density; 2) desired target reliability by adjusting the broadcast transmission per area rate.

In addition to the technical challenges discussed above, a concerning issue regards the potential conflicting approaches on the application of software updates to the lighting devices in a WCLS. In fact, multiple entities such as the manufacturer of the embedded lighting devices, the provider of the lighting service and the system administrator of the WCLS may want to have control over the methodology and scheduling to adopt. For example, the system administrator (or the building manager) may want to influence the type and occurrence of the software updates to be applied to the system, in general preferring to postpone (or omit) the application of software updates in favor of preserving the stability of the system. Oppositely, device manufacturers may want to deliver software updates as soon as possible, for example when a security vulnerability has been discovered. In this context, the approach of postponing (or not applying) updates to favor system stability may compromise security, whereas the approach of pushing software updates as soon as possible may compromise the functioning and therefore the lighting experience provided by a WCLS.

III. Recommendations

The following recommendations have been derived in line with the requirements and challenges illustrated above. They are meant to provide guidelines for the design of software updates solutions for WCLS and possibly other similar large scale wireless mesh networks of embedded devices.

- Standardized solutions can facilitate the distribution of software updates in large scale wireless networks of varied embedded devices, such as WCLSs. These solutions need to provide the flexibility to be customizable for the different existing and future use cases, which are characterized by varying security and system stability requirements. For example, critical security updates should be distributed to the network as soon as possible to patch security holes, as opposite to feature enhancement updates which can be postponed to periods which have minimal interference with the regular system operation. Also, the criticality of security updates and the importance of a feature enhancement updates can vary among different applications and should therefore be accounted in the design of these solutions.
- The scheduling of software updates need to be designed in accordance with the tolerable level of service disruption of the specific target application. As an example, the application of software updates to hundreds of lighting devices should be preferred not be performed at times of high building occupation and human activity, since the software update traffic may affect the regular operations traffic, possibly resulting in a severe degradation of the lighting experience. As an opposite example, the general functionality of a large wireless sensor network in a greenhouse will be less impacted by the possible service disruption due to the application of software updates to its nodes, which therefore could be applied at any time.
- Protocols such as MPL, designed for the distribution of data to large number of nodes in a mesh network, are recommended to be used in WCLS and similar systems. With these protocols, it should be possible (as it is in MPL) to tune reliability and network load of the software update distribution in accordance with the specific use case requirements, eventually guaranteeing timely and consistent delivery of software updates while minimizing the impact on the regular operation of the system.
- The OTA software updates process could benefit from automation, aligned with the specific application requirements, such that minimal intervention is needed by entities such as the manufacturer of the embedded lighting devices, the provider of the lighting service or the system administrator. In this way, potential conflicts which arise from the different influences these entities could have on the execution of software updates can be minimized.
- Methods to deliver and install individual software components into the embedded devices could be adopted as a way to reduce the excessive data transfers required today for software updates in a WCLS or similar large scale wireless mesh networks. In fact, the embedded devices currently used in WCLSs do not support the installation of single software components, but in the future they could greatly benefit from this feature as seen happening for “apps” in the mobile market.

References

[1] *Standards: ZigBee OTA Upgrade Cluster Specification*, ZigBee Alliance. [Online]. Available: http://www.zigbee.org/download/standards-zigbee-ota-upgrade-cluster-specification/

[2] Libelium, “Waspmote Technical Guide.” [Online]. Available: www.libelium.com/downloads/documentation/waspmote_technical_guide.pdf

[3] Silicon Labs, “AN760: Using the Standalone Bootloader.” [Online]. Available: http://www.silabs.com/Support%20Documents/TechnicalDocs/AN760.pdf

[4] J. Hui, R. Kelsey, “Multicast Protocol for Low power and Lossy Networks (MPL).” [Online]. Available: https://tools.ietf.org/html/rfc7731

[5] Thread Group, “Thread Technical Overview,” 2015. [Online]. Available: https://www.threadgroup.org/Portals/0/documents/resources/Thread_Technical_Overview.pdf

[6] P. Levis, N. Patel, D. Culler, and S. Shenker, “Trickle: A Self-regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks,” in *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, ser. NSDI’04. Berkeley, CA, USA: USENIX Association, 2004, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251175.1251177