cdbb

Centre for Digital Built Britain

IOT Network Behaviours and Dependencies

Poonam Yadav, Qi Li and Richard Mortier

November 2018
Series No: CDBB_WP_005
DOI: https://doi.org/10.17863/CAM.31463

UNIVERSITY OF CAMBRIDGE

cdbb

Centre for Digital Built Britain

This is a working paper, published in the CDBB publication series.

Acknowledgements:

This research was funded by the Centre for Digital Built Britain (CDBB). www.cdbb.cam.ac.uk

IOT Network Behaviours and Dependencies

Poonam Yadav, Qi Li and Richard Mortier

Department of Computer Science & Technology, University of Cambridge

To realise Digital Built Britain will require creation and use of many sources of digital data, particularly to achieve BIM levels 3 and 4. Many current claims suggest that many of these sources will arise from deployment of off-the-shelf Internet-of-Things (IoT) sensors in buildings and infrastructure at scale. In this paper we take an early look at some baseline data gathered from a set of such sensors. We perform some simple analysis of the types and rates of data these sensors will generate and distribute, the supporting services they will invoke, and the resulting infrastructure dependencies they take. We consider a wide range of sensors from environ- mental sensors (e.g., temperature, CO2) to some more immediately sensitive sensors (e.g., video cameras).

Introduction

Organisations including Gartner, BI Intelligence and IHS predict that the number IoT devices in 2020 will exceed 20 billion [9, 16, 18]. If this growth is even only partly obtained, it will pose many challenges ranging from ensuring security of data generated and actuation enabled by the IoT, obtaining privacy of citizens monitored by and using IoT devices, and managing the energy demands, net- work traffic, and service dependencies produced by use of IoT devices. To realise BIM levels 3 and 4 will require that we tackle these challenges as we deploy this new, often experimental – or at least technologically optimistic! – infrastructure in the form of smart buildings and cities.

The challenges posed by the current state of the IoT ecosystem are widespread, ranging across human, technological and environ- mental threats. In this context a *threat* could be defined as possible danger that might exploit vulnerabilities in a system to cause potential harm [4]. To minimise possible threats, the system needs to provide for various security requirements: authentication, confidentiality, integrity, non-repudiation and availability [1, 22]. Additionally, privacy risks must be managed, involving data subjects' right to control, edit, manage and delete information about them- selves, as well as deciding when, how and the extent to which in- formation about them may be communicated to others [27].

Recent years have seen both privacy and security perspective explored, analysed and presented in many research articles [4]. How- ever, research on how the critical end-to-end services and infrastructure components of the IoT ecosystem could affect scalability, availability and integrity of these systems is still patchy and lacking in detailed analysis. In this paper, we focus our focus on understanding the threats linked to the scalability, availability, and integrity that IoT sensors and actuators are going to create. We do so by collecting network packet traces for a range of devices over approximately one month, and analysing these data to take a first look at the protocol behaviours of them.

Methodology

We use the term “smart environment” to refer to a smart home, office or city. In each we will likely see different, but overlapping, classes of IoT device.

In a smart home, typical devices might include environmental sensors, security cameras, personal health and wearable devices, voice-controlled assistants, and robots. In smart hospitals, various IoT enabled wearable health monitoring and medical devices (e.g., drug monitoring and delivery systems, pacemakers), and even smart medical robots are integrated to provide a end-to-end efficient work- flow and service with in the hospitals [30]. Smart offices may use smart home devices as well as adding systems more pertinent to the shared work space such as resource scheduling solutions that allow employee badges to register their presence in the office. Smart City environment involves deployment of smart sensors on city wide scale; for example, smart traffic light systems, smart parking solutions. These solutions involve both local and central processing of information.

To begin to understand the data types, rates, and traffic patterns caused by different IoT devices, we deployed a set of off-the-self IoT devices in a small test area in an office in our lab, and captured IP and local wireless traffic generated by these devices. Other occupants of the office were notified that the devices were present, and we carefully did not analyse data captured from the devices for anything other than its gross network characteristics. The data captured thus represents a “minimum” level of traffic, as the de- vices were largely idle and not interacted with as they might be in a more realistic deployment.

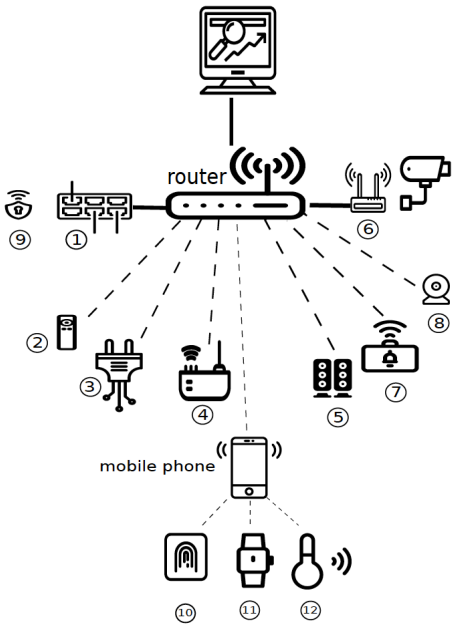


Figure 1: Table 1 describes the devices shown.

The Hive Hub (1) and Arlo Security Camera Hub (6) are connected via wired Ethernet; the Hive Motion Sensor (9) communicates with the Hive Hub using Zigbee, and the Security Camera connects over WiFi to the Arlo Security Camera Hub. Other devices (3, 4, 5, 7, 8 and the controlling smart phone) connect over Wi-Fi except for devices 10, 11, 12 which connect to the smart phone over Bluetooth. The router reaches the Internet via wired Ethernet and the University’s network, and runs *tcpdump* to collect packets. Packet capture (PCAP) files are periodically copied from the router to a directly connected Linux host.

Experimental Setup

Within this test area, we deployed a number of off-the-self commercial IoT devices described in Table 1). All were connected to a local Netgear N600 Wireless Dual Band Router WNDR3700v2 running Linux OpenWrt version 2.6.39.4 [6] either wirelessly over standard 802.11b Wi-Fi or via an Ethernet cable. We capture all traffic to or from these devices which are connected to this router via WiFi or over an Ethernet cable. The experimental setup is

shown in Figure 1.

We categorise all IoT devices in two categories: (1) **Hub** is an IoT device which discovers and control other IoT devices. (2) **Sensor** is an IoT device which connects to the router directly and then communicates with IoT cloud services without using any Hub. We categorise our IoT devices in these categories as shown in Table 1 and analyse if there is any different traffic pattern by these two categories.

Data Collection & Analysis

For data collection from all IoT devices in our network, we run monitor and collect scripts on the router. First, we get a list of MAC addresses of the devices. On the router, we run monitor script on the interface that provides wifi access to the devices, filtering the traffic based on the MAC addresses we're interested in using tcpdump [25]. At this stage, the traffic from different devices, different protocols are not separated, they reside in the same pcap files. On the other side, we schedule a cron job to periodically upload collected data to the Linux machine (on the same university network) for offline processing. One reason for this is that the router only provides limited persistence storage space, which is less than 50M. We then perform offline analysis on all the pcap files, and analyse traffic based on the metadata. We used tcpdump [25], a network traffic analysis utility, to get the meta-data information of the traffic flow. The information captured by the tcpdump is in this format.

Analysis

We next perform some simple analyses of traffic collected according to the setup shown in Figure 1, transmitted and received by the devices listed in Table 1.

Protocol Breakdown

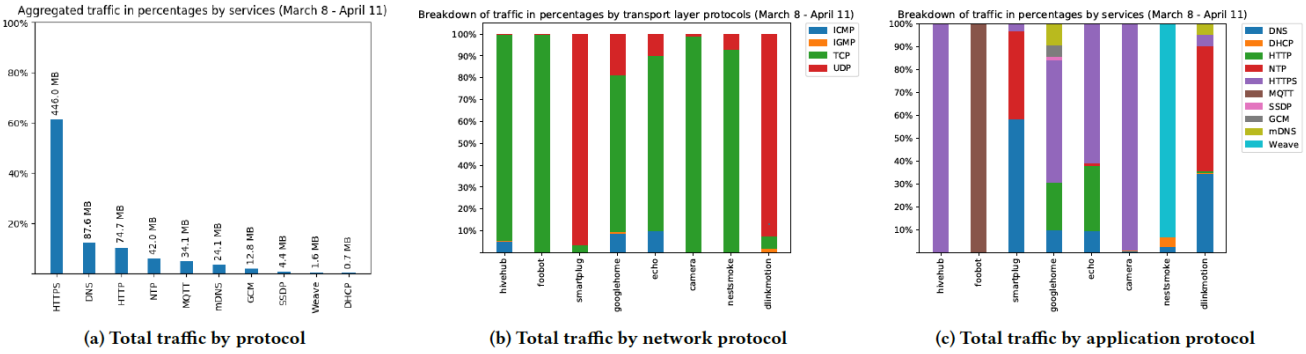
Figure 2 presents a breakdown of the entire dataset by application protocol (Figure 2a), and by network and application protocol per device (Figures 2b and 2c). It is surprising to observe how much NTP, DNS and mDNS is in use by two devices (the Smartplug and DLink Motion Sensor). It is also interesting to observe that only one device makes significant use of a classical IoT protocol (MQTT, used by the Foo bot), though the Nest device also uses an IoT specific protocol (Weave) that was proprietary until released into Nest's developer platform in 2015.

Local Network. For pairing and device discovery, many IoT hub uses low power and low range communication protocols to con- nect to the devices (sensors). These protocols include Zigbee(IEEE 802.15.4) [2], Lora [24], Zwave [29], Lightwave [17], Bluetooth [12], RFID communication(LF, (125 - 134 kHz), HF, (13.56 MHz), UHF, (433, and 860-960 MHz)) [23]. In our setup, we have few devices directly connecting to Hub using Zigbee, Bluetooth and WiFi. For example, Hive motion sensor connects to Hive hub using Zigbee protocol. We have three sensor devices which communicates to Mobile Phone Apps using Bluetooth and then Mobile apps, com- municates to app servers through the standard WiFi connection using the router.

Encrypted Traffic. One of the interesting observations we would like to make from the collected traffic traces is to find how secure the communication between IoT devices and the outside world [10].

	Device	Device Type	Communication	Protocols	Encrypted	Energy (watts)	IP traffic rate (bytes/sec)
1	Hive Starter Kit Hub [13]	Hub	Ethernet	TCP, IGMP, ICMP	Yes	1.8	120
2	Foobot Air Quality Monitor	Sensor	WiFi	TCP	Yes	1.79	18
3	TP-link Smart Plug [26]	Hub	WiFi	UDP, TCP	Both	2.05	100
4	Google Home Mini [11]	Hub	WiFi	UDP, TCP, IGMP, ICMP	Both	1.4	
5	Amazon Echo Dot [3]	Hub	WiFi	UDP, TCP, ICMP	Both	1.95	125
6	Arlo Security Camera Base tion [5]	Hub	Ethernet WiFi	UDP, TCP	Both	4.6	70
7	Nest Smoke Alarm [19]	Sensor	WiFi	UDP, TCP	Both	NA	0.02
8	D-Link Motion Sensor [7]	Sensor	WiFi	UDP, TCP, IGMP	Both	1.4	NA
9	Hive Motion Sensor [14]	Sensor	Zigbee	NA	yes	Battery	NA
10	ParrotPot Smart Flower Pot	Sensor	Bluetooth	NA	NA	Battery	NA
11	MiBand Smart Bracelet [28]	Sensor	Bluetooth	NA	NA	Battery	NA
12	Smart Bluetooth Tracker [15]	Sensor	Bluetooth	NA	NA	Battery	NA

Table 1: IoT devices and their traffic behaviour summary



(a) Total traffic by protocol (b) Total traffic by network protocol (c) Total traffic by application protocol

Figure 2: Traffic breakdown by protocol and device, 8th March to 11th April 2018

As this is our preliminary analysis, at this stage we investigate if an IoT device sends unencrypted or encrypted traffic. We categorised traffic generated from each device based on application layer protocols (see summary in Figure 2c) and found that all IoT devices at least send some part of their traffic as HTTPS. We found that **Hub** IoT devices send more encrypted traffic (> 50%) as com- pared to **Sensor** IoT devices. At this stage of analysis, we have not investigated how secure is the use of HTTPS by different devices.

Traffic Patterns

IoT traffic rate and pattern have a huge impact on the infrastructure planning and supported services. To understand, traffic pat- tern, we analysed time-series traffic of all IoT devices we have in our Experiment setup. We present an hour time-series data in Figure 3 and found that MQTT, HTTPS and NTP are continuous and more frequent traffic and DHCP is the less frequent.

To make a preliminary investigation of traffic and service dependencies, we analysed 24 hours of data from of each device as well as one - week long traffic by calculating bandwidth with 5, 10, 15 minutes traffic aggregation window. We found there was not significant pattern difference with varying window sizes, so present our analysis here using 15 minutes traffic aggregation window size on one week time-series traffic. We found following interesting observations.

We looked at the traffic traces of individual devices and present 24-hrs traces. We found > 99% of the Hive hub (Figure 4a) and Arlo Security Camera Hub (Figure 5c) total traffic composed of HTTPS packets and rest of the traffic include few periodic DHCP, NTP and DNS packets. Similarly, Foo bot majority traffic consists of MQTT running over TCP. Some of the devices like Smart Plug (see Figure 4e), Amazon Echo (see Figure 5a) and Dlink Home motion sensor (see Figure 5g) send frequent NTP traffic, and therefore, we can see it makes a significant percentage of total traffic send by these devices. As compared to all other devices, the traffic rate generated by Nest Smoke sensor is minimum when it is in its ideal listening mode; it sends total 6 packets a day (total around 180 bytes in a day, Figure 5e). Nest smoke sensor uses *Weave* protocol over *TCP* to communicates periodically twice a day to Nest Cloud Service.

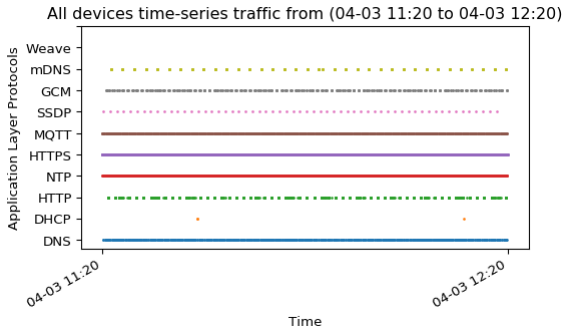


Figure 3: One-hour time-series of the traffic generated by IoT devices.

Every dot in the graph represent a packet to show the traffic packets generated by various application layer protocols / services.

Protocol & Service Dependency

When deploying IoT devices it is unavoidable that one takes dependencies on different standard Internet LAN and WAN protocols and services. For example, we found DHCP message frequency is relatively frequent and periodic, varying from (4 to 6 packets per day) for all IoT devices in our Experiment Setup. DHCP is only a local LAN protocol however. More significantly, we see dependencies on DNS and NTP. We found that voice assistant hub devices such as Google home mini and Amazon Echo makes frequent DNS enquires to their own cloud services and servers. The various services queried over DNS are summarised in Table 2).

We found NTP servers used by and frequency of NTP data generated by different devices varies significantly. There are few IoT devices like Hive hub, which access and exchange NTP during only setup process. The Foo bot air quality monitor and Nest smoke alarm uses their IoT protocols, MQTT and weave over TCP and uses embedded timing

protocols instead of standard NTP proto- cols. All IoT devices except TP-link communicate with their cloud services during initial setup phase as well for data APIs. This leads high risk TP-link smart plug communicates with global NTP Pool project servers [20].

We observed an interesting correlation between DNS queries and NTP traffic generated by TP-link smart plug and D-link Motion Sensor. Both these devices also make a large number of DNS queries to global NTP servers (DNS queries summary shown in Table 2). The total DNS traffic generated by just 4 devices makes nearly 12% of *total* traffic generated from our Network setup. On the other hand, Hive hub and Foo bot air quality monitor makes very few DNS queries to their servers.

In our setup, only two devices are using IoT protocols, rest of the devices uses standard Internet application layer protocols. We found that protocols customised for IoT devices are more efficient in term of traffic bandwidth and also uses less standard service protocols such as NTP. However, these are highly dependent on communication with the device cloud services, which leads to high unavailability risk due to single point of failure.

To realise smart buildings integrated with a large number and variety of IoT devices, we also made a very simple investigation of the energy consumption by these IoT devices and provide an estimate of additional energy requirements. We measured power consumed by IoT devices in our setup by connecting each device to a TP-Link smart plug for a fixed interval and provided mean- average values in the Table 1.

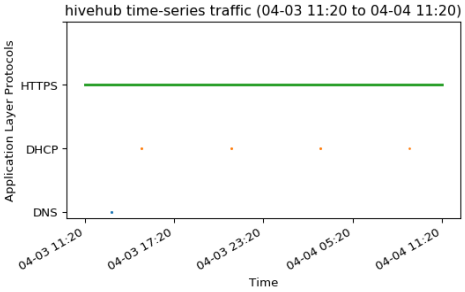
Finally, we used IP geolocation (using IP address allocation and routing data to infer where, geographically, a host with a particular IP address resides) to estimate the different countries hosting the services used by our devices. The results are in Table 3. Due to the strange and excessive use of NTP and DNS by the Echo and the Smartplug, both make use of the Internet-hosted services in dozens of countries. However, all the devices make use of Internet-hosted services in other countries, with most involving countries outside the EU. The implications of such devices and services be- come prevalent on the resilience and vulnerability to outside interference of our infrastructure are unclear and need further investigation.

Conclusions

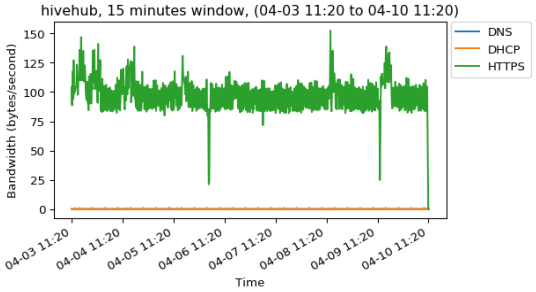
This early stage IoT device traffic and service analysis provided us some early insight into the network and application layer protocol, bandwidth, and time-series behaviour of a range of IoT devices. This allows us to begin to explore the service and infrastructure dependencies that will be taken in “smart” environments due to deployment of IoT devices. We are continuing to analyse the data in more detail to better understand the implications of large-scale deployment of these devices. In order to get the kind of coverage required of the vast range of devices available, we are also exploring how to setup systems able to receive and automatically process trace data submitted by third-parties concerning other devices. Ultimately, we hope to produce tools that can form the basis of a certification process for IoT devices suitable for providing data to BIM levels 3 and 4.

Acknowledgements

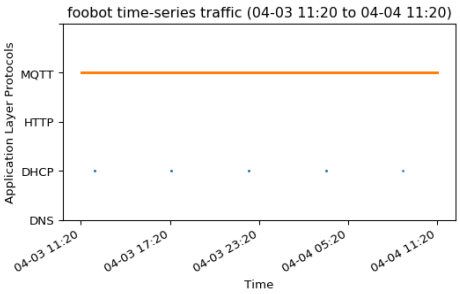
This project was supported by a mini-projects award from the Centre for Digital Built Britain and Innovate UK under Grant 90066, and by EPSRC EP/N028260/1.



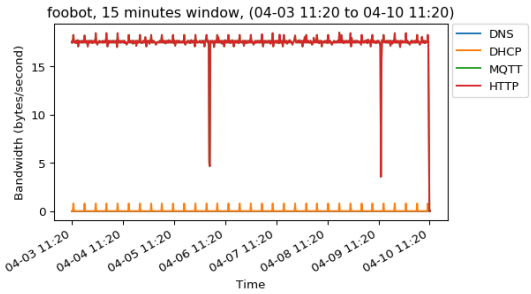
(a) Hive Hub pattern



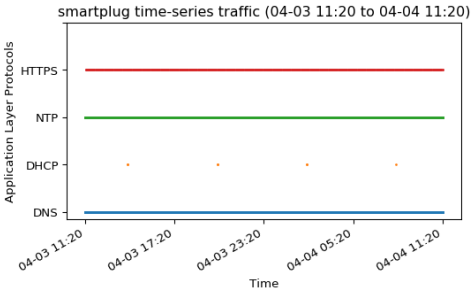
(b) Hive Hub bandwidth



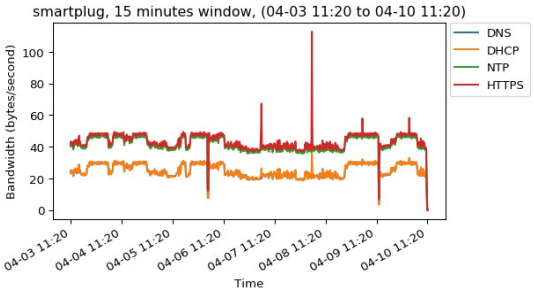
(c) Foo bot



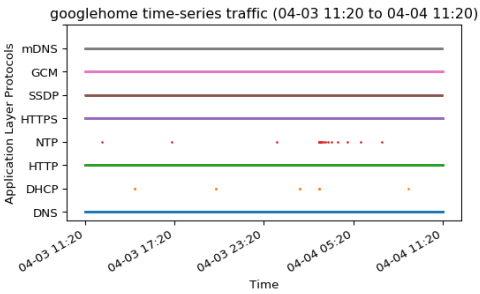
(d) Foo bot bandwidth



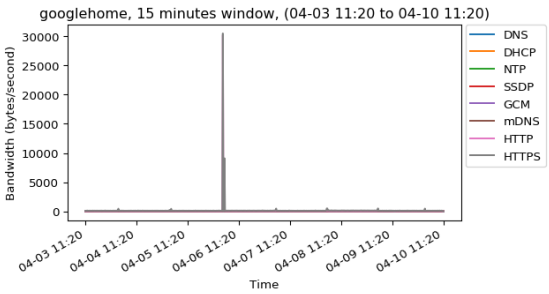
(e) Smartplug



(f) Smartplug bandwidth

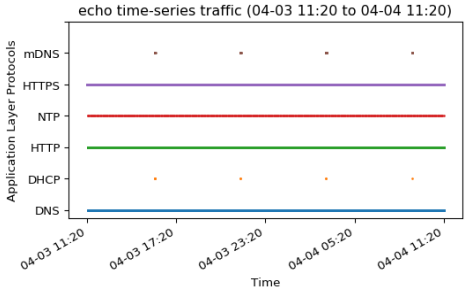


(g) Google Home

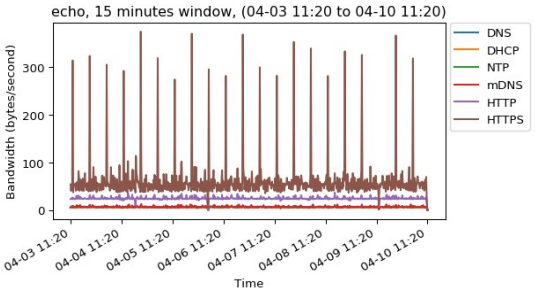


(h) Google Home bandwidth

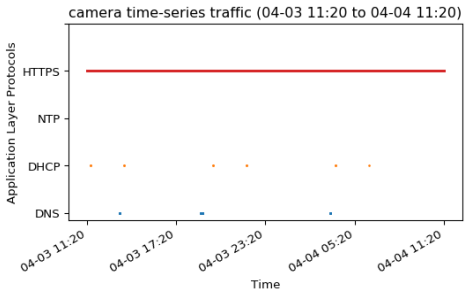
Figure 4: Time-series data representation of the traffic generated by each device, in terms of pattern in a 24h period and bandwidth (in 15 minute buckets) over a whole week.



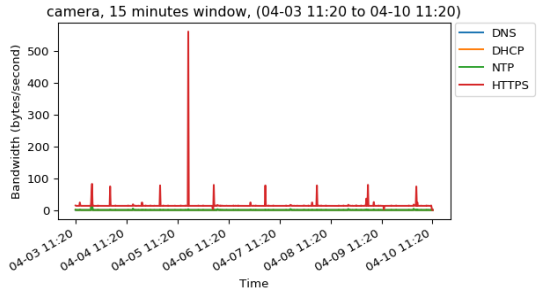
(a) Amazon Echo Dot



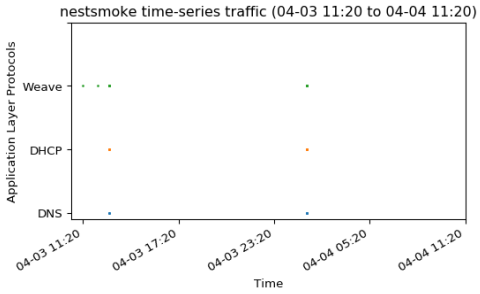
(b) Amazon Echo Dot bandwidth



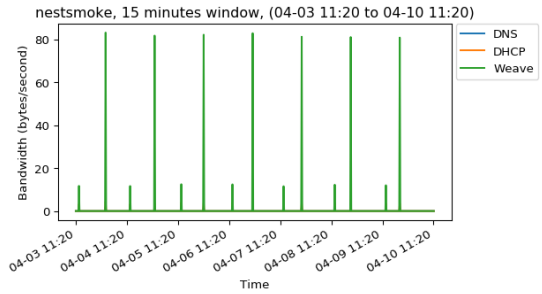
(c) Security Camera



(d) Security Camera bandwidth

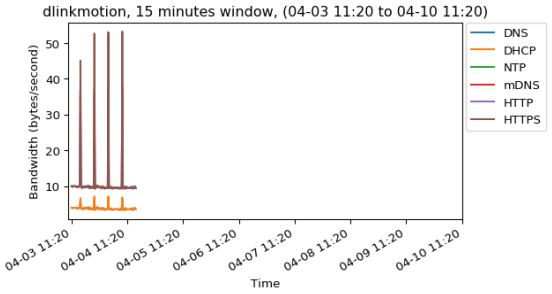


(e) Nest Smoke Alarm



(f) Nest Smoke Alarm bandwidth

(g) DLink Motion



(h) DLink Motion bandwidth.

Figure 5: Time-series data representation of the traffic generated by each device, in terms of pattern in a 24h period and bandwidth (in 15 minute buckets) over a whole week.

Refereneces

1. Hamoud M. Aldosari. 2015. A Proposed Security Layer for the Internet of Things Communication Reference Model. Procedia Computer Science 65 (2015), 95 –https:// doi.org/ 10.1016/ j.procs.2015.09.084 International Conference on Communications, management, and Information technology (ICCMIT'2015).
2. Zigbee Alliance. 2018. an IEEE 802.15.4-based specification for a suite of high- level communication protocols. (2018). http:// www.zigbee.org/
3. Amazon. 2018. Amazon Echo Dot. (2018). https://www.amazon.co.uk/ Amazon-Echo-Dot-Generation-Black/ dp/B01DFKBL68
4. O. Arias, J.Wurm, K. Hoang, and Y. Jin. 2015. Privacy and Security in Internet of Things and Wearable Devices. IEEE Transactions on Multi-Scale Computing Systems 1, 2 (April 2015), 99–109. https:// doi.org/ 10.1109/TMSCS.2015.2498605
5. Arlo. 2018. Arlo smart security camera. (2018). https://www.arlo.com/ uk/products/ arlo/ default.aspx
6. Arlo. 2018. Smart home base station. (2018). https://www.arlo.com/ uk/products/ arlo/ default.aspx
7. D-Link. 2018. Home WiFi Motion Sensor. (2018). https:// eu.dlink.com/ uk/en/ products/ dch-s150-motion-sensor
8. Foobot. 2018. Smart Indoor Air Quality Monitor. (2018). https:// foobot.io/features/
9. Gartner. 2017. Prediction of number of IoT devices. (2017). https://www.gartner.com/newsroom/id/3598917
10. M. Gebski, A. Penev, and R. K. Wong. 2006. Protocol Identification of Encrypted Network Traffic. In 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06). 957–960. https:// doi.org/ 10.1109/ WI.2006.139
11. Google. 2018. Google Home Mini. (2018). https:// store.google.com/ product/google_home_mini
12. Bluetooth Special Interest Group. 2018. Bluetooth: a wireless technology standard for exchanging data over short distances. (2018). https://www.bluetooth.com/
13. Hive. 2018. Hive Hub. (2018). https://www.hivehome.com/ products/hive-hub
14. Hive. 2018. Hive Motion Sensor. (2018). https://www.hivehome.com/products/ hive-motion-sensor
15. Imixcity. 2018. Mini Smart Bluetooth Tracker. (2018). https://www.amazon.co.uk/ Bluetooth-Tracker-Wireless-anti-lost-Reminder/dp/B01NCOAALX/ref=sr_1_1?s=computers&ie=UTF8&qid=1518174338&sr=1-1&keywords=smart+tag+tracker
16. BI Intelligence. 2018. Prediction of number of IoT devices. (2018). http:// uk.businessinsider.com/ the-internet-of-things-2017-report-2018-2-26-1
17. LightwaveRF. 2018. Lightwave RF. (2018). https:// lightwaverf.com/
18. IHS Markit. 2017. Prediction of number of IoT de-vices. (2017). https:// technology.ihs.com/ 596542/number-of-connected-iot-devices-will-surge-to-125-billion by-2030-ihs-markit-says
19. Nest. 2018. Nest Protect smoke and CO alarm. (2018). https:// nest.com/ uk/smoke-co-alarm/overview/
20. NTP. 2018. NTP Pool Project. (2018). http://www.pool.ntp.org/ en/
21. Parrot. 2018. Parrot POT. (2018). https://www.parrot.com/ uk/connected-garden/ parrot-pot#parrot-pot
22. J. Ren, H. Guo, C. Xu, and Y. Zhang. 2017. Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing. IEEE Network 31, 5 (2017), 96–105. https://doi.org/ 10.1109/MNET.2017.1700030
23. RFID. 2018. Radio Frequency Identification. (2018). https:// en.wikipedia.org/wiki/ Radio-frequency_identification
24. Semtech. 2018. Lora: a patented wireless data communication technology. (2018). https:// lora-alliance.org/
25. TCPDump. 2018. a command-line packet analyzer. (2018). https://www.tcpdump.org/

26. tp link. 2018. Wi-Fi Smart Plug with Energy Monitoring. (2018). https://www.tp-link.com/ uk/ products/ details/ cat-5258_HS110.html
27. Alan F. Westin. 1967. Privacy and Freedom. New York: Atheneum.
28. Xiaomi. 2018. Mi Band 2. (2018). http://www.mi.com/ en/miband2/
29. Zensys. 2018. Zwave: a wireless communications protocol used primarily for home automation. (2018). http://www.z-wave.com/
30. H. Zhang, J. Li, B. Wen, Y. Xun, and J. Liu. 2018. Connecting Intelligent Things in Smart Hospitals using NB-IoT. IEEE Internet of Things Journal (2018), 1–1. https:// doi.org/ 10.1109/JIOT.2018.2792423

Device	DNS Queries	Frequency
Hive Starter Kit hub	google.com	4
	honeycomb.fw.bgchprod.inf	19
	broker.prod.bgchprod.info	4
Foobot Air Quality Monitor	broker-gw-eu.foobot.io	8
	api.foobot.io	6
TP-link Smart Plug	de.pool.ntp.org	9335
	uk.pool.ntp.org	9334
	2.asia.pool.ntp.org	9330
	ca.pool.ntp.org	9329
	ru.pool.ntp.org	9329
	0.cn.pool.ntp.org	9329
	1.asia.pool.ntp.org	9329
	time-a.nist.gov	9329
	fr.pool.ntp.org	9328
	us.pool.ntp.org	9326
Google Home Mini	clients1.google.com	30984
	www.google.com	30977
	channel.status.request.url	11973
	clients4.google.com	7760
	clients3.google.com	1088
	android.googleapis.com	940
	connectivitycheck.gstatic.com	551
	time.google.com	551
	cast.google.com	301
	google.com	147
Amazon Echo Dot	kindle-time.amazon.com	9991
	spectrum.s3.amazonaws.com	4124
	ntp-g7g.amazon.com	2505
	device-metrics-us.amazon.com	1482
	2.android.pool.ntp.org	873
	pindorama-eu.amazon.com	104
	api.amazon.com	54
Security Camera Kit	time-h.netgear.com	212
	time-g.netgear.com	212
	mcs.netgear.com	38
	updates.netgear.com	25
	arlo-device.messaging.netgear.com	15
	registration.ngxcld.com	8
	advisor.ngxcld.com	6
	xbroker2-z1.ngxcld.com	2
	presence.ngxcld.com	2
	vzwow62-z1-prod.vz.netgear.com	1
Nest Smoke Alarm	frontdoor.nest.com	37
	log-rts01-iad01.devices.nest.com	22
	czfe72.front01.iad01.production.nest.com	2
	czfe44.front01.iad01.production.nest.com	2
	czfe106.front01.iad01.production.nest.com	2
	czfe39.front01.iad01.production.nest.com	2
	czfe97.front01.iad01.production.nest.com	1
	czfe68.front01.iad01.production.nest.com	1
	czfe23.front01.iad01.production.nest.com	1
	czfe42.front01.iad01.production.nest.com	1
D-Link Motion Sensor	ntp1.dlink.com	23010
	wrpd.dlink.com	64
	signal.mydlink.com	64
	mp-eu-signal.auto.mydlink.com	48

Table 2: IoT devices and their traffic behaviour summary

Device	Country (and State if USA)
camera	IE NL CO-US EU GB
dlinkmotion	OR-US VA-US IE CA-US SG
JP echo	VA-US IE MA-US WA-US N/A-US GB DE TX-US NL FR HU NV-US CA-US CH CA RU NJ-US BR UA SE BG PL AT BE DK AU LV NY-US PT CZ NZ FL-US KR MD MD-US SK GR HR IA-US IL-US IT KH LU MN-US MO-US NO PY RO WI-US BY CO-US ES LT CL CN DE-US EE GE IN-US IS JP KG KS-US LI MI-US NC SG TR ID-US UT-US
foobot	IE
googlehome	CA-US N/A-US NE-US
hivehub	IE WA-US CA-US DE
nestsmoke	VA-US
smartplug	CN MD-US CA GB RU FR DE IE TX-US SE CA-US SG KR JP HK NJ-US KZ FL-US TW KH AM ID NL NV-US N/A-US IQ NY-US DE-US SA AU IR NO WA-US VA-US NE-US MN-US CO-US TH MY GA-US LK GE DK MO-US IN UT-US MI-US IL-US KG AZ PH WI-US IA-US ID-US MV NC-US KS-US OH-US OR-US PK PA-US VN RI-US MA-US IN-US AZ-US MM ES VT-US EU IL AR-US AE BD ME-US UZ SC-US TR AL-US NP

Table 3: Packets transmitted to different countries