Efficient Probing of Heterogeneous IoT Networks

Lionel Metongnon*, Eugène C. Ezin†, Ramin Sadre*
*Université catholique de Louvain, Belgium †Université d'Abomey-Calavi, Benin
Email: lionel.metongnon,ramin.sadre@uclouvain.be Email: eugene.ezin@imsp-uac.org

***Abstract*—The Internet of Things leads to the inter-connectivity of a wide range of devices. This heterogeneity of hardware and software poses significant challenges to security. Constrained IoT devices often do not have enough resources to carry the overhead of an intrusion protection system or complex security protocols. A typical initial step in network security is a network scan in order to find vulnerable nodes. In the context of IoT, the initiator of the scan can be particularly interested in finding constrained devices, assuming that they are easier targets. In IoT networks hosting devices of various types, performing a scan with a high discovery rate can be a challenging task, since low-power networks such as IEEE 802.15.4 are easily overloaded. In this paper, we propose an approach to increase the efficiency of network scans by combining them with active network measurements. The measurements allow the scanner to differentiate IoT nodes by the used network technology. We show that the knowledge gained from this differentiation can be used to control the scan strategy in order to reduce probe losses.**

***Index Terms*—IoT, 6LoWPAN, Network scans, Fingerprinting.**

I. Introduction

The Internet of Things (IoT), as an extension of the classical Internet, is leading to the inter-connectivity of a wide range of devices. While end nodes in the classical Internet confront us with two or three dominating operating systems, CPU families and network technologies, IoT technologies depict a much larger variety. With respect to network communication, we see for example technologies such as GSM for long-range communication, IEEE 802.11 for local area networks, and IEEE 802.15.4-based protocols for power-constrained devices.

This heterogeneity poses significant challenges to the deployment of IoT solutions. Obviously, this also applies to the area of security. Constrained IoT devices often do not have enough resources to carry the overhead of an intrusion protection system or complex security protocols. Furthermore, some devices are deployed in such a way that they are easily ”forgotten” by users, network managers or they are difficult to update compared to servers or end-user personal computers (which explains the recent successful attacks against home routers and CCTV cameras) [1], [2], [3]. It can be expected that IoT infrastructures and applications will become more and more interesting for criminal hackers in the near future.

If an attacker wants to compromise IoT (or other) nodes, a typical first step is to scan the network in order to find IP addresses used by the nodes. Ideally, the scan also involves some kind of finger-printing to identify the type of hardware, operating system and applications behind those addresses. For the reasons explained above, we may also find that the initiator of the scan is particularly interested in finding constrained devices, assuming that they are easier targets with their weaker security mechanisms. In general, one wants to perform the scan as efficient as possible in terms of duration and number of exchanged packets, since scans with long duration or high bandwidth can have a significant impact on the network. This is especially important when IPv6 is used because of its large address space. However, performing a scan with a high discovery rate can be a challenging task in an IoT environment, since low-power networks such as IEEE 802.15.4 are easily overloaded.

In this paper, we propose an approach to increase the efficiency of network scans. For this, we assume that the IoT is a heterogeneous environment hosting a mix of devices with different capabilities and using different communication technologies. In particular, we consider an infrastructure of constrained devices communicating over low-power wireless IEEE 802.15.4 links and more powerful devices using the faster WiFi (IEEE 802.11).

Our approach is based on round trip time measurements with variation of probe size and probing speed. The measurements allow the scanner to differentiate IoT nodes by the used network technology. We show that the knowledge gained from this differentiation can be used to control the scan strategy in order to reduce probe losses and, hence, increase the efficiency of the scan process.

To validate the feasibility of our approach, we have built an ns3 model of a mixed IoT infrastructure connected to the Internet and simulated network scans using our measurement scheme. Our experiments show that our approach indeed allows more efficient scans of heterogeneous IoT networks with speed improvement.

This paper is organized as follows: We present related work in Section II. In Section III, we introduce our approach. We present the experimental methodology in Section IV and discuss results in Section V. We conclude the paper in Section VI.

II. Related Works

The technique using active measurements that we develop in our approach identify the network technology used inside the IoT nodes. In this section, we discuss related works.

1) Available Bandwidth Estimation: Available bandwidth estimation (ABE) injects probes into the network for measurement and is commonly used in wired or wireless networks to determine the throughput of a particular network link or of an entire end-to-end path. Different approaches exist based on the network type. Since the goal of ABE is to obtain an accurate estimation of the bandwidth, most approaches send a series of

978-3-901882-89-0 @2017 IFIP

equal-sized probing packets and increase the emission rate of the packets gradually [4] or use special flight patterns [5].

In wireless networks, bandwidth measurements depend also on the probe packet size and cross-traffic rate [6]. The added probing traffic and the end-to-end technique sometimes influence the measurement and that is more remarkable on multi-hop networks like 802.11 or 802.15.4. In [7], Farooq and Kunz proved that the MAC layer overhead leads also to congestion and influences the bandwidth estimation.

In this paper, our goal is to obtain a fast and light-weight distinction between nodes using IEEE 802.11 and 802.15.4. In contrast to ABE approaches, we achieve this by performing an initial RTT measurement and then adapting the probing properties to be able to quickly identify the network technology.

2) OS Fingerprinting: In the introduction, we have argued that a motivation for scanning a network can be to identify constrained devices. In this paper, we propose an indirect approach to achieve this based on the identification of the used network technology. A more direct approach is OS Fingerprinting. OS Fingerprinting has as goal to identify the OS on a particular node. In passive approaches, one monitors the network and analyzes the packets to find clues on the OS. This is possible because of the differences in TCP/IP stack implementations [8]. In an active approach, the node is probed with malformed packets coupled with particular payload. All OS have specific ways to handle such packets, which allows to identify them easily provided that a response is received.

Various ways exist to defeat fingerprinting methods [9]. In this paper, our starting point therefore is that the scanner has no reliable way to identify the OS with sufficient precision.

III. Proposed Approach

In this section, we explain our approach to increase the efficiency of network scans. We give some background on IoT networks in Section III-A. We describe the considered scenario in Section III-B. Our approach is described in Section III-C.

A. Background

The IoT is an aggregation of many technologies and many types of devices and nodes. Proposed use cases range from the inter-connectivity of home devices to sensoring for industrial facilities or environmental supervision. Various open and proprietary network technologies and protocols have been proposed to implement this inter-connectivity. An obvious choice to connect IoT devices is through TCP/IP over IEEE 802.11. However, while being fast and flexible, IEEE 802.11 is not very suitable for resource-constrained devices as typically proposed for the IoT. An alternative can be IEEE 802.15.4 [10]: The standard defines low-power wireless embedded radio communications with data rates ranging between 20 to 250 kbit/s depending upon the frequency. The physical layer payload is 127 bytes, with 72 to 116 bytes of payload available after link-layer framing, addressing, and optional security [11].

When connecting large IoT infrastructures to existing TCP/IP networks, the small address space of IPv4 becomes a limitation. Therefore, IPv6 with its much larger 128 bit addresses has early attracted the interest of researchers. However, IPv6 requires a minimum Maximum Transmission Unit (MTU) of 1280 bytes and does not support fragmentation [12], which conflicts with the small size of IEEE 802.15.4 frames. Therefore, the IETF defined 6LoWPAN, a compression and encapsulation layer that allows to transmit IPv6 packets over Low power Wireless Personal Area Networks.

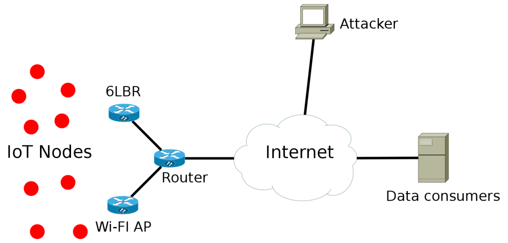

Fig. 1. Scenario considered in this paper

In practice, a border router (6LBR) is required to connect a 6LoWPAN network to a normal IPv6 network. Typically, the border router has also other tasks, including the generation of a compact 16-bit Interface Identifier (IID) for the nodes, the (de-)fragmentation of packets and the management of the routing between nodes, for example by using RPL [13].

B. Considered Scenario

We consider the scenario depicted in Fig. 1: An IoT network with a large number of nodes is connected to the Internet, in this way allowing the exchange of information between IoT nodes and users/servers that consume the data produced by them. To connect those nodes to the Internet, various protocols are used. As explained, we focus on IEEE 802.11 and IEEE 802.15.4 in the following. In practice, the operator of such a heterogeneous IoT will divide the network into smaller sub-networks based on the used network technology. In the example shown in the figure, we see two such sub-networks connected via 6LBRs or WiFi access points to a router.

IoT nodes connected to the Internet are exposed to many attacks. In this paper, we focus on network scans, a ”reconnaissance” attack type that is very frequent in the Internet. Once an attacker has guessed the address of the IoT network of interest, for example through information obtained from DNS servers or from passive monitoring of the data exchange with the servers, the attacker will scan the corresponding entire address range. We assume that the scanner probes for services that can be typically expected in an IoT, such as CoAP servers.

Scans in the Internet are usually performed very fast. Tools like zmap can scan the entire public IPv4 address space in less than 5 minutes at 10 Gbit/s [14]. When scanning an IPv6 network, the speed becomes even more important: Even the smallest network in IPv6 has more addresses than the entire IPv4. While the high scan rate of zmap is suitable for the Internet, it would be too aggressive for a network containing a mix of IoT nodes connected via IEEE 802.11 or IEEE 802.15.4. A scan rate adapted to the speed of modern Ethernet or even IEEE 802.11 is far too high for IEEE 802.15.4: the result would be a low *hit rate* due to many lost probes.

Note that the huge address space of IPv6 might give an illusion of safety against scans. However, by using techniques such as in [15], the potential target address range can be reduced. Since IoT node manufacturers and their Organizationally Unique Identifier (OUI) are known, an IoT network can be probed in a reasonable amount of time.

C. Description

Our proposed approach is based on Round Trip Time measurements. It consists of three steps.

Step 1 – Fast probing: The first step is to probe the entire address range of interest. To save time, we start with a high scan rate adapted to the fastest network technology we expect behind those addresses, i.e. IEEE 802.11 in our case. Naturally, for the scenario depicted in Fig. 1, such a fast scan will only work reliably for the IEEE 802.11 sub-network, while resulting in many lost probes in the slower IEEE 802.15.4 sub-network. Nevertheless, this first scan gives us a first insight into what parts of the address range are in use.

Step 2 – Probing with size variation: This step's goal is to identify what network technologies are used by the nodes discovered in step 1. To this end, we perform a slow scan against a randomly chosen small subset of those nodes.

As already explained, IEEE 802.15.4 with a speed of 250 kbit/s is much slower than IEEE 802.11 (11 Mbit/s for IEEE 802.11b). This results in a noticeable difference between the RTTs of probes. To amplify this difference, we modify the probe size during the second step. The key is the fragmentation performed by the 6LBR: IEEE 802.15.4 only supports 127 bytes for the maximum physical layer service data unit (PSDU) while IEEE 802.11 supports the minimum MTU of 1280 bytes required by IPv6. This means that probes larger than 128 bytes are fragmented by the 6LBR and result in a clear increase of the RTT for IEEE 802.15.4 nodes while the RTT for IEEE 802.11 nodes will stay relatively stable.

Step 3 – Slow re-probing: In this last step, we rescan at a lower speed only those parts of the address range where we have identified IEEE 802.15.4 nodes in step 2. The assumption is that our fast scan in step 1 has missed many nodes belonging to the IEEE 802.15 sub-network. By scanning with a lower speed we avoid probe losses and achieve a higher hit rate.

By starting with a fast scan (step 1) and slow-scanning (step 3) only those parts of the address range that were identified as (probably) containing IEEE 802.15.4 nodes (step 2), our approach achieves a good comprise between number of sent probes and scan duration. A non-adaptive scan strategy would have to scan the entire address range slowly to achieve a comparable hit rate.

IV. Experimental Methodology

We use ns3 (www.nsnam.org), which is one of the few simulators for heterogeneous networks with IEEE 802.11 and 6LoWPAN over IEEE 802.15.4. Ns3 is based on C++, very powerful and complete [16]. Ns3 picks consecutive UID for all the nodes in the simulation, i.e. for routers, servers, and 802.11 nodes, except for IEEE 802.15.4 nodes where the same UID is used for every sub-network. The current ns3 version 3.24.1 still lacks support for *RPL*, therefore we will restrict our experiments to single-hop networks with star topology. We do not consider this a limitation of our validation since a multi-hop network would make the RTT differences between IEEE 802.15.4 and IEEE 802.11 even more obvious.

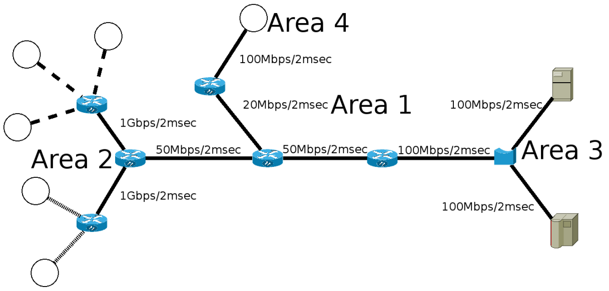

Fig. 2. Simulated network topology

Fig. 2 shows the simulated network. It consists of:

- *Area 1* contains the Internet and the border routers of the different sub-networks.
- *Area 2* represents the IoT network with its sub-networks for 802.11 and 802.15.4 nodes. Each node hosts a CoAP server on UDP port 5683.
- *Area 3* contains two servers that periodically collect data from the IoT nodes.
- *Area 4* contains the host that scans the IoT nodes by sending CoAP requests.

We call the legitimate traffic exchange between area 2 and 3 *background* traffic. We perform our experiments with *light* and *normal* background traffic. The IEEE 802.11 nodes send data packets of 512 bytes every 20 seconds for the light background traffic and every 10 seconds for the normal traffic to a server which replies automatically with a 512-byte packet. A second server sends 100-byte requests to all the IEEE 802.15.4 nodes every 5 seconds and receives a response of 512 bytes.

We let the simulation perform a warm-up time of a few seconds to initialize the routing protocol and to populate all the routing tables. We divide our validation into two parts:

1) Tests with one network technology: The goal of the first part is to verify our assumptions on the behavior of IEEE 802.11 and IEEE 802.15.4 when facing a simple scan that does *not* use our approach. To this end, we study the two network technologies separately. We perform scans by sending CoAP probes with different rates and sizes and estimate the RTT by measuring the time until a response (if any) is received.

For these tests, area 2 consists of three sub-networks with 100, 50 and 20 nodes, respectively. The links between the access points (IEEE 802.11) or 6LBR (IEEE 802.15.4) of the sub-networks and the main network have a data-rate/delay of 1 Gbps/2ms, 100 Mbps/2ms and 50 Mbps/2ms, respectively.

2) Tests with mixed network technologies: In the second part, we test our approach. We simulate a heterogeneous network in area 2, consisting of two sub-networks: The first contains 100 IEEE 802.15.4 nodes and the second contains 50 IEEE 802.11 nodes. The uplinks of the 6LBR router and the access point have a data-rate/delay of 1 Gbps/2ms. We first perform a fast scan (step 1), then probe 15 random nodes out of the found nodes to identify IEEE 802.15.4 nodes (step 2), and finally perform a slower scan on that part of the address range that contained those nodes (step 3).

V. Results

As explained, we first study the two network technologies separately in Section V-A and Section V-B, before testing our scan approach on a heterogeneous network in Section V-C.

A. Behavior of IEEE 802.11 nodes

In our first experiment, we scan area 2 with a constant moderate inter-probing time of $T = 80$ ms and three different probe sizes. The RTT results are shown in Fig. 3 and Fig. 4 for respectively light and normal background traffic. The three sub-networks containing the IEEE 802.11 nodes appear as three groups (from left to right). Since they do not use consecutive addresses, we have gaps without any measurement results.

We observe that the amount of background traffic has no influence on the RTTs, since it is relatively small compared to the available network bandwidth. Furthermore, we observe that the probe size only has a small effect. Also note that both figures show anomalous results for the first sub-network. We believe that these are either simulation glitches or warm-up effects in ns3. We have decided to include these results here for sake of completeness and repeatability.

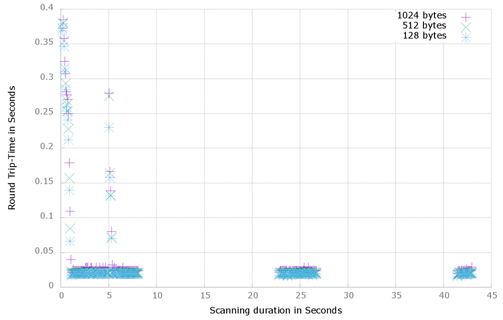

Fig. 3. RTT for IEEE 802.11; light background, $T = 80$ ms.

Fig. 5 and Fig. 6 show the obtained results when scanning with a shorter inter-probing time T of 20 ms and 10 ms, respectively, and normal background traffic. The results show a slightly increased RTT compared to the test with $T = 80$ ms and stable results for probes of 128 bytes and 512 bytes. However, we can clearly see that the 1024-byte probes suffer from queueing effects when sent with $T = 10$ ms: The RTTs successively increase during the scan and reach up to 80 ms.

Independently from the probe size and probing speed, the scans in the above experiments were always able to find 100% of the nodes, i.e. no probes were lost.

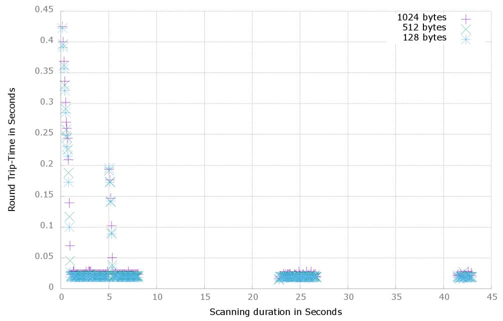

Fig. 4. RTT for IEEE 802.11; normal background; $T = 80$ ms.

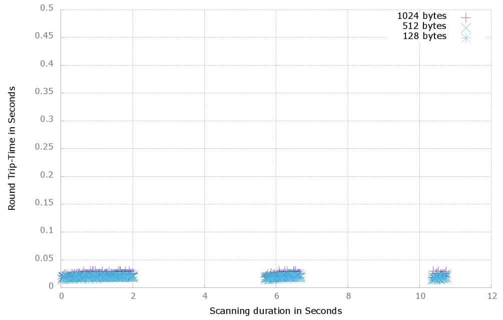

Fig. 5. RTT for IEEE 802.11; normal background, $T = 20$ ms.

B. Behavior of IEEE 802.15.4 nodes

In our second experiment, we simulate area 2 with three IEEE 802.15.4 sub-networks. Again, we begin with a constant moderate inter-probing time of 80 ms and three different probe sizes. The results for light and normal background traffic are shown in Fig. 7 and Fig. 8. The corresponding boxplot in Fig. 9 shows that the probe size has a great impact on the measured RTT. Even at this slow scan speed, the results for 128-byte probes and 1024-byte probes differ by more than one order of magnitude. This is due to the low bandwidth and small MTU of IEEE 802.15.4, as explained in Section III-C. Furthermore, we observe a small influence of the background traffic on 512-byte and 1024-byte probes in the first (and largest) sub-network.

Fig. 10 and Fig. 11 show the results for $T = 20$ ms and

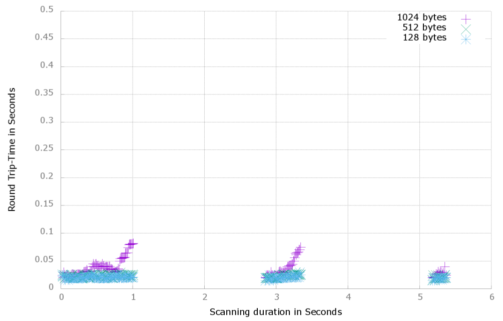

Fig. 6. RTT for IEEE 802.11; normal background; $T = 10$ ms.

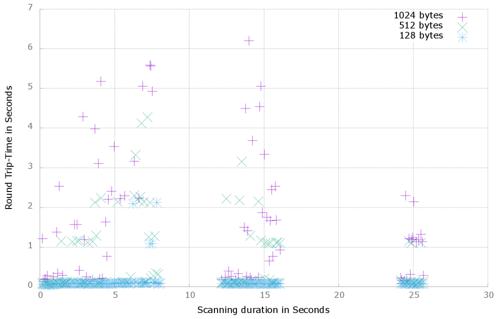

Fig. 7. RTT for IEEE 802.15.4; light background, $T = 80$ ms.

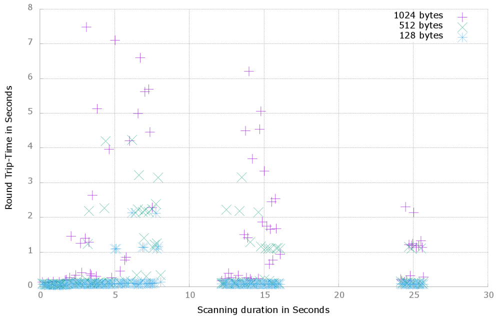

Fig. 8. RTT for IEEE 802.15.4; normal background, $T = 80$ ms.

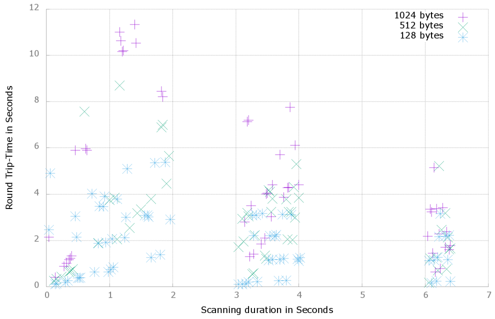

Fig. 10. RTT for IEEE 802.15.4; normal background, $T = 20$ ms.

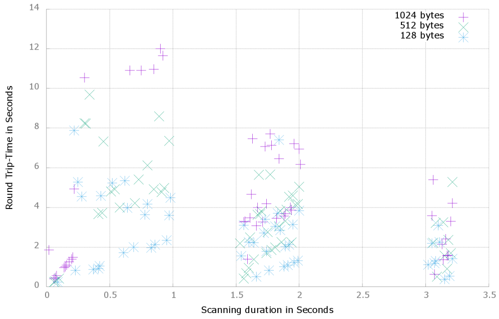

Fig. 11. RTT for IEEE 802.15.4; normal background, $T = 10$ ms.

$T = 10$ ms, respectively. We observe a very high RTT with large variations for all probe sizes. In addition, a second effect becomes visible: a large number of probes are lost because of the high scan speed. Fig. 12 shows the percentage of nodes discovered by the scan with normal background traffic. For small probe sizes (≤ 200 bytes) and slow scan speed ($T = 80$ ms), more than 95% of the nodes are found. This number drops quickly for faster scans and larger probes. For example, only 42.94% of the nodes are found with 512-byte probes and $T = 40$ ms.

C. The complete heterogeneous scenario

Our previous experiments have shown that IEEE 802.11 and 802.15.4 networks behave quite differently when probed at fast speed or with large probe packets, resulting in large RTTs and high probe losses in 802.15.4 networks. In the following, we measure how our scan approach makes use of these effects to improve scan efficiency. As described in Section IV, we simulate an IoT network with two sub-networks, containing 100 IEEE 802.15.4 nodes resp. 50 IEEE 802.11 nodes.

The first step of our approach is a fast scan of the entire area 2. Fig. 13 shows the measured RTT with $T = 10$ ms and a 32-bytes probe size under normal background traffic. The low inter-probing time results in the desired very short scan duration, which works well for the second sub-network with IEE 802.11 nodes (and would also work well for wired hosts), but results in many probe losses in the first sub-network containing the IEEE 802.15.4 nodes.

At this point, one might be tempted to directly classify the first sub-network as an IEEE 802.15.4 network and argue that

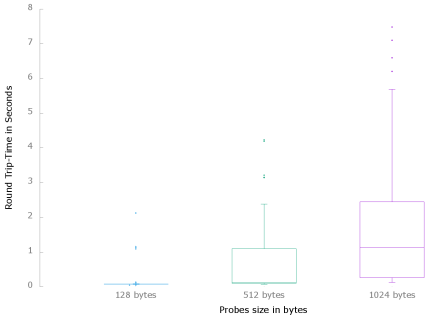

Fig. 9. Probes size impact on IEEE 802.15.4; normal background, $T = 80$ ms.

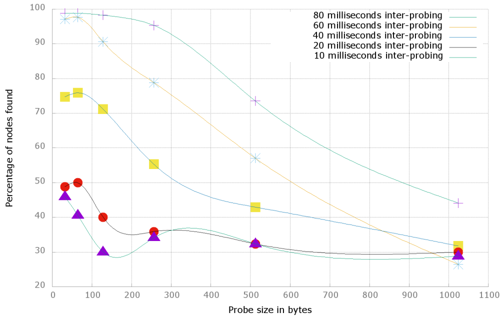

Fig. 12. Number of IEEE 802.15.4 nodes found; normal background.

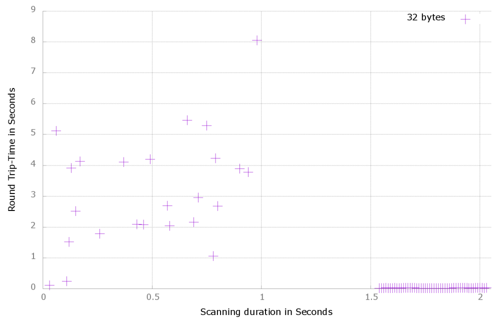

Fig. 13. Mixed network; normal background, $T = 10$ ms, 32-byte probes.

step 2 of the approach is not really needed since Fig. 13 clearly shows a difference between IEEE 802.15.4 and 802.11 nodes. However, one should not forget that an IoT installation does not necessarily contain two types of networks and there might be other reasons for a large RTT, for example a generally slow network connection, slow reaction speed of nodes, etc.

In step 2, the scan algorithms selects randomly a few nodes and sends slow probes to them with different sizes. Fig. 14 and Fig. 15 show the measured RTTs for probes of size 32-bytes and 512-bytes and $T = 80$ ms. The large increase of the RTT for 512-byte probes despite the slow scan speed clearly identifies the IEEE 802.15.4 nodes.

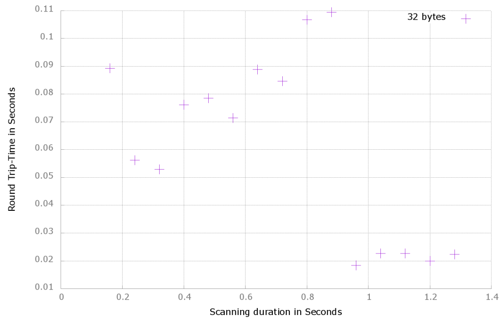

Fig. 14. Selected nodes; normal background, $T = 80$ ms, 32-byte probes.

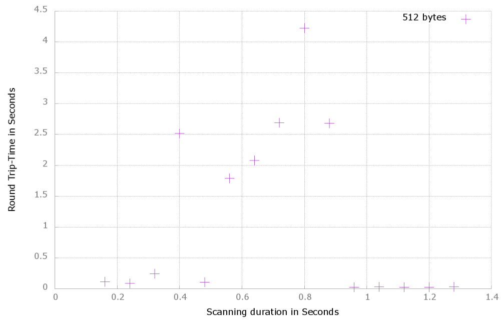

Fig. 15. Selected nodes; normal background, $T = 80$ ms, 512-byte probes.

Finally, in step 3 the part of the address range that contains IEEE 802.15.4 nodes is rescanned at slow speed ($T = 80$ ms) and small probes (32-bytes). The results are shown in Fig. 16.

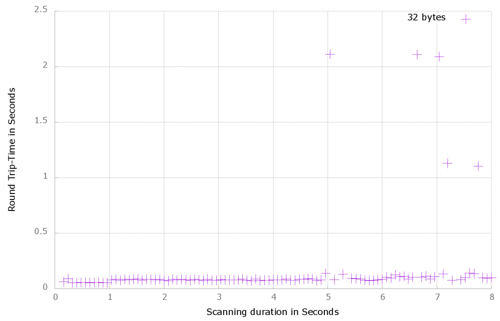

Fig. 16. Mixed network; normal background, $T = 80$ ms, 32-byte probes.

In contrast to the fast scan in step 1, we are able to find 96% of the IEEE 802.15.4 nodes. This means that we have achieved a discovery percentage comparable to a slow scan (see Fig. 12), however without having to scan slowly the *entire* IoT network. Our approach takes 2 seconds for the fast scan in step 1, less than 1.5 seconds for the selective probing in step 2, and 8 seconds for the partial slow scan in step 3, i.e., 11.5 seconds in total, compared to 16 seconds for a slow scan of the entire IoT network.

VI. Conclusion

In this paper, we have presented a light-weight approach for the efficient probing of heterogeneous IoT networks consisting of IEEE 802.11 and IEEE 802.15.4 sub-networks. Our three-phased approach combines RTT measurements and variations of the probe size and speed to identify the communication technology used by the network hosts. Using this information, we can achieve a high discovery rate by slowly scanning those parts of the IoT network where IEEE 802.15.4 nodes have been detected. We have validated our approach by simulating an IoT network with 170 nodes and two servers that create legitimate (non-probing) background traffic.

The gained knowledge on the network technology used by a particular host can help to fingerprint it. For example, the usage of IEEE 802.15.4 indicates a constraint-restricted node, and therefore allows certain conclusions on its OS and application software. This insight can be exploited by the probing party in various ways. Our paper has shown that one of the main characteristics of the Internet of Things, namely its heterogeneity in terms of software and hardware, can lead to new attack patterns.

As future work, we plan to validate our approach on more complex networks, in particular multi-hop 6LoWPAN networks. To this end, we are experimenting with ns3 versions with RPL support. In a multi-hop network, it will be even easier to detect IEEE 802.15.4. We are also working on other means to identify and fingerprint IoT nodes. We are also thinking about a collaborative intrusion detection system to mitigate this kind of scanning.

References

[1] O. Gayer, O. Wilder, and I. Zeifman. Cctv botnet in our own back yard. https://www.incapsula.com/blog/cctv-ddos-botnet-back-yard.html.

[2] L. Constantin. Thousands of hacked cctv devices used in ddos attacks. http://www.pcworld.com/article/3089346/security/thousands-of-hacked-cctv-devices-used-in-ddos-attacks.html.

[3] D. Cid. Large cctv botnet leveraged in ddos attacks. https://blog.sucuri.net/2016/06/large-cctv-botnet-leveraged-ddos-attacks.html.

[4] C. Sarr, C. Chaudet, G. Chelius, and I. G. Lassous. Bandwidth estimation for ieee 802.11-based ad hoc networks. *IEEE transactions on Mobile Computing*, 7(10):1228–1241, 2008.

[5] V. J. Ribeiro, R. H. Riedi, R. G. Baraniuk, J. Navratil, and L. Cottrell. pathchirp: Efficient available bandwidth estimation for network paths. In *Passive and active measurement workshop*, 2003.

[6] A. Johnsson, B. Melander, and M. Björkman. Bandwidth measurement in wireless networks. In *Challenges in Ad Hoc Networking*, pages 89–98. Springer, 2006.

[7] M. O. Farooq and T. Kunz. Proactive bandwidth estimation for ieee 802.15.4-based networks. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–5. IEEE, 2013.

[8] R. Beverly. A robust classifier for passive tcp/ip fingerprinting. In *International Workshop on Passive and Active Network Measurement*, pages 158–167. Springer, 2004.

[9] G. Taleck. Ambiguity resolution via passive os fingerprinting. In *International Workshop on Recent Advances in Intrusion Detection*, pages 192–206. Springer, 2003.

[10] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. **RFC 4944** transmission of ipv6 packets over ieee 802.15. 4 networks, 2007.

[11] J. Hui and P. Thubert. **RFC 6282** compression format for ipv6 datagrams over ieee 802.15. 4-based networks, 2011.

[12] S. E. Deering. **RFC 2460** internet protocol, version 6 (ipv6) specification, 1998.

[13] T. Winter. **RFC 6550** rpl: Ipv6 routing protocol for low-power and lossy networks, 2012.

[14] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman. Zippier zmap: internet-wide scanning at 10 gbps. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[15] F. Gont and T. Chown. **RFC 7707** network reconnaissance in ipv6 networks, 2016.

[16] Sardar M Bilalb, Mazliza Othmana, et al. A performance comparison of network simulators for wireless networks. *arXiv preprint arXiv:1307.4129*, 2013.