Fast and Efficient Probing of Heterogeneous IoT Networks

Lionel Metongnon[1,2], Ramin Sadre[1]

[1] *{firstname.lastname}@uclouvain.be*
Université catholique de Louvain, Belgium
[2] *Université d'Abomey-Calavi, Benin*

SUMMARY

The Internet of Things leads to the inter-connectivity of a wide range of device types running an equally wide range of operating systems and applications. 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. Furthermore, they are often not properly managed and updated.
Network scans are a valuable tool to discover vulnerable devices. In the context of IoT, the initiator of the scan can be particularly interested in finding constrained devices, assuming that they are easier targets for attacks. However, in IoT networks hosting devices of various types, performing a scan with a high discovery rate can be a challenging task, since a scan working well for, e.g., a WiFi network might easily overload a low-power network such as IEEE 802.15.4.
In this paper, we propose an approach to increase the efficiency of network scans in heterogeneous environments by combining them with active round-trip time measurements. The measurements allow the scanner to differentiate IoT nodes by the used network technology. Using the knowledge gained from this differentiation, our approach adapts the scan strategy in order to reduce probe losses, and hence the speed and efficiency of the scan. We validate our approach through simulations of a mixed IoT infrastructure consisting of WiFi and multi-hop IEEE 802.15.4 sub-networks.

Received ...

KEY WORDS: IoT, 6LoWPAN, Network scans, Fingerprinting.

1. INTRODUCTION

Nowadays the number of interconnected devices has already surpassed the number of physical users, and the Internet of Things (IoT) will make this difference even larger [13]. At the same time, the heterogeneity of deployed hardware and software is increased by the IoT. While the classical Internet confronts us with two or three dominating operating systems, CPU families and network technologies, IoT devices 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 is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

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 and network managers, or they are difficult to update compared to servers or end-user personal computers. This has resulted in the recent successful attacks against home routers and CCTV cameras [14, 7, 6]. It can be expected that IoT infrastructures and applications will become more and more interesting for criminal hackers in the near future.

In computer security, networks scans are a valuable tool equally used by network managers as well as malicious parties. With the increasing popularity of IoT, we expect that network administrators will use scan algorithms like the one presented here to keep track of all the devices deployed in their network. Depending on its nature, a network scan allows to discover the IP addresses used by the devices. Ideally, a 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, the initiator of the scan might be particularly interested in finding constrained devices, assuming that they are easier targets for attacks 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 also 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 slow 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 faster IEEE 802.11 (WiFi) links.

Our approach is based on round-trip time (RTT) measurements with variation of probe size (i.e. the size of the probing packet) and probing speed (i.e. the number of probes sent per time unit). The measurements allow the scanner to differentiate 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.

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. As a secondary goal, our paper shows that designers of intrusion detection systems will have to face new attack techniques adapted to IoT environments.

The remainder of this paper is organized as follows: we present related work in Section 2. In Section 3, we provide a brief introduction to networking for constrained devices. In Section 4, we introduce our approach. We present the experimental methodology in Section 5 and discuss results in Section 6. We conclude the paper in Section 7.

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

2. RELATED WORKS

In this section, we discuss publications related to our paper. Apart from the obvious connection to existing work on network scans, our approach makes use of techniques that are related to bandwidth estimation and fingerprinting.

2.1. Network scans

Network scan implementations can be differentiated by the targeted network layer, the used protocol, the scanning speed, the number of scanners, and the scan targets [2]. For large Internet-wide scans, trending tools are ZMap [11] and MASSCAN [10], which can probe the entire IPv4 address range in less than a day with enough bandwidth. In [8], the authors point out new scanning procedures using botnets. By employing a large number of scanning hosts, the detection by simple intrusion detection systems can be avoided since the individual scanners can operate at slow speed and still achieve together a high probing speed. Recently, the usage of compromised IoT nodes by scanners has become popular, too [21].

In our paper, we consider an environment where the choice of an appropriate scan speed is important in order to achieve a satisfying discovery rate. Moreover, that scan speed can vary considerably across hosts in the same target network because of the usage of different communication technologies. In our approach, we use active RTT measurements and a multi-phase scan procedure. This paper is an extension of prior work [19]; we have added a detailed description of the scan algorithm, discussions on its properties, and, most important, we now also discuss multi-hop scenarios with RPL routing.

2.2. 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 equal-sized probing packets and increase the emission rate of the packets gradually [23] or use special flight patterns [22].

In wireless networks, bandwidth measurements depend also on the probe packet size and cross-traffic rate [18]. 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 [12], 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.

2.3. 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 [4]. In an active approach, the node is probed with malformed packets coupled with carefully chosen payloads. 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 [25]. In this paper, our starting point therefore is that the scanner has no reliable way to identify the OS with sufficient precision.

3. BACKGROUND: NETWORKING FOR CONSTRAINED DEVICES

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 important principle of the IoT is that nodes should inter-connect across existing network infrastructures. We assume in the following that IoT nodes are accessible from outside. An obvious choice to connect IoT devices is through 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. The IEEE 802.15.4 standard [3] defines low-power wireless embedded radio communication with data rates ranging between 20 to 250 kbit/s depending on the frequency [20]. The physical layer payload is 127 bytes, with 72 to 116 bytes of payload available after link-layer framing, addressing, and optional security [17].

When connecting large IoT networks to existing 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 [9], 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. Figure 1 depicts a comparison between the IP and 6LoWPAN protocol stacks.

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 the Routing Protocol for Low-Power and Lossy Networks (RPL) [26]. RPL allows multi-hop routing inside 6LoWPAN networks and defines control messages that are used by the border router and the nodes to construct and maintain a logical routing graph.

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

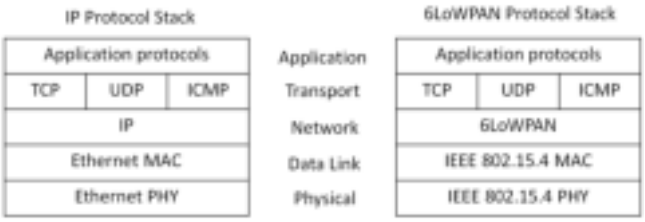

Figure 1. IP and 6LoWPAN protocol stacks (after [24])

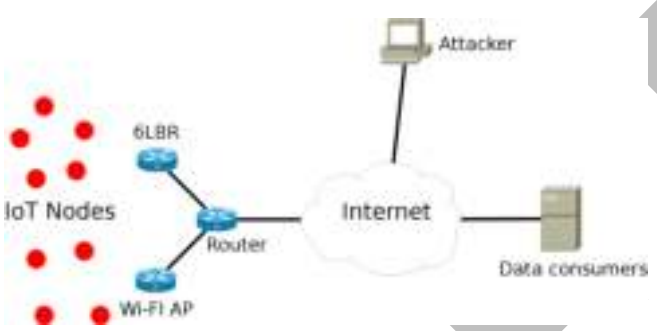

Figure 2. Scenario considered in this paper

For the sake of completeness we want to note that various other wireless protocol specifications exist for constrained devices, such as ZigBee, Z-Wave, INSTEON and Wavenis [15]. Most noteworthy, like 6LoWPAN, ZigBee is based on IEEE 802.15.4. ZigBee is a specification of protocols defined by the ZigBee alliance* for embedded applications requiring low-power consumption and low data-rates. The specification comprises network protocols as well as high-level application protocols based on a component model. However, we will use 6LoWPAN for constrained devices in the following because it is well supported by tools and IoT software/hardware, and because hosts in a 6LoWPAN network can easily inter-operate with hosts in the Internet.

4. PROPOSED APPROACH

In this section, we explain our approach to increase the efficiency of network scans. We describe the considered scenario in Section 4.1. The scan procedure is described in Section 4.2 and 4.3.

4.1. Considered Scenario

We consider the scenario depicted in Figure 2: 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 (with 6LoWPAN) in the following. In practice, the operator of such a heterogeneous

*http://www.zigbee.org/

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 an WiFi access point and a 6LoWPAN border router (6LBR) 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, but also used by network administrators to discover misconfigured or forgotten hosts. Once the scanner 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 data consumers in the Internet, 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.

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 [1]. 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* or *discovery rate* (i.e. the fraction of discovered nodes) 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 [16], the potential target address range can be reduced. Since typical IoT node manufacturers and their Organizationally Unique Identifier (OUI) are known, an IoT network can be probed in a reasonable amount of time.

4.2. General description

Our proposed approach is based on round-trip time (RTT) 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 small probe size and a high scan rate adapted to the fastest network technology we expect behind those addresses, i.e. a variant of IEEE 802.11 in our case. Naturally, for the scenario depicted in Figure 2, 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 as maximum physical layer service data unit (PSDU) while IEEE 802.11 supports the minimum

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

MTU of 1280 bytes required by IPv6. This means that probes larger than 127 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 even for larger packets. If nodes in the IEEE 802.15.4 are more than one hop away from the border router, the RTT is even further increased.

Step 3 – Slow re-probing In this last step, we re-scan at a lower speed and with small probes 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 re-scanning with a lower speed we avoid probe losses and achieve a higher hit rate.

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

4.3. Algorithm

Algorithm 4.1 shows the basic building block of the implementation, a (non-adaptive) scan procedure. It sends probes of size $pSize$ with probing speed $pSpeed$ to a list of addresses. This list consists either of all addresses of a sub-network (indicated by $ratio = ALL$; line 3) or of individual addresses sampled from the provided address list with ratio $ratio$. We try to take at least 10 addresses (line 6).

The main procedure is shown in Algorithm 4.2. It obtains a list of sub-networks ns to scan. To simplify the explanations in the following, the depicted algorithm processes sub-networks one by one (line 1), although a concrete implementation does not have to follow this order. It first performs a fast scan with small probes for all addresses in a sub-network (line 2), in this way obtaining a list of addresses $received_1$ that responded and an average RTT rtt_1. Some (with ssize ratio) of the addresses that responded are scanned again slowly with large probes (line 3). Since protocols like UDP are unreliable, the second scan doesn't always receive a satisfying number of replies for the following steps. Therefore, we repeat the slow scan up to three times (lines 4–7).

A comparison is done with the mean from the first and the second phase to trigger error handling (line 8). If an error occurred (unexpected delays in the first scanning), the two first phases are redone but only on 10% of the alive hosts previously found. The final classification of the sub-networks is based on two criteria: (i) the mean of the second phase is greater than $tlimit$ (typically 0.5 seconds), and (ii) we have at least a certain $factor$ (typically 4) between the mean of the first and the second phase. Finally, the network type is saved and used in the last phase to rescan and find more nodes in the 802.15.4 networks.

In our experience, the probability of false positives (i.e. sub-networks wrongly classified as 802.15.4) is quite low due to the dual condition to determine the type of networks (line 13 of algorithm 4.2). The values for the threshold and the factor work for a very wide range of scenarios and have been verified by us in simulation and with real devices. We have noticed in

Algorithm 4.1: Basic scanning

Input: *listAddr* : addresses to scan, *pSize* : probe size, *pSpeed* : probe speed, *ratio*

```
as ← queue() // queue with all the address we will scan
if ratio == ALL then
    as := all addresses from the sub-network listAddr
end
else
    as := addresses randomly picked from listAddr with ratio ratio (but minimum 10)
end
send probes of size pSize with probe speed pSpeed to all addresses in as;
wait at most 5 seconds for replies;
received := all addresses that replied;
rtt := average measured RTT over all addresses in received;
return (received, rtt)
```

Algorithm 4.2: Main procedure

Input: *ns* : sub-networks to scan

```
for each sub-network n in ns do
    (received_1, rtt_1) = scan(n, psize_small, pspeed_fast, ALL);
    (received_2, rtt_2) = scan(received_1, psize_large, pspeed_slow, ssize);
    while size(received_2) < mvalue and number of attempts < 3 do
        (received', rtt') = scan(received_1, psize_large, pspeed_slow, ssize);
        add (received', rtt') to (received_2, rtt_2)
    end
    while rtt_2 < rtt_1 and number of attempts < 3 do
        (received_1, rtt_1) = scan(activeAddrList, psize_small, pspeed_fast, ssize);
        (received_2, rtt_2) = scan(activeAddrList, psize_large, pspeed_slow, ssize);
    end
    if rtt_2 >= rtt_1 then
        if rtt_2 >= tlimit and rtt_2 >= factor · rtt_1 then
            type[n] := 802.15.4
        end
        else
            type[n] := 802.11;
            received[n] := received_1
        end
    end
    else
        // No classification possible for this sub-network
        abort
    end
end
// rescan sub-networks identified as 802.15.4
for each sub-network n with type[n] == 802.15.4 do
    (received[n], rtt) = scan(n, psize_small, pspeed_slow, ALL)
end
return (received, type)
```

our experiments (see Section 6) that the probe size doesn't affect much the 802.11 networks in

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

Figure 3. Simulated network topology

contrast to the 802.15.4 ones where factors of 4 or greater between the RTT of the first step and second step are measured.

On the other hand, there is a significant chance for false negatives related to the amount of nodes available for the second phase and the number of replies collected. In fact we have observed that when less than 5 nodes are used in the second phase, we may get a very small RTT, especially if we don't get all the replies in one go due to losses. To handle such situations, we redo the scans when the number of obtained nodes is too low or when the measured RTT behaves not in the expected way (lines 4 and 8). In the worst case, a final test to detect strange behavior between the first and second phase (line 12) will abort the classification.

5. EXPERIMENTAL METHODOLOGY

We use ns3 (www.nsnam.org), which is one of the few simulators supporting heterogeneous networks with IEEE 802.11 and 6LoWPAN over IEEE 802.15.4. Ns3 is written in C++, very powerful and complete [5]. 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. For the latter the same UID is used for every sub-network. The current ns3 version 3.26 still lacks support for RPL. We have used an experimental version with RPL support for our experiments with multi-hop networks.

Some parameters of the algorithm 4.2 have been set based on multiple experiments to assure the best compromise between time consumption and efficiency of our method. We choose $ssize = 10\%$ representing the minimum amount of nodes needed for the second phase. $tlimit = 0.5s$ is the time limit and $factor = 4$ the ratio used to differ between 802.15.4 and 802.11 nodes.

Figure 3 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 802.11g and 802.15.4 nodes. Each node hosts a CoAP client 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 area 3 *background* traffic. We perform our experiments with *light* and *normal* background traffic.

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

5.1. Tests with one network technology

The goal of the first part is to verify our assumptions on the behaviour 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. 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 bandwidth/delay of 1 Gbps/2ms, 100 Mbps/2ms and 50 Mbps/2ms, respectively. 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. The following experiments are performed:

IEEE 802.11 network The IEEE 802.11 devices form a circle of radius 80 with the access point at its center. The probing starts around 1000 seconds after the beginning of the experiment with different inter-probing times. 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 one of the servers which replies automatically with a 512-byte packet. In addition to this push-style communication, all nodes also have the CoAP port open.

IEEE 802.15.4 network without RPL routing The IEEE 802.15.4 nodes also form a circle with radius 80 with the 6LBR as the center. The probing starts 5000 seconds after the beginning of the experiment. For the background traffic, one of the servers sends a 30 bytes CoAP request to the IEEE 802.15.4 nodes consecutively every 20 seconds for the light background traffic and every 10 seconds for the normal traffic. The nodes reply with a 512-byte response. Note that all nodes are in the communication range of the 6LBR and no routing protocol is used.

IEEE 802.15.4 network with RPL routing For this test, we modify the above IEEE 802.15.4 setup: We place the nodes on two different circles (radius 80 and 160) around the 6LBR and use RPL for multi-hop routing. To compare results, we also test the original setup (one circle) with RPL. The background traffic is generated as described in the test without RPL. RPL was configured with the *ETX metric* and *objective function 0* [26].

5.2. Tests with mixed network technologies

In this second set of tests, we validate our proposed approach for efficient scanning. 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 up-links of the 6LBR router and the access point have a bandwidth/delay of 1 Gbps/2ms. We first perform a fast scan

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

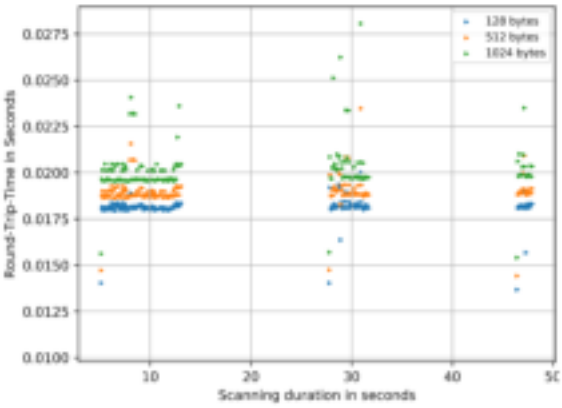

Figure 4. RTT for IEEE 802.11; light background, $T = 80$ ms.

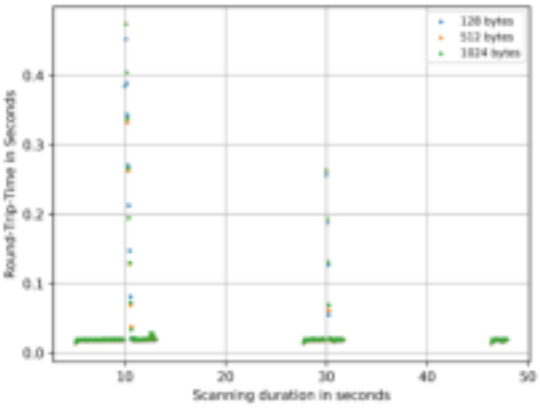

Figure 5. RTT for IEEE 802.11; normal background; $T = 80$ ms.

(step 1), then probe 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).

6. RESULTS

Following our methodology described in the previous section, we first study the two network technologies separately in Section 6.1 (IEEE 802.11) and Section 6.2 (IEEE 802.15.4), before testing our scan approach on a heterogeneous network in Section 6.3.

6.1. Behavior of IEEE 802.11 nodes

In our first experiment, we scan area 2 with a constant and moderate inter-probing time of $T = 80$ ms and three different probe sizes. The measured RTTs of the scanned nodes are shown in Figure 4 and Figure 5 for respectively light and normal background traffic. The three sub-networks with 100, 50 and 20 nodes are visible as three distinct groups (from left to right) in the result. 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 Figure 5 shows a few outliers. We believe that these are simulation glitches or caused by background processes. We have decided to keep them for sake of completeness and repeatability.

Figure 6 and Figure 7 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. We can also see that the 1024-byte probes suffer from a small queuing effect in the largest (left-most) sub-network when sent with $T = 10$ ms.

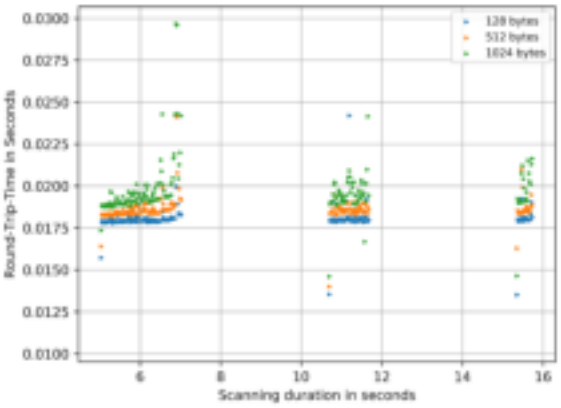

Figure 6. RTT for IEEE 802.11; normal background, $T = 20$ ms.

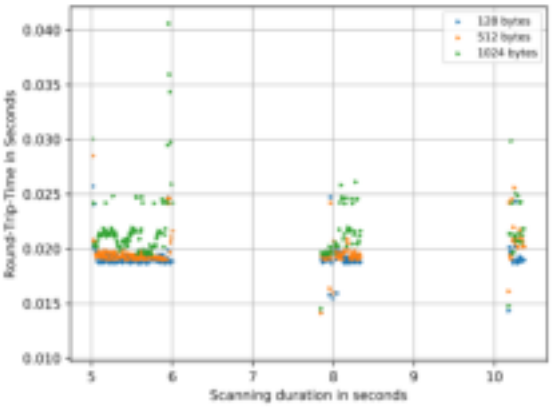

Figure 7. RTT for IEEE 802.11; normal background; $T = 10$ ms.

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.

6.2. Behaviour of IEEE 802.15.4 nodes

In our second set of experiments, we simulate area 2 with three IEEE 802.15.4 sub-networks. We present in Section 6.2.1 the results when the nodes are directly connected to the 6LBR without RPL, followed by the results with RPL in Section 6.2.2.

6.2.1. IEEE 802.15.4 nodes without RPL

Again, we begin with a constant and moderate inter-probing time of $T = 80$ ms and three different probe sizes. The results for light and normal background traffic are show in Figure 8 and Figure 9. The corresponding box-plot in Figure 10 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 one order of magnitude and the queuing effect is much more pronounced than in the experiments with IEEE 802.11. This is due to the low bandwidth and small MTU of IEEE 802.15.4, as explained in Section 4.2.

Figure 11 and Figure 12 show the results for $T = 20$ ms and $T = 10$ 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. Figure 13 shows the percentage of nodes discovered by the scan with normal background traffic. For small probe sizes (≤ 200 bytes) and slow scan speed ($T \geq 80$ ms), more than 95% of the nodes are found. This number drops quickly for faster scans and larger probes. For example, less than 30% of the nodes are found with 512-byte probes and $T = 40$ ms.

6.2.2. IEEE 802.15.4 nodes with RPL

We begin with the same single-hop configuration as in the previous test, but this time we activate RPL. The results for light and normal background traffic are shown in Figure 14 and Figure 15. The disparity between 128-byte and 1024-byte probes becomes larger, which is also clearly visible in the box-plot in Figure 16.

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

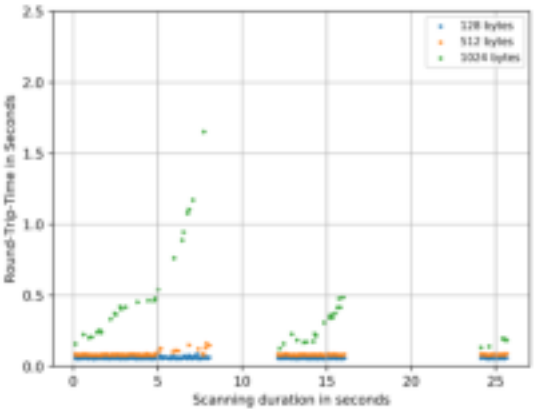

Figure 8. RTT for IEEE 802.15.4 without RPL; light background, $T = 80$ ms.

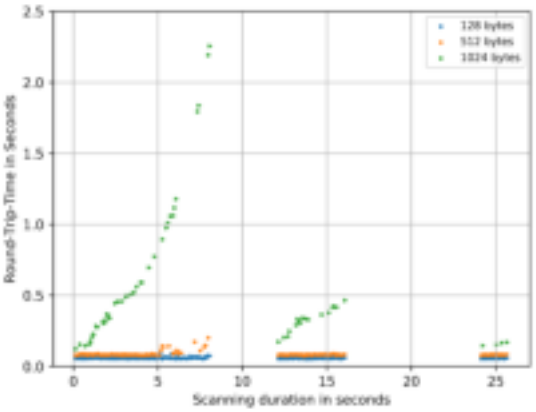

Figure 9. RTT for IEEE 802.15.4 without RPL; normal background, $T = 80$ ms.

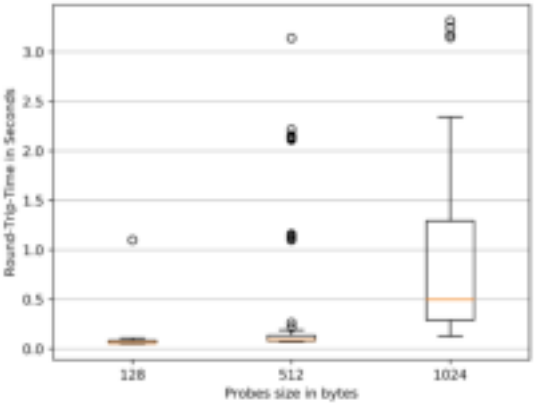

Figure 10. Probe size impact on IEEE 802.15.4 without RPL; normal background, $T = 80$ ms.

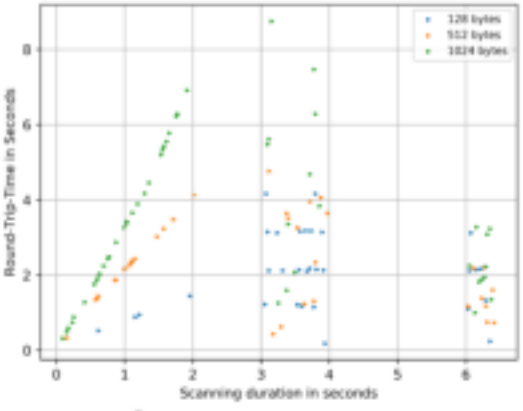

Figure 11. RTT for IEEE 802.15.4 without RPL; normal background, $T = 20$ ms.

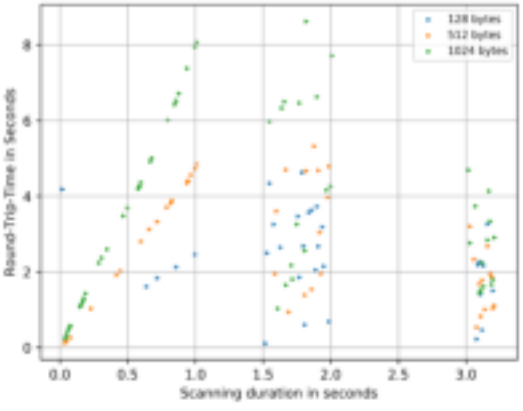

Figure 12. RTT for IEEE 802.15.4 without RPL; normal background, $T = 10$ ms.

Figure 17 and Figure 18 show the impact of faster probing. Many losses are present with fast probing, leading to less than 10% nodes found as shown in Figure 19. We need $T = 150$ ms to be able to find more than 90% of the nodes. Furthermore, Figure 13 and Figure 19 show the

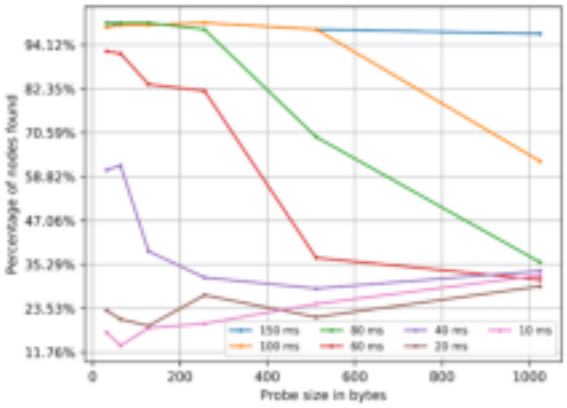

Figure 13. Number of IEEE 802.15.4 nodes found (without RPL); normal background.

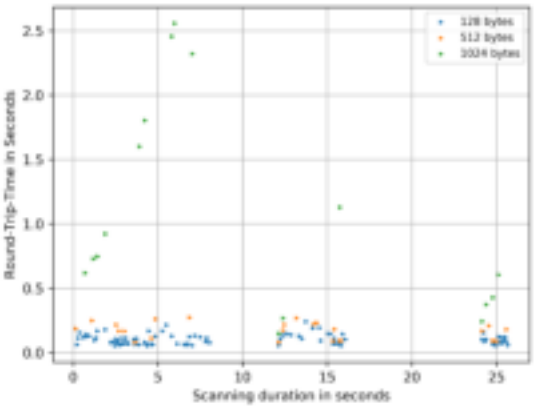

Figure 14. RTT for IEEE 802.15.4 with RPL and 1 hop; light background, $T = 80$ ms.

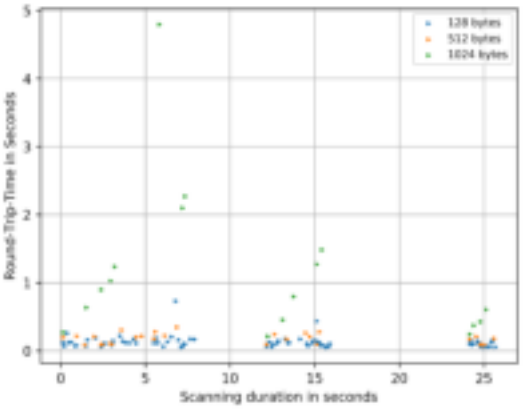

Figure 15. RTT for IEEE 802.15.4 with RPL and 1 hop; normal background, $T = 80$ ms.

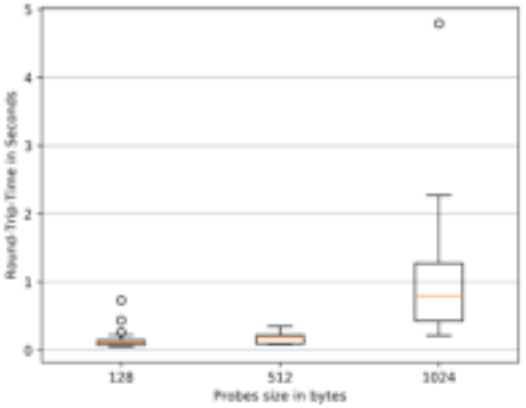

Figure 16. Probes size impact on IEEE 802.15.4 with RPL and 1 hop; normal background, $T = 80$ ms.

loss rate is higher than in the tests without RPL although both scenarios simulate a one-hop network. The reason for this are collisions with RPL control messages.

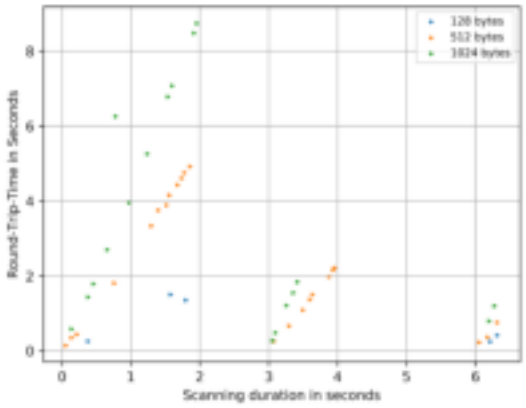

Figure 17. RTT for IEEE 802.15.4 with RPL and 1 hop; normal background, $T = 20$ ms.

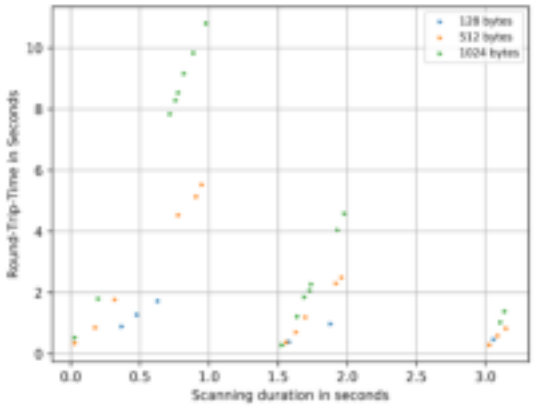

Figure 18. RTT for IEEE 802.15.4 with RPL and 1 hop; normal background, $T = 10$ ms.

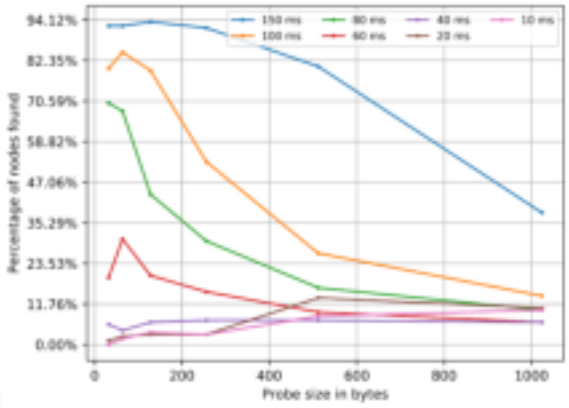

Figure 19. Number of IEEE 802.15.4 nodes found with RPL and 1 hop; normal background.

Figure 20 and Figure 21 show the results for light and normal background traffic and $T = 80$ ms in a two-hop IEEE 802.15.4 network with RPL. The box-plot in Figure 22 shows that the variation of the measured RTT has increased due to the additional hop. However, note that many large probes are lost in this scenario and therefore the results are not directly comparable. Figure 23 shows that we find only around 55% of the nodes when probing with 128-byte messages and $T = 80$ ms, compared to a discovery rate of nearly 100% when not using RPL. We need a very low scan speed ($T \geq 100$ ms) and small probes (≤ 100 bytes) to discover 90% of the nodes in the two-hop scenario.

6.3. 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 5, we simulate an IoT network with two sub-networks, containing 100 IEEE 802.15.4 nodes resp. 50 IEEE 802.11 nodes. For our first test, we do not use RPL and all IEEE 802.15.4 are close to the 6LBR.

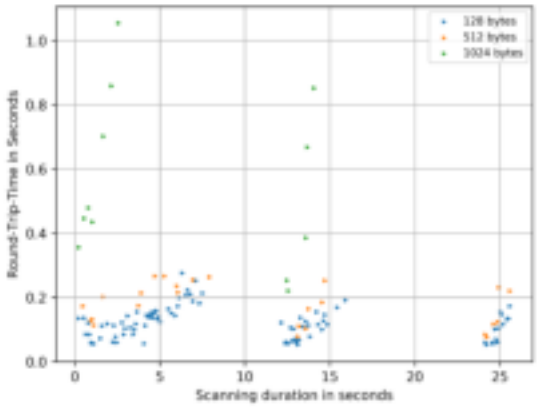

Figure 20. RTT for IEEE 802.15.4 with RPL and 2 hops; light background, $T = 80$ ms.

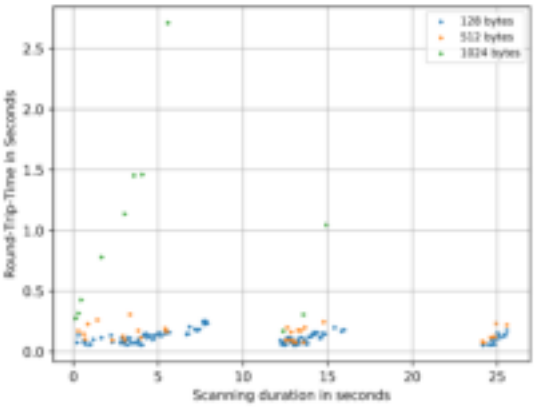

Figure 21. RTT for IEEE 802.15.4 with RPL and 2 hops; normal background, $T = 80$ ms.

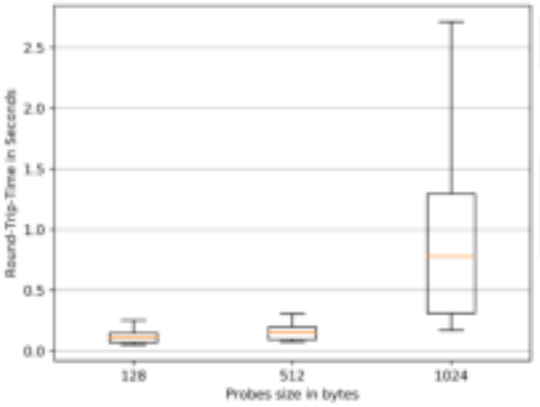

Figure 22. Probes size impact on IEEE 802.15.4 with RPL and 2 hops; normal background, $T = 80$ ms.

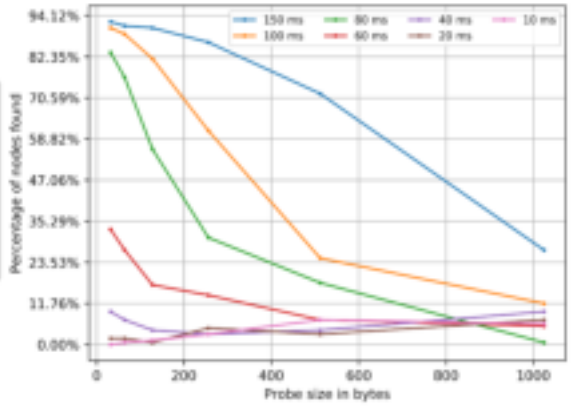

Figure 23. Number of IEEE 802.15.4 nodes found with RPL and 2 hops; normal background.

The first step of our approach is a fast scan of the entire area 2. Figure 24 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

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

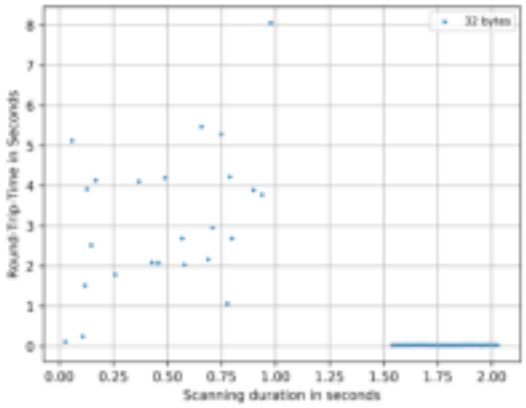

Figure 24. Mixed network; no RPL, normal background, $T = 10$ ms, 32-byte probes.

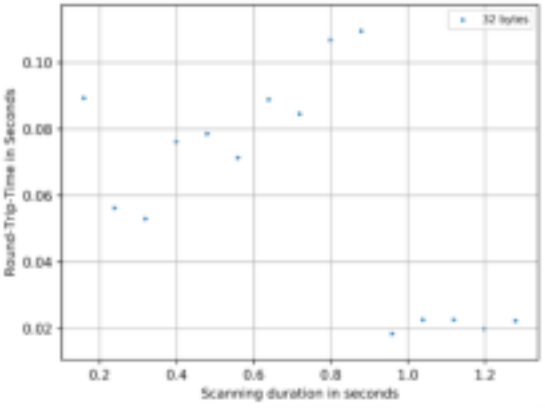

Figure 25. Selected nodes; no RPL, normal background, $T = 80$ ms, 32-byte probes.

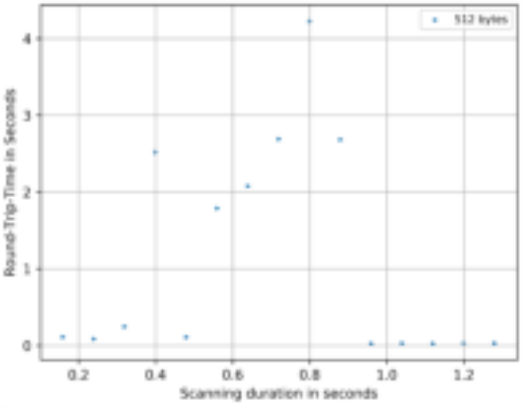

Figure 26. Selected nodes; no RPL, normal background, $T = 80$ ms, 512-byte probes.

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 step 2 of the approach is not really needed since Figure 24 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. Figure 25 and Figure 26 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.

Finally, in step 3 the part of the address range that contains IEEE 802.15.4 nodes is re-scanned at slow speed ($T = 80$ ms) and small probes (32-bytes). The results are shown in Figure 27. 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 Figure 13), however without having to scan slowly the *entire* IoT network. Our approach

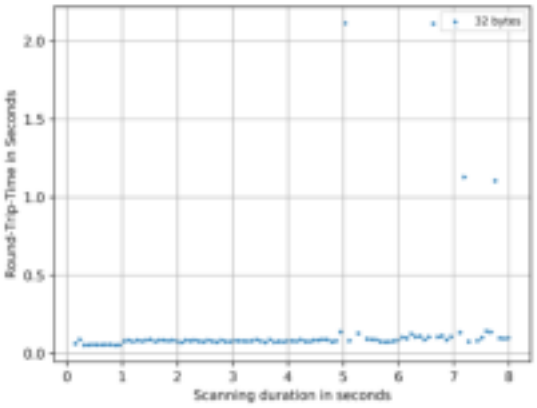

Figure 27. Mixed network; no RPL, normal background, $T = 80$ ms, 32-byte probes.

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 ($T = 80$ ms) of the entire IoT network.

We have omitted figures for the heterogeneous scenario with RPL since they are very similar to the ones already shown. In order to be able to achieve a discovery rate of significantly more than 90% in a two-hop IEEE 802.15.4 network, the scan speed in step 3 has to be decreased to $T = 150$ ms. The duration of step 3 increases by 7 seconds. Consequently, the total scan duration of our approach increases to 18.5 seconds, compared to 30 seconds for a slow scan of the entire IoT network with $T = 150$ ms.

7. 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 resource-constrained 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, including scenarios with wide-area communication based on LTE or LoRa or different short-range technologies. We are also working on other means to identify and fingerprint IoT nodes. Finally,

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.

we will develop a collaborative intrusion detection system to detect and mitigate scans and other attacks against IoT networks.

References

1. 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.
2. R. Barnett and B. Irwin. Towards a taxonomy of network scanning techniques. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, pages 1–7. ACM, 2008.
3. P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer communications*, 30(7):1655–1695, 2007.
4. R. Beverly. A robust classifier for passive tcp/ip fingerprinting. In *International Workshop on Passive and Active Network Measurement*, pages 158–167. Springer, 2004.
5. S. M. Bilalb, M. Othmana, et al. A performance comparison of network simulators for wireless networks. *arXiv preprint arXiv:1307.4129*, 2013.
6. D. Cid. Large cctv botnet leveraged in ddos attacks. https://blog.sucuri.net/2016/06/large-cctv-botnet-leveraged-ddos-attacks.html.
7. 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.
8. A. Dainotti, A. King, F. Papale, A. Pescape, et al. Analysis of a/0 stealth scan from a botnet. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 1–14. ACM, 2012.
9. S. E. Deering. **RFC 2460** internet protocol, version 6 (ipv6) specification, 1998.
10. Z. Durumeric, M. Bailey, and J. A. Halderman. An internet-wide view of internet-wide scanning. In *USENIX Security*, pages 65–78, 2014.
11. Z. Durumeric, E. Wustrow, and J. A. Halderman. Zmap: Fast internet-wide scanning and its security applications. In *Usenix Security*, volume 2013, 2013.
12. 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.
13. J. Frahim, C. Pignataro, J. Apcar, and M. Morrow. Securing the internet of things: A proposed framework. https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html. Accessed: 2017-03-31.
14. 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.
15. C. Gomez and J. Paradells. Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6), 2010.
16. F. Gont and T. Chown. **RFC 7707** network reconnaissance in ipv6 networks, 2016.
17. J. Hui and P. Thubert. **RFC 6282** compression format for ipv6 datagrams over ieee 802.15. 4-based networks, 2011.
18. A. Johnsson, B. Melander, and M. Björkman. Bandwidth measurement in wireless networks. In *Challenges in Ad Hoc Networking*, pages 89–98. Springer, 2006.
19. L. Metongnon, E. C. Ezin, and R. Sadre. Efficient probing of heterogeneous iot networks. In *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*, pages 1052–1058. IEEE, 2017.
20. G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. **RFC 4944** transmission of ipv6 packets over ieee 802.15. 4 networks, 2007.
21. Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow. Iotpot: analysing the rise of iot compromises. *EMU*, 9:1, 2015.
22. 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.
23. 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.

24. Z. Shelby and C. Bormann. *6LoWPAN: The wireless embedded Internet*, volume 43. John Wiley & Sons, 2011.
25. G. Taleck. Ambiguity resolution via passive os fingerprinting. In *International Workshop on Recent Advances in Intrusion Detection*, pages 192–206. Springer, 2003.
26. T. Winter. **RFC 6550** rpl: Ipv6 routing protocol for low-power and lossy networks, 2012.

This is a draft version of the paper published in <International Journal of Network Management, 2017>. Please cite the published version <http://onlinelibrary.wiley.com/doi/10.1002/nem.1997>.