Demo: Uncovering Device Whispers in Smart Homes

Simon Mayer, Christian Beckel, Bram Scheidegger, Claude Barthels, Gábor Sörös
Institute for Pervasive Computing
ETH Zurich
8092 Zurich, Switzerland
{simon.mayer, beckel, soeroesg}@inf.ethz.ch
{brams, claudeb}@student.ethz.ch

ABSTRACT

As the Internet of Things finds its way into private households, more and more everyday objects communicate with services that are running inside the home and on the Internet. For individuals to trust their smart homes, they should be aware of possibly privacy-sensitive data flows and control commands. In this demo paper, we present a system that combines a real time network analysis tool with an augmented reality user interface to visualize data streams within the home network and to remote services. Our system requires no modifications to a typical home network infrastructure, as it operates by merely observing packets sent over the network.

1. INTRODUCTION

More and more sensors, actuators, and everyday objects like household appliances and entertainment devices are being connected to the Internet [6] and provide services such as remote monitoring and control and social media integration. Increasingly being enriched with such "smart things", households slowly turn into "smart homes" enabling a plethora of applications across domains ranging from energy efficiency to ambient assisted living, entertainment, and security. To use energy more efficiently, for example, cold appliances can be remotely controlled by utility companies to adapt their cooling cycles and avoid peak loads in the energy grid [1]. Alternatively, information about the occupancy state of a smart home can be used to automatically infer heating schedules that reduce energy waste while still providing a comfortable indoor temperature level [5].

Such highly connected environments exhibit two major challenges, which, if left unaddressed, might hinder the overall adoption of smart homes. First, in such environments, devices and applications form complex networks where devices communicate invisibly behind the back of inhabitants while potentially making fully autonomous decisions. Problems persist regarding the configuration and maintenance of the network itself: even with the rather simple home network infrastructures that are common today, users have problems facing the "invisibility of settings and configuration information" of the network as well as "poor strategies for diagnosis and troubleshooting" [7]. One of the main reasons for these difficulties is that traffic flows between machines are invisible and collecting information about the network requires tools that are hard to operate and almost never used by ordinary end users [4]. Network management tools made specifically for end users (e.g., Cisco Systems *Network Magic*) offer a simpler interface but focus on displaying static network information such as high-level security features (e.g., printer access, parental monitoring) while neglecting dynamic information such as network activity. To monitor and debug *dynamic* properties of their networks, most home network administrators therefore still rely on status displays in the form of indicator lights on routers and cable modems [7].

The second challenge in connecting everyday objects to the Internet lies in the value of the sensor data itself: smart home sensors provide information about the inhabitants' lives which represents significant value for companies. Already, privacy issues in an increasingly connected world have become a topic of common interest [3]. A user study with 55 participants at different sites in five European countries presented by Röcker et al. concludes that the "fear of pirating" constitutes the most severe reservation of people regarding the usage of remote services in the context of their smart homes [8].

We propose a system that allows end users to monitor information flows in smart homes in an easy, intuitive, and non-invasive fashion which enables inhabitants to better understand interactions of devices inside their smart homes. To achieve this, our system uses a mobile device (e.g., a smartphone or tablet computer) to visualize network traffic between devices and connections to remote services in real time. Making use of techniques from the domains of network sniffing and augmented reality, the uncovered data flows are overlaid on the mobile device's live camera view. Since our implementation of the network sniffing application runs on a widely available commodity router, it can be deployed in private households without the need for changes in the network infrastructure.

Figure 1 illustrates an application scenario of our system. It shows a picture of the kitchen in our office building that contains several everyday devices such as a fridge, coffee machine, notebook, plant, and a loudspeaker. These objects are connected to each other and to the Internet through a wireless router. On the second image, the traffic flows between the devices and remote services are displayed as an

Copyright is held by the author/owner(s).
MUM '12, Dec 04-06 2012, Ulm, Germany
ACM 978-1-4503-1815-0/12/12.
.

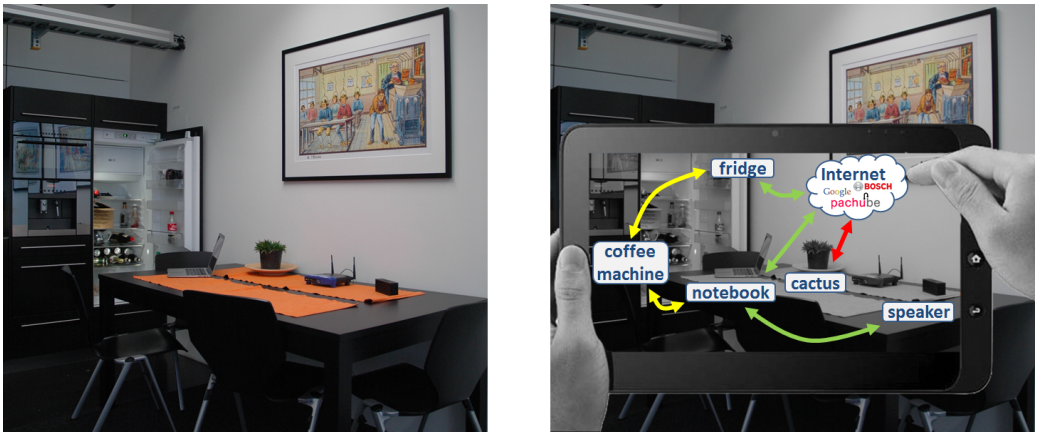


Figure 1: Uncovering device whispers: application scenario (mock-up). The user can discover traffic flows between devices by using a tablet PC as a "magic lens". Green lines represent safe, yellow lines represent unusual, and red lines represent critical connections.

overlay on a tablet computer.

2. SYSTEM OVERVIEW

The back end of our system is a sniffer application that collects network traffic data and shares this information via a Web interface. The information is then visualized on a mobile front end textually, graphically, and as an augmented reality overlay. As hardware platform for the back end, we use a Linksys WRT54GL, a widely used router[1] that runs the Linux-based OpenWrt[2] operating system.

2.1 Sniffer Back End

The sniffer application consists of several modules which take care of intercepting packets (the actual sniffing) and of handling the storage and brokerage of traffic information and device metadata for the mobile front end. To sniff traffic, our system uses the Linux *netfilter* firewall: an *iptables* rule is added to the firewall's `FORWARD` chain which affects all packets that are not intended for local delivery. This way, information about all packets that pass through this chain is logged by the syslog daemon and can be retrieved from shared memory. From the collected data, the sniffer application distills information about data flows between endpoints and exposes this information via a Web interface.

2.2 Mobile Front End

The collected network traffic data is presented to the user on his/her mobile device, for instance on a smartphone or tablet computer. The user selects a networked smart thing by either scanning its QR-code or by pointing the mobile device's camera towards the smart thing's AR marker – both can be resolved to find the device's URL. Next, the mobile application requests further information about the networked device by contacting the sniffer application, which delivers connection information such as the IP addresses of contacted devices, protocols used, the strength of connections, and the direction of traffic (in- vs. outbound). Furthermore, the sniffer's response contains metadata like node names, descriptions, pictures, etc. To enable it to provide this information, the user manually registers newly encountered tagged devices by scanning the device's QR-tag, and providing its name, a picture, and a description.

Our application prototype (Android operating system) features multiple ways of visualizing the data streams inside a network: it can summarize all incoming and outgoing connections of a selected node textually (*text view*) or draw an abstract graphical representation of the connections (*graphical view*). Additionally, the application features an augmented reality interface (*AR view*) where information about the data flows is overlaid on the camera image, thereby turning the mobile device into a "magic lens" (cf. [2]). In the *AR view*, networked devices are tracked in the live camera view using their associated markers, and connection information of these devices is rendered within the camera image as an augmented reality overlay (Figure 2). If a marker is recognized in the camera image, the picture of the associated device gets superimposed over the marker. If markers of multiple networked nodes are recognized and the respective nodes have been communicating with each other, the devices' pictures are connected by a line in the camera image. If a node has connections to other nodes whose markers are not visible in the camera image, their pictures are displayed as hovering above the node to make the user aware of the connections. Multiple connected devices are arranged in a circle around the central node. If no pictures are available, the application uses a single placeholder image that represents multiple devices to avoid overloading the viewport with too many images and rather emphasize those nodes that have been registered by the user. The application has been implemented on top of ARToolKit for Mobile from ARToolWorks[3] and can distinguish between 64 unique markers

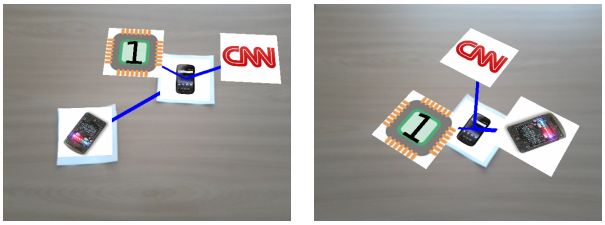
3. DEMONSTRATION SETUP

For demonstration purposes, we use a setup with a Web-enabled wireless Sun SPOT sensor node and a mobile phone, which both carry a visual marker. The mobile front-end of

[1] Currently holding `amazon.com` "Best Sellers" rank #7 in the category *Routers*

[2] `openwrt.org`

[3] `artoolworks.com/products/mobile/artoolkit-for-mobile/`



(a) (b)

Figure 2: (a) Augmented Reality overlay of connections between the Sun SPOT, mobile phone, and the CNN website. (b) The same situation without the Sun SPOT's marker being visible to the tablet's camera.

our system is running on a tablet computer. All devices are connected to an OpenWRT router with Internet access that runs the sniffer application. During the demonstration, the mobile phone is used to access the Sun SPOT's Web interface and to browse the `www.cnn.com` website. The sniffer application stores information about these connections and exposes this information via its Web interface. The application on the tablet device polls the sniffer's Web interface and updates its *text* and *graphical* views accordingly. In the *Augmented Reality view*, the tablet application identifies and tracks the markers and overlays information about the network traffic flows on the camera image.

The tablet's screen is shown in Figure 2(a) with the markers representing the mobile phone and the Sun SPOT present in the camera image. As a consequence of the previous interactions between the devices, the mobile phone is shown to have been connected to both the Sun SPOT and the `www.cnn.com` website (represented by the *CNN* logo). The line connecting the mobile phone and the Sun SPOT is thicker because the Sun SPOT was accessed more often. In Figure 2(b), the Sun SPOT's marker has been removed. Thus, this device's image is now displayed as hovering over the scene and is shown as being connected to the mobile phone. In both situations, a placeholder image is shown as being connected to the mobile phone, which indicates that the phone also established a connection to one additional endpoint which the application cannot provide a visual representation of. In this case, this unknown endpoint is the name server used to resolve the `www.cnn.com` domain.

In our demo setup, the proposed system works well and the mobile application (specifically its *AR View*) is responsive enough to display connection information between devices and remote services in real time. Using ARToolKit to track the markers is stable, where we have tested the system with up to 4 simultaneously tracked tags without performance problems.

4. CONCLUSIONS

The system described in this demo paper visualizes network traffic between smart things on a mobile device using an augmented reality overlay. This enables inhabitants of smart homes to keep track of their smart devices' communication behavior – such as the interaction of smart devices with each other and their connections to remote services on the Internet. As our system does not require any specialized network infrastructure, it can be deployed on a commodity router that is present in many private homes already today. In the future, we plan to integrate contextual information about devices (e.g., their location, or type) to further augment the displayed information. Furthermore, we will integrate techniques of software-defined networking, such that users of the application in home networks could then not only monitor the communication of their devices but also use the mobile interface to define policies that govern which devices are allowed to communicate with which endpoints.

5. ACKNOWLEDGEMENTS

This work has been partially supported by the Hans L. Merkle Foundation and by the Swiss National Science Foundation under grant number 134631.

6. REFERENCES

[1] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht. SmartCap: Flattening Peak Electricity Demand in Smart Homes. In *Proceedings of the 10th IEEE International Conference on Pervasive Computing and Communications (PerCom 2012)*, Lugano, Switzerland, 2012.

[2] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, pages 73–80, Anaheim, CA, 1993. ACM, New York, NY, USA.

[3] R. Gellman. Privacy in the Clouds: Risks to Privacy and Confidentiality from Cloud Computing. Technical report, Feb. 2009.

[4] R. E. Grinter, W. K. Edwards, M. W. Newman, and N. Ducheneaut. The Work to Make a Home Network Work. In *Proceedings of the 9th European Conference on Computer Supported Cooperative Work (ECSCW 2005)*, pages 469–488, Paris, France, 2005.

[5] W. Kleiminger, C. Beckel, and S. Santini. Opportunistic Sensing for Efficient Energy Usage in Private Households. In *Proceedings of the Smart Energy Strategies Conference 2011*, Zurich, Switzerland, 2011.

[6] F. Mattern and C. Floerkemeier. From the Internet of Computers to the Internet of Things. In K. Sachs, I. Petrov, and P. Guerrero, editors, *From Active Data Management to Event-Based Systems and More*, volume 6462 of *LNCS*, pages 242–259. Springer, Berlin, Germany, 2010.

[7] E. S. Poole, M. Chetty, R. E. Grinter, and W. K. Edwards. More Than Meets the Eye: Transforming the User Experience of Home Network Management. In *Proceedings of the 7th ACM Conference on Designing Interactive Systems*, DIS '08, pages 455–464, Cape Town, South Africa, 2008. ACM, New York, NY, USA.

[8] C. Röcker, M. D. Janse, N. Portolan, and N. Streitz. User Requirements for Intelligent Home Environments: A Scenario-Driven Approach and Empirical Cross-Cultural Study. In *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence (sOc-EUSAI 2005)*, pages 111–116, Grenoble, France, 2005. ACM, New York, NY, USA.