sDOMO communication protocol for home robotic systems in the context of the internet of things

Marcel-Titus Marginean[1], Chao Lu
Computer and Information Sciences
Towson University, Maryland, USA
Email: [1]mtm@mezonix.com, clu@towson.edu

sDOMO is a communication protocol developed for home automation and the building of robotic systems. Being optimized for small devices, sDOMO allows some 8 bit microcontrollers to be a full featured, independent node in the domotic network, yet the protocol is powerful enough to provide soft-real-time communication for our computer-vision distributed system for domestic robots. Being designed with security and privacy concerns in mind, sDOMO has unique features to protect the residents.

Keywords: Home Automation; Domestic Robotics; Communication Protocol; Privacy.

1. Overview

Presented in our previous paper [1] sDOMO is a communication protocol developed originally to support our Distributed Processing Architecture for Domestic Robots [2] and during the development process, it was optimized for small devices transforming it into a general purpose soft-real-time protocol for domotics, and building of robotic systems. sDOMO uses low overhead binary data packing that can be implemented by very small, microcontroller based devices, and proposes a stand-alone integrated domotic system with the actual robots being just the mobile components of the smart house of the future. This distributed architecture, takes advantage of the already existing house network and provides by protocol design advanced security and privacy features intended to protect the residents.

sDOMO protocol from the beginning was designed to be implemented in a self-contained home automation and robotics system, and its design reflects this purpose. During development, it was observed that the protocol could be easily extended to any type of system integrating multiple devices (sensors and actuators) that need to behave plug-and-play; therefore, it is proposed also as a protocol for building robots and unmanned vehicles out of generic parts.

1.1 *sDOMO at a glance*

The typical architecture of a domotic network in the sDOMO model consists of a set of devices, a House Hub, Home Intelligence Unit, and an Internet Gateway. A device can be any functional node of the home automation network and is identified by a unique ID.

House Hub performs the function of a Rendezvous Server accepting and configuring devices in the network and routing sDOMO's messages between devices.

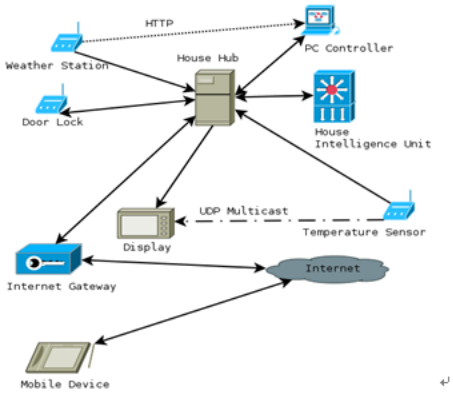

Fig 1: sDOMO System Architecture

Devices communicate with the Hub via sDOMO Packets. They are packets for Discovery and Configuration, Session Control and Message Carrying Packets. Packets are carried inside the datagrams as provided by the underlying network technology. In the case of Ethernet and Wi-Fi, the UDP/IP datagrams are chosen to carry packets. From a software point of view, a device is composed from one to up to 255 software Objects. The Objects are the sender and target for Messages. The preferred method of data exchange in an sDOMO system is by passing Messages between devices routed by the House Hub. This allows small, microcontroller devices to be part of the network while benefiting from all the security and privacy features the House Hub can offer. Messages are binary chunks of data of up to 4GB in size subject to limitation of the device and Hub memory. Messages are carried between the Devices as one or more sDOMO Message Carrier Packets routed by the Hub.

There are 2 categories of messages exchanged by devices via the Hub: Direct Messages (one to one communication) and Notifications (one to many). A Direct Message is sent from an Object part of a device specifically toward a given object part of another device. Notifications are sent and usually triggered

by external events, and are delivered by the Hub to all devices that have at least one object subscribing for them.

Besides communicating via sDOMO messages routed by the Hub, the model also allows devices to communicate directly using their own protocols, which is outside of sDOMO specifications, therefore allowing a transition path toward sDOMO. However, devices communicating that way must implement their own security and privacy mechanisms. A device using its own protocol can still use the House Hub as a simple Rendezvous Server in order to discover the other devices in the network. For the rest of this section, we will speak only about communication that takes place via sDOMO messages routed by the Hub.

As start-up devices advertise themselves and get invited by the House Hubs to join their network, they get configured and receive a Session Key used to sign all the packets using Hash based Message Authentication Code (HMAC). Encryption is optionally available for devices transmitting sensitive information. In the packet used to advertise the device, a Device Type String is provided from which the Hub is able to calculate a unique URL from where it can download an XML file called Device Specification File (Spec File).

Each device is described by an XML document called a Device Specification File, describing the capabilities of the device, configuration options, list of software objects exported by the device, defined requirements and limits, proposed Access Control List (ACL) entries, and links to a list of Interface Definition Files. Interface Definition Files are also XML files describing in a machine understandable way all the messages exchanged by the device and how those messages can be combined to implement Remote Method calls. Interfaces can inherit other interfaces, allowing a new device to extend the functionality of an old one therefore enabling backward compatibility. Interface files are the entry point for an interface compiler that generates code stubs for both the device itself and for the software communicating with the device. It also provides security information to help build an ACL Database. XML files are downloaded by the Hub and by Developer tools and there is no requirement for the devices to be able to parse or even provide them in order to accommodate small memory foot-print devices. However, envisioned enhancements to the protocol will add the option for devices to provide themselves the Spec Files (if hardware permits it) in order to allow them to operate in Internet disconnected environments.

The House Hub enforces strict access rights, by checking via Access Control Lists (ACL) each Direct Message sent and each Notification Subscription. Hub's ACL database is built automatically by the system from the Device Specification Files provided by the manufacturers and by analyzing Standardized Expert Advice Files provided by trusted third party experts.

The Session Key is generated by the Hub and uploaded to the device via a key exchange algorithm described in [1]. The key exchange algorithm is using the device unique Device ID and Device Unique Key.

In an sDOMO model, a device is not supposed to access the Internet directly, the model proposes allocating of a certain range of IP addresses for home automation devices and will have the firewall blocking those IPs from accessing the outside world. When a device needs to get updates or data from the Internet, it has to request the required information from an Internet Gateway which analyses the request, and performs a controlled access to the network to request the required resource. The type and purpose of the information required from the outside-world is to be described in the XML Spec file for the device and it may be subject to override by trusted expert advice.

House Intelligence Unit (HIU) is a piece of software running on the Base Station computer and having access privilege to the House Hub. The main role of HIU is to provide a higher level of automation by choreographing coordinated actions across multiple devices and external information.

1.2 *Highlights*

The compact binary packed protocol implemented by sDOMO and by the fact that the protocol outsources main routing and security decisions to the House Hub solves the problem of integration of cheap, small sensors in the domotic network, without the need to use protocol adapters. We demonstrated the scalability of sDOMO toward low end devices by implementing a minimalist but fully functional thermostat running on an Arduino Uno (8 Bit AVR Microcontroller, 2KB RAM, 32 KB Program Flash) speaking directly sDOMO over UDP/IP via an “Ethernet Shield”.

Security is a big concern in any home automation network. A hacker sending fake temperature reports to an A/C unit can induce hypothermia in a diabetic patient unable to properly sense the temperature. If a thief can get control of the door-lock and the alarm system, the houses will become free-for-all self-serve depots. The advent of domestic robotics brings a whole new dimension to the security and privacy problem since a robot is capable of directly inflicting bodily harm or even killing the unsuspecting residents.

The “Internet of Things Research Study, 2014 Report” [14] conducted by Hewlett-Packard examined the security of some commercial devices directly connected via Internet to their manufacturers (Cloud based IoT model) and the findings were alarming. For example 70% of the devices used unencrypted network services, 80% used weak passwords schema, while 90% of the devices

and their services collected personal information about the users. To help alleviate these security and privacy problems, these issues have to be considered from the design phase of the communication protocol and home automation model. To gain trust, the communication protocols can be fully documented and the specifications available for public scrutiny by other experts. We are doing this with sDOMO and in our work we tried to incorporate from the protocol design layers of protective measures for the security and privacy of the users.

The first line of defense is the fact that an sDOMO home automation system is designed to operate as a self-contained network, independent of Internet connectivity or protected behind a firewall. The data traffic between devices is taking place on the local house-network reducing the opportunity for snooping or direct attack.

To prevent attacks, in cases when an intruder got a foothold inside the local network, sDOMO requires each packet to be signed and verified at its destination, it also implements a schema for enforcing the uniqueness of each packet preventing even reply attacks. This works by attaching a unique id to each packet and forcing a generation of a new Session Key when the packet counter approaches the upper limit.

We also implemented a unique feature that discourages a Trojan horse from even attempting to hijack a device by the threat of automatic disclosure of the attackers and by allowing HIU to run scripts that retaliates against the detected intruder as presented in details in [1].

Privacy is a huge concern with a home automation network, especially since we live in a time when an increasingly large number of companies purposely violate the expectation of privacy of people and collect whatever data they can gather to make money by selling it to marketers. Inside our homes most people have a total expectation of privacy, but with the model of "Cloud Based Internet of Things" this is nothing but a false hope as presented above.

By proposing a self-sufficient domotic network that can operate locally independent of "The Cloud" and having all the Internet requests filtered by Internet Gateway, we are taking the first step toward protecting the privacy of the users by avoiding the reliance on services of various companies that prey on user's data. Analyzing plug-ins running on the Gateway can help monitor and raise alarm if a device is trying to upload data to the cloud by disguising it as a request for updates, or other "creative ideas".

The enforcement by ACL of which a device can access data will also reduce the risk that companies will design Trojan devices that collect information from other sensors into the house to pack and upload it to the manufacturer site.

Another layer of enforcement is provided by an alternative use of HIU to increase the privacy by coordinating actions between devices that we do not trust to have access to each other. For example, a thermostat may need access to motion information from the cameras to determine if the house is occupied or not. But if the parent company of the thermostat maker is in the business of selling data to marketers, we may be concerned to let it access the video stream from cameras. However, HIU can subscribe to cameras and spoof toward the thermostat motion information without any identifiable feature. That way, commercial off the shelf devices from vendors that we don't trust can be safely integrated in our homes.

Finally, since device Specification Files must be public documents, they can be audited by third party experts, who can, not only raise red-flags but can override ACL proposed by manufacturer to solve discovered security and privacy issues. The core idea is that each Hub will have a list of trusted experts and their public keys and periodically scan their sites for XML files that provide machine understandable expert advice files. That way, third party experts can prevent dishonest manufacturers from spying on their customers.

2. sDOMO and IoT

The concept of an Internet of Things is a very broad concept covering various aspects from embedded RFID tags into clothing, to fully integrated city wide utility information systems [9]. Therefore, the scope of communication protocols used in IoT is equally wide. In [13] we see a few critical considerations about the design of smart homes.

A survey [12] conducted at the university of Essex highlighted the major concerns of people in regard with intelligent homes, among which some are important to note: the feeling of being in control, privacy and cognitive workload. As we are headed toward pervasive intelligent environments, the human factor must also be considered and we believe that more studies of this kind are necessary, since the design engineers need to understand the response of the users living inside smart environments.

In the following literature review, we will look only at the protocol with a somehow similar scope with sDOMO highlighting the differences and similarities. Extensible Messaging and Presence Protocol (XMPP) [3] developed originally for Instant Message - “Jabber” provides near-real-time data exchange via XML based messages. The protocol is implemented as an open standard, using a client-server architecture. System speaking, the protocol can be isolated from the internet by implementing behind a firewall a private XMPP server. However, the verbosity of XML and the lack of native binary data transfer

(which must be encoded base64 inside XML) makes it difficult to be used for small microcontroller based devices. Nest Service Data Model is tied to direct Internet access. All Nest devices connect directly to Nest Service which is hosted online by company servers and are used to access the devices via web-site or applications. The system relies on JSON data packing using REST interface over HTTPS excluding small microcontroller devices from being able to speak to the Nest API directly. Hosting the data services on company websites also raises questions of privacy and security of personal information.

Because of the scope of the protocol, sDOMO is designed to be able to operate on local networks, isolated from the Internet by a firewall. The boundaries of the system are defined by a local network (domestic or robot based) and not the provider of the devices. SDOMO is more compact and able to work on devices that are out of reach for XMPP or Nest JSON.

The Constrained Application Protocol (CoAP) [4] is a very interesting recent development in IoT. It appears to be even more compact than sDOMO fitting inside 8-bit microcontrollers. The minimally binary packed header allows very small CPU and bandwidth overhead but it lacks security and privacy features. The message size in CoAP is also limited to the datagram size (about 1.1KB over UDP) while sDOMO is using the Message Carrier Packs to allow messages with a theoretical limit of 4GB. CoAP has been designed to have their message easily translated to HTTP by an adapter but does not in itself define a network model beyond that of device to device communication. sDOMO in contrast defines the network model and services like HIU and Hub's virtual Devices.

In [7] there has been presented a framework for connecting devices to “the cloud” via REST Web Services. In the proposed architecture sensors speaking their own native protocol are connected to the Internet via an adapter connected to an open source “Internet of Things” website that allows publishing sensor data using HTTP. The security implications of this approach are immediately observed. Publishing data on the Internet sometimes even using unsecured HTTP protocols is incompatible with requirements for a home automation system where security and privacy must be taken very seriously. By contrast sDOMO is addressing these issues from protocol design.

Originally, accepted with enthusiasm without too much attention to security and privacy, in recent years IoT is finally receiving the needed scrutiny. In [8] the authors analyze the current state of cloud-based IoT and discuss a number of consideration about it. And in [10, 11] we can find another quick overview of the issues.

3. Usage Example

The protocol has been originally developed to support our Distributed Processing Architecture for Domestic Robots [2]. The modules of this system are currently fully functional communicating using sDOMO to provide soft-real-time data exchange, making the system a good illustration of how sDOMO is intended to operate.

From the software point of view the system is composed of Camera Modules (CM) collecting images from fixed cameras and processing them with the object tracking algorithm MPTracker [6]. As a result to tracking objects CMs emit a notification to which subscribes Situation Awareness Module (SAM) and an Engineering Console.

SAM and the Console can request via Direct Messages images or extended information from the tracker. The request is answered with another Direct Message by CM, the request/response being in fact an sDOMO remote method call.

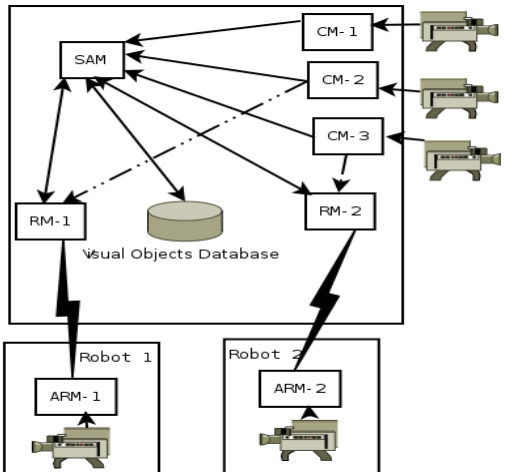

Fig 2: Software Module in our Robotic System

SAM is using homography to build from images a 3D model of the room and then translates the tracking information from CMs into the model coordinates. The translated tracking coordinates are also broadcasted as an sDOMO Notification.

Robot Module (RM) running on the Base Station subscribes to these Notifications and uses them to track its own movement. RM can also request as needed images from CM and ARM to perform epipolar geometry matching. RM sends high level movement commands to ARM. Autonomous Robot Modules (ARM) runs on the embedded board on the robot itself. It receives

Direct Messages from RM with requested movement vectors and responds to requests. An optical flow Notification is sent by ARM from locally processed images acquired by the robot camera.

4. Conclusions and Future Work

sDOMO is a communication protocol optimized for Home Automation and Robotic Systems, small enough to accommodate microcontroller based devices as peers in the network but powerful enough to handle a house wide computer vision system for controlling domestic robots.

With a CPU/Bandwidth overhead a little over that of CoAP, sDOMO is able to provide security and privacy features some of them unmatched [1] by more heavy-weight protocols, like the security features that discourages a Trojan Horse from even attempting to hijack a device or the ability of trusted third party experts to publish XML documents that fixes on the flight privacy features discovered with devices already deployed.

The protocol is in active development under a liberal open source license. We hope to be able to provide a practically-functional release by the end of 2016 or so. A companion protocol to sDOMO is being envisioned as an Inter-Hub protocol. This will allow multiple Hubs to coexist on the same domain increasing reliability and throughput. An extension to the protocol to allow devices to provide themselves the Spec File upon request, combined with network topology detection, will enable sDOMO devices to be used as building blocks for robotic systems, where a “hand” or a “leg” can be plugged into a body, detect it position and be configured on the fly.

The Public Key Infrastructure and Trust Management rules required to support the Expert Advice Files is still pending to be defined.

References

1. Marcel-Titus Marginean and Chao Lu, “sDOMO – A Simple Communication Protocol for Home Automation and Robotic Systems”, IEEE International Conference on Technologies for Practical Robot Applications; May 11 - 12 2015.
2. Marcel-Titus Marginean and Chao Lu, “A Distributed Processing Architecture for Vision Based Domestic Robot Navigation”, International Conference on Computers, Communications and Systems; Nov 1 2013.
3. Internet Engineering Task Force, RFC-6120, RFC-6121. P. Saint-Andre, March 2011. ISSN: 2070-1721.
4. Internet Engineering Task Force, RFC-7252, Z. Shelby et al, June 2014. ISSN: 2070-1721.

5. Castro M, et al. “Enabling end-to-end CoAP-based communications for the Web of Things”. Journal of Network and Computer Applications (2014), http://dx.doi.org/10.1016/j.jnca.2014.09.019i.
6. Marcel-Titus Marginean and Chao Lu, “A Multi-Paradigm Object Tracker for Robot Navigation Assisted by External Computer Vision”. ACM RACS 2014.
7. Rajeev Piyare and Seong Ro Lee, "Towards Internet of Things (IoTs): Integration of Wireless Sensors Network to Cloud Services for Data Collection and Sharing".International Journal of Computer Networks & Communications (IJCNC) Vol.5, No.5, September 2013.
8. Singh, J.; Pasquier, T.; Bacon, J.; Ko, H.; Eyers, D., "Twenty Cloud Security Considerations for Supporting the Internet of Things," in Internet of Things Journal, IEEE , vol.PP, no.99, pp.1-1.
9. Ángel Asensio et al., "Managing Emergency Situations in the Smart City: The Smart Signal".Sensors 2015, 15, 14370-14396; doi: 10.3390/s150614370.
10. M.U. Farooq et al. "A Critical Analysis on the Security Concerns of Internet of Things (IoT)". International Journal of Computer Applications (0975 8887) Volume 111 - No. 7, February 2015.
11. Rodrigo Roman, Pablo Najera, and Javier Lopez, "Securing the Internet of Things". IEEE Computer, vol. 44, no. 9, pp. 51-58, September 2011.
12. Matthew Ball and Vic Callaghan, "Who is in Control of Intelligent Environments? A Question of Autonomy". 10th International Conference on Pervasive Computing, June 18, 2012.
13. Sam Solaimani et al. “Critical design issues for the development of Smart Home technologies”. J. Design Research, Vol. 11, No. 1, 2013.
14. Hewlett-Packard, “Internet of Things Research Study, 2014 Report”. Online: http://h20195.www2.hp.com/V2/GetDocument.aspx?docname=4AA5-4759ENW.