2018-01-29

Multi-homed device detection using clock skew

Martin, Bryan; Tummala, Murali; McEachen, John

IEEE

Martin, Bryan, Murali Tummala, and John McEachen. "Multi-homed device detection using clock skew." Signal Processing and Communication Systems (ICSPCS), 2017 11th International Conference on. IEEE, 2017.
http://hdl.handle.net/10945/60891

Multi-Homed Device Detection Using Clock Skew

Bryan Martin
Department Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
bmartin@nps.edu

Murali Tummala
Department Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
mtummala@nps.edu

John McEachen
Department Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA
mceachen@nps.edu

***Abstract*— The aim of this paper was to determine the feasibility of identifying a device connected to the Internet through multiple interfaces (i.e., multi-homed) using the information provided by passively observing network traffic. Since multi-homed hosts allow an alternate means for outside entities to circumvent the security of a firewall and gain access to a network, it is important for a network's security to be able to detect and remove such devices. In this work, the idea of using clock skew—the difference in perceived time between two system clocks—as a unique signature is utilized to identify hosts on a network that are potentially multi-homed. Testing was done on a software-defined network that contained a multi-homed host. After traffic between hosts was collected and processed, analysis of the confidence intervals of the device's clock skew was conducted to determine if IP addresses originating from the same host could be successfully detected solely from network traffic. Results confirmed that the proposed scheme provided a valid means of detecting a multi-homed device on a network.**

Keywords— Software-defined network, Multi-homed host, Network monitoring, Network fingerprinting

I. Introduction

Network security remains a major concern for all communications systems. With the advent of network management techniques such as software-defined networking (SDN), the ability of a system administrator to leverage the monitoring functions of a panoptic controller have led to the development of a large range of applications for network control and security to include monitoring applications for maintaining the security and integrity of one's network [1].

A variety of security and cyber related concerns exist for any network. Before an attack can be conducted on a network, an attacker must first gain access. One method to prevent this is the use of a firewall between a private network and the Internet. A potential security flaw in a network is the existence of a multi-homed host [2]-[4]. Through the use of multiple interfaces on a host, the security of a network and the integrity of its firewall can be circumvented.

A multi-homed host is a device connected to the Internet through multiple interfaces [2]-[4]. If one of these connections is to a private network and the other to the open Internet, this provides a possible access vector that bypasses the network's firewall [4]. This threat calls for the need to be able to detect if a multi-homed host exists on a network and is the motivation behind this research.

The goal of this paper is to develop a scheme for detecting multi-homed hosts in a panoptic network such as a SDN. A framework for an application that can be used to detect hosts using multiple interfaces that are independent of their Internet Protocol (IP) or Media Access Control (MAC) address is provided in this paper. We investigate the use of the clock skew of a host compared to a designated fingerprinter as a unique identifier. If a unique clock skew correlates to two or more unique IP addresses on the network, this represents a possible multi-homed device. Analyses are conducted based on the confidence intervals of the calculated clock skews to determine if two similar clock skews represent the same, multi-homed host.

The idea for using the clock skew of a host for remote physical device fingerprinting was first suggested in [5]. It was shown that modern computer chips had detectable and distinguishable clock skews that could be calculated by observing the Transmission Control Protocol (TCP) timestamps from traffic on the network. It was then verified that the clock skew of a device remained constant even when using separate Ethernet and Wi-Fi interfaces originating from the same device [6]. This idea was further used as an enumeration tool in [7]. Researchers used clock skews of a device to determine the number of hosts active behind a network address terminal (NAT). This was accomplished by counting the number of unique clocks skews encountered from traffic exiting a NAT and correlating them to unique devices [7].

In this paper, these ideas are expanded upon and are used to detect multi-homed devices active on a SDN. We also conduct the confidence interval analysis of the clock skew data encountered on the network to identify devices that appear to be separate based on IP address but are originating from the same device. This paper is adapted from [8], previously published by the Naval Postgraduate School.

II. Background

Network security continues to be a vital concern for a constantly connected society. One such concern is the access afforded to a network via a multi-homed host [2]. Mitigating the threat on a SDN by detecting such a device is the focus of this research. Before proposing the detection scheme for such a device, the relevant background information is presented in this section to introduce the threats and tools that are used to mitigate them.

A. Multi-Homed Host

A multi-homed host is one that has multiple connections to a network or networks. This can be accomplished by having

multiple network interface cards (NICs) installed in the same host, which provides a host with multiple MAC and IP addresses [3]. Multi-homed hosts are used in a network for redundancy purposes [2]. With a multi-homed host on a network, the reliability of a network's access can be increased. Access node failure can be mitigated, and the connectivity from an Internet service provider (ISP) can be made more reliable by having separate connections to separate ISPs [9].

The threat from a multi-homed host comes from the fact that a multi-homed host can be used to bypass the firewall between an internal network and the Internet [2]. Certain operating systems, such as Windows, were never meant to isolate two interfaces within a host and often integrate traffic from one to the other [2]. This results in the ability for an infection on one network to be passed to another.

Closed networks are protected from the Internet by firewalls, which only allow designated traffic to flow between the two mediums. If a host is multi-homed, this allows for the opportunity to bypass the firewall and provide access to a closed network [4]. Once access to a host on a closed network is gained, potential threats can map a network and begin an exploitation process or infect the network with malicious code. An example of such a network configuration is depicted in Figure 1.

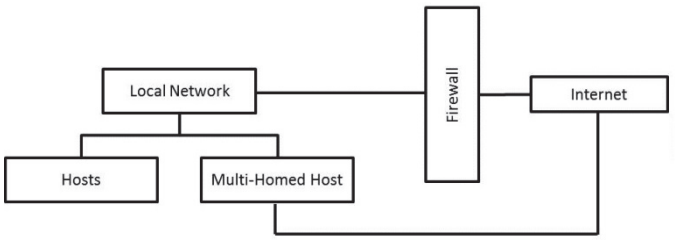

Fig. 1. A Network with a Multi-Homed Host Implemented to Bypass the Firewall to the Internet

A software-defined network is an innovative networking scheme in which the control and data planes within a network are logically separated. In a SDN, the routing functions for the network are controlled from a centralized location, known as the controller [1], [10], [11]. This centralized controller is able to view the operation of the entire network, allowing it to monitor and react to any potential hazards that may exist [11].

A SDN is divided into three planes that each interact to control the functionality of the network. The lowest plane is the data plane, which consists of switches that forward packets based on flow rules [1]. Above the data plane is the control plane. From here, network traffic is monitored, and the flow rules for designated packets are determined [1], [10], [12]. The controller can be programmed by applications, allowing the network to dynamically react to any changes within the network. This is done at the upper plane, known as the application plane of the network [1], [12].

Routing within a SDN is completed using flow rules that are determined at the control plane and stored at the data plane. Due to this functionality, routing is now a rule-based process vice a destination-based process [1]. A SDN operates as a Transmission Control Protocol/Internet Protocol (TCP/IP) network and uses the OpenFlow protocol for its rule-based routing. The OpenFlow protocol matches packets to designated flow rules within a flow table at the data plane. If no such rule exists, the packet is forwarded to the control plane where a decision is made as to how it should be routed. Once this determination is made, the packet is forwarded back to the data plane for routing along with updates for the flow tables for future routing decisions [11], [12].

B. System Clocks

Networked devices all have internal electric clocks that are built from both hardware and software components. These clocks control all timing functions for the device [13]. Within these electronic clocks, crystal oscillators are used to determine the clock signal and the rate at which the clock ticks [14]. These crystal oscillators each operate at unique frequencies due to the crystal type, the manufacturing parameters, and the small imperfections that are inherent to all manufacturing procedures [14], [15]. Due to these factors, clocks within a device operate at slightly different frequencies independent of clock type or manufacturing series [14]. This makes the system clock within a device a unique characteristic that can be exploited to identify that device.

The TCP header consists of a standard 20 bytes of information followed by a portion of data allocated to options within the protocol [16]. In the options section of the header of a TCP packet is a field for the TCP timestamp. The TCP timestamp is a one-up counter based off a device's system clock that was introduced in RFC 1323 as a means of accurately measuring the round-trip time (RTT) between two devices. The need for accurately measuring the RTT of a packet is to provide a basis for determining the retransmission timeout interval (RTO) for lost or unacknowledged packets [17].

The TCP timestamp value is determined by a virtual "timestamp clock" that is based on the frequency of operation of the device's system clock. By observing the values of TCP timestamps, one can observe the operation of the system clock [5]. The TCP timestamp is a second-order effect of the system clock and is the means in which the clock skew is calculated in this research.

The clock skew of a device is the difference in the operating frequencies of its system clock relative to the clock frequency of another device [5]. It is this parameter that can be used to identify the device based solely through passively observing network traffic. When using clock skew as a unique identifier, the identifier is valid only in relation to a designated device. This device is known as the fingerprinter [5]. In a SDN, the controller can be designated as the fingerprinter due to its ability to monitor all devices connected to the network.

III. MULTI-HOMED DEVICE DETECTION SCHEME USING CLOCK SKEW

In order to detect a host on a network using multiple connections through multiple NICs, we must determine whether or not the traffic between different IP addresses can be correlated in order to determine if those IP addresses belong to

a multi-homed host. Two assumptions are made in developing the proposed scheme. The first is that passive means of collection are used over the network. The second is that the observer can observe and collect traffic from all IP addresses of a multi-homed host.

A. Proposed Scheme

The proposed solution is to collect TCP timestamp data from a host in order to calculate its clock skew for use as a fingerprint. The clock skew of a host is unique and has very little variation over time. It has been demonstrated that the clock skew of a host stays relatively constant even if two interfaces (Ethernet and Wi-Fi) are used to connect to a network. For these reasons, the clock skew can be used as an identifier for a given host [5], [6], [18].

The first step in the proposed process is to monitor and collect traffic across the network. The traffic of interest is the TCP segments exchanged between hosts, specifically those containing TCP timestamps. From this information, the clock skew of each host relative to a central host (the fingerprinter) can be calculated. After the clock skews of each host on the network are determined, analysis is conducted based on hypothesis testing using confidence intervals to identify potential multi-homed hosts. A testbed network with a fingerprinter is shown in Figure 2, and the process for detecting a multi-homed host is outlined in Figure 3.

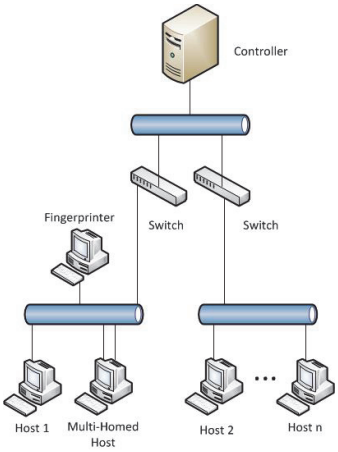

Fig. 2. Generic Network Configuration of a SDN with a Controller, Two Switches, and n Number of Hosts with One Acting as the Fingerprinter for Testing and Another Multi-Homed

In previous work, this method was utilized for the determination of the number of hosts behind a NAT. It was suggested in [5] and shown in [7] that one could determine the number of hosts sending traffic through a NAT by calculating and comparing the unique clock skews encountered. In this paper, we propose the use of correlating clock skews between multiple IP addresses to determine if they are originating from the same, multi-homed device.

B. Network Confifuration

To test our proposed scheme, we collect and analyze traffic from hosts on a network. A version of the network layout is shown in Figure 4. Multiple hosts are connected to each switch with one host among them being multi-homed. The multi-homed host uses separate Ethernet connections to connect to the network. A central host acts as the fingerprinter for determining the clock skews of all hosts on the network [5]. The fingerprinter is chosen so that it has the ability to observe traffic from both connections of the multi-homed host.

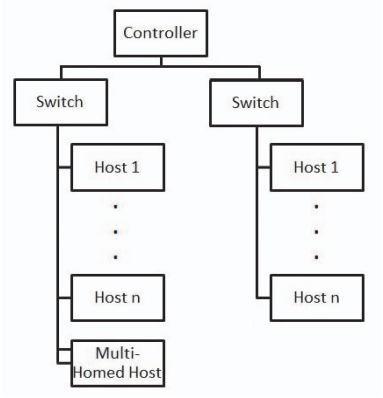

Fig. 4. Configuration of a SDN with a Multi-Homed Host Connected to a Single Switch

C. Clock Skew

In order to test the proposed scheme, network traffic containing TCP segments with timestamps was collected. Using this data, the clock skew of each host can be calculated.

The fingerprinter monitors the network from a centralized location for TCP segments in the network traffic. Not all of these TCP segments seen in the network traffic will contain timestamps. It is the segments with TCP timestamps originating from a host and sent to the fingerprinter that are of interest. These segments are aggregated, and the TCP timestamps are used in the calculation of the clock skew. These TCP timestamps are collected along with the time of collection based on the fingerprinter's own clock.

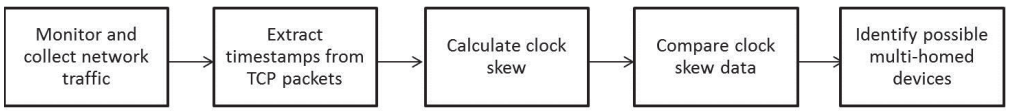

Fig. 3. Process of Detecting Multi-Homed Devices Using Clock Skew

TCP timestamps were introduced as a means to provide a simple and accurate tool to measure the RTT of a packet transmission [17]. TCP is meant to be a reliable connection-oriented protocol, and this reliable connection is achieved by the retransmission of lost or dropped packets. The duration of time before retransmissions are sent is known as the RTO and is calculated by knowing the RTT of a packet. TCP timestamps, included in the TCP options portion of the header and consists of 10 bytes of data, provide a simple and accurate means of determining this RTT by sending and echoing relative timing information within the TCP packet [17].

The value of the timestamp comes from a virtual internal clock that is known as the "timestamp clock" and is based upon the device's own clock [17]. TCP timestamps are a second-order effect of the host's system clock, and through their collection and measurement, the operation of a host's system clock can be observed [5].

The clock skew is a physical trait of a host's processor caused by the different operating frequencies of crystal oscillators within electronic clocks. The discrepancy in operating frequencies is a product of the manufacturing process and results in small differences in clock speed of each clock [14], [18]. This difference in frequencies between the system clocks of separate devices is calculated as the first derivative of a function that includes the offset of their observed times [5], [6], [18].

Once the TCP timestamps have been collected, the clock skew can be calculated based on the procedure provided in [5]. The first step is to determine the time and TCP timestamp offsets of a collected packet versus the initial time of collection. The first packet collected by the fingerprinter from a host is used as the baseline for the offset. The time offset is given by [5]

$$x_i = t_i - t_1 \qquad (1)$$

where x_i is the difference between the time of collection of the ith packet at time t_i and the initial time of collection t_1. The timestamp offset w_i for the ith packet is given by

$$w_i = \frac{T_i - T_1}{f} \qquad (2)$$

where T_i is the timestamp of the ith packet, T_1 is the timestamp of the first packet at the initial time of collection and f is the operating frequency of the host's clock.

Once the time and timestamp offsets are known, the difference y_i between the observed time at the fingerprinter and the observed time from the source host based on its timestamps is calculated as

$$y_i = w_i - x_i \ . \qquad (3)$$

Given the set of points x and y for the data collected, the set of offset values O_T for N collected packets is represented as

$$O_T = \{(x_i, y_i) : i \in \{1,...,N\}\} \ , \qquad (4)$$

and we model the data as a slope-intercept line equation.

The clock skew is the first derivative (or slope) α of this line

$$\alpha \cdot x_i + \beta \geq y_i \qquad (5)$$

with a y-intercept of β that fits the upper bound of the set of points O_T. The solution to (8) is obtained using a linear programming technique with the goal to minimize the objective function J

$$J = \frac{1}{N}\sum_{i=1}^{N}(\alpha \cdot x_i + \beta - y_i) \qquad (6)$$

for N packets [5]. This procedure is repeated for each host on the network.

D. Dectection of Multi-Homed Hosts

Once the clock skews have been calculated, a comparison must be made in order to determine which IP addresses represent the potential multi-homed host in the network. To improve accuracy, a large number of trials are required. Based on the central limit theorem, the sample mean of independent random variables approaches a Gaussian distribution [19]. Consequently, given a relatively large number of trials, we assume that the clock skews calculated for each host over these trials approaches a Gaussian distribution.

After the mean clock skew is determined for a host, analysis is done using the confidence intervals for the clock skew of all hosts and hypothesis testing to determine whether the IP addresses belong to a multi-homed host.

The sample mean m_i of the clock skew for the ith host is determined as

$$m_i = \frac{1}{N}\sum_{i=1}^{N}\alpha_i \qquad (7)$$

where α_i is the calculated clock skew for the ith host [20]. Now we formulate the following hypotheses. The first hypothesis H0 states that m_j for the jth host's clock skew is within the range of the ith host's confidence interval. The second hypothesis H1 states that m_j for the jth host's clock skew is

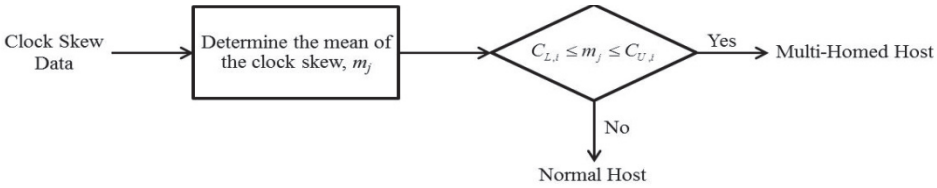

Fig. 5. Process of Testing Hypotheses Using Confidence Intervals to Determine If Hosts Are Multi-HomeResults

outside of this range. The lower bound of the confidence interval for a host i is represented by $C_{L,i}$ and the upper bound is represented by the value $C_{U,i}$. If m_j falls within the confidence interval

$$C_{L,i} \le m_j \le C_{U,i} \tag{8}$$

when $i \neq j$, then hypothesis H0 is accepted and the IP addresses are flagged as originating from the same host. If not, then hypothesis H1 is accepted and the IP addresses did not originate from the same host [19]. The process for this analysis is shown in Figure 5.

IV. Results

A scheme for calculating clock skew based on TCP traffic was proposed in Section III. This scheme was validated using a SDN test bed for data collection. The configuration of the network used and the means for generating and capturing the test traffic is described in this section. We then calculate the clock skew of each host and apply the confidence interval analysis on the clock skew of each host to identify the multi-homed host.

A. Network Configuration and Traffic Collection

A portion of the SDN test bed that was built for testing in [21] was used in this experiment and consisted of two HP switches and seven Raspberry Pis as hosts. The switches used were the HP 2920 and the HP 3800, and the Raspberry Pis were connected to the network using their built-in 10/100 Mbps Ethernet connection. The network configuration that was used is shown in Figure 6.

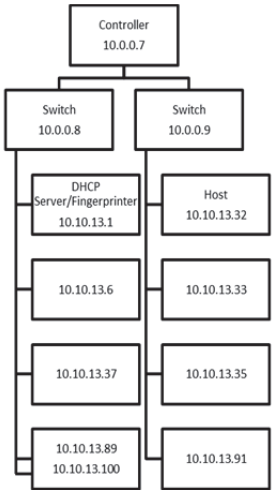

Fig. 6. Network Configuration Used in Testing

One of the Raspberry Pis had an added USB 2.0 Gigabit LAN adapter that was used as its second connection to the network. This was the dual-homed device used in testing and the host that was to be experimentally identified. This host used the IP addresses 10.10.13.89 and 10.10.13.100. Both connections from this host were connected to the HP 2920 switch.

Also connected to the network was a Dell T1600 running Ubuntu that was acting as the DHCP server for the network. The DHCP server was used as the fingerprinter in this experiment and was chosen due to the fact that it maintained a static IP address of 10.10.13.1 throughout testing.

In order to establish the necessary TCP connections for the purpose of creating TCP timestamps, traffic was generated by creating an Secure Shell (SSH) connection between the fingerprinter and the hosts on the network. This SSH connection allowed for the required TCP handshakes to be made and timestamps to be exchanged between the host and the fingerprinter for collection. Packets with TCP timestamps that were originating from a host were collected using Wireshark.

B. Clock Skew Calculation and Results

Given the test traffic collected by Wireshark, the next step was to calculate the clock skew of each host. One hundred samples of data were collected at ten minute intervals, and MATLAB was used for calculations. Using the MATLAB function *linprog*, we solved (6) from Section III for each host. The solution provided the values of α and β, which are the slope and y-intercept of the solution to (5). The value of α corresponds to the clock skew and is the value of concern in this scenario.

The clock skew for each host was calculated independently for each trial and the upper-bound solution, which was used because the delays found within a network between hosts are all positive, for the set of points O_T was solved for each host [5]. As shown in Figure 7, the solution for the set of data points in red corresponding to host 10.10.13.100 provides a slope of 0.0000101203 or 10.1203 ppm for the line in blue representing the upper bound of the data set. This slope is the clock skew for this host when compared to the clock of the fingerprinter, 10.10.13.1.

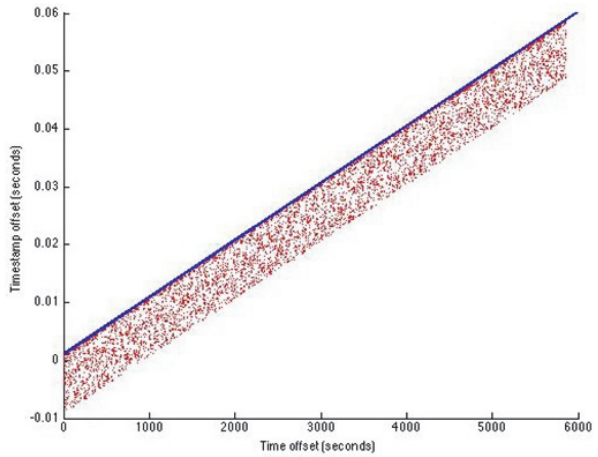

Fig. 7. Upper-Bound Solution for Host 10.10.13.100 over a Single Trial

Comparing the slopes for the upper-bound solution of the data sets of all hosts over a single trial shows the variation of the clock skews found in this network. As seen in Figure 8, there is a range of positive and negative values for the clock skew corresponding to a host's clock being ahead of or behind the clock of the fingerprinter. The hosts using the IP addresses of 10.10.13.89 and 10.10.13.100 both have solutions with similar slopes and stand out as possibly being multi-homed due to the fact that the solution to (5) for each host appears to be represented by two parallel lines.

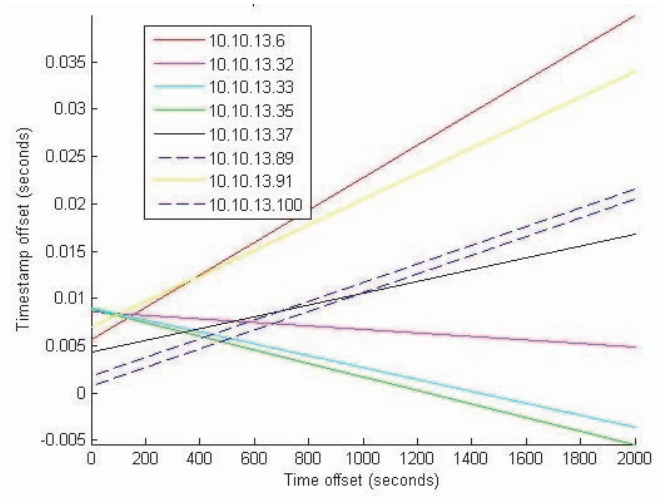

Fig. 8. Upper Bound Solution of All Hosts over a Single Trial

The data in Figure 8 is supported by further trials. The mean value for each clock skew after 100 trials is depicted in Table 1. This data shows that the clock skews for 10.10.13.89 and 10.10.13.100 are similar. When compared to the differences between clock skews of the other hosts tested, as shown in Table 2, the difference between 10.10.13.89 and 10.10.13.100 appears to be negligible.

TABLE I. MEAN CLOCK SKEW OF ALL HOSTS OVER 100 TRIALS (IN PPM)

Host	Clock Skew (ppm)
10.10.13.6	17.126
10.10.13.32	-1.953
10.10.13.33	-6.405
10.10.13.35	-7.313
10.10.13.37	6.700
10.10.13.89	10.132
10.10.13.91	13.020
10.10.13.100	10.140

For these comparisons and for the calculation of the confidence intervals, the data was assumed to approach a Gaussian distribution after the 100 trials. As shown in Figure 9, the range of clock skews collected for host 10.10.13.6 over these trials approaches a normal distribution.

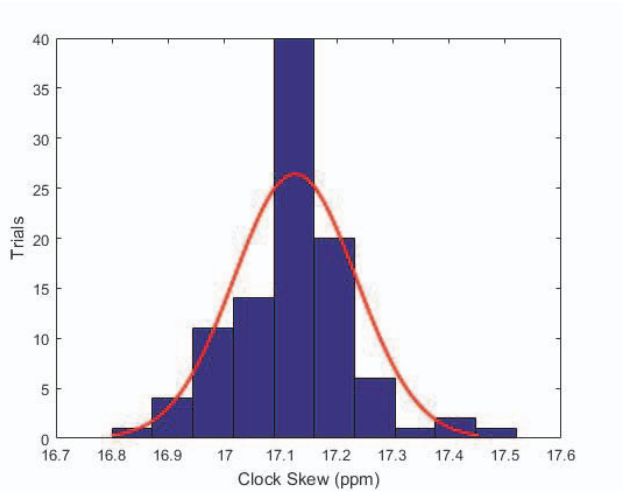

Fig. 9. Histogram for the Calculated Clock Skews of Host 10.10.13.6 Over 100 Trials as they Approach a Gaussian Distribution

A 95% confidence interval for the clock skew of each host was calculated over the 100 trials conducted. The confidence interval was solved using the *paramci* function within MATLAB. The results for the confidence intervals are shown in Figure 10. In Figure 10 the value of the clock skew for each host is shown as a bar graph in blue. The error bar in red covers the range of values from the lower to the upper bounds of the confidence interval. The confidence interval for each clock skew is quite small, which suggests that the clock skew varies only slightly over time; this result has been observed in previous work [5], [6].

TABLE II. DIFFERENCE OF CLOCK SKEW BETWEEN ALL HOSTS (IN PPM)

Host	10.10.13.6	10.10.13.32	10.10.13.33.	10.10.13.35	10.10.13.37	10.10.13.89	10.10.13.91	10.10.13.100
10.10.13.6	0.000	19.078	23.531	24.439	10.426	6.994	4.106	6.986
10.10.13.32	19.078	0.000	4.453	5.360	8.653	12.084	14.972	12.092
10.10.13.33	23.531	4.453	0.000	0.908	13.105	16.537	19.425	16.545
10.10.13.35	24.439	5.360	0.908	0.000	14.013	17.445	20.332	17.452
10.10.13.37	10.426	8.653	13.105	14.013	0.000	3.432	6.320	3.440
10.10.13.89	6.994	12.084	16.537	17.445	3.432	0.000	2.888	0.008
10.10.13.91	4.106	14.972	19.425	20.332	6.320	2.888	0.000	2.880
10.10.13.100	6.986	12.092	16.545	17.452	3.440	0.008	2.880	0.000

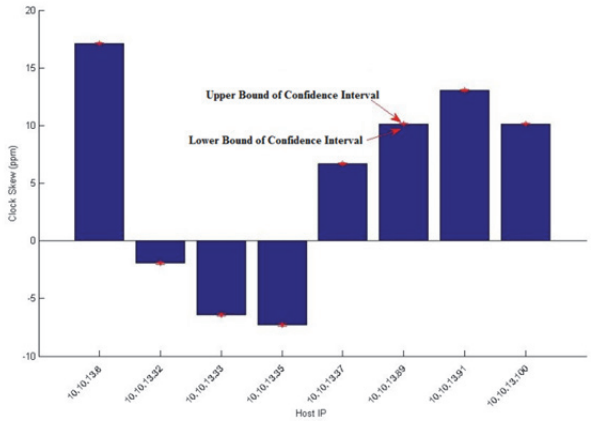

Fig. 10. Confidence Interval of 95% for the Clock Skew of All Hosts over 100 Trials

C. *Dectection of the Dual-Homed Host*

As described in Section III, analysis of the confidence intervals of the clock skew for each host was used to determine which hosts were possibly multi-homed. Using the confidence intervals as presented in Figure 10, we applied the ideas presented in Section III to the given data. When the mean clock skew of each host is compared to the confidence interval calculated for all other hosts, the possible dual-homed host can be identified. The upper and lower bounds for the confidence interval for the clock skews of all hosts are shown in Table 3 along with the mean value of the clock skews calculated over 100 trials.

When the mean value of each calculated clock skew is compared to the confidence interval of the clock skew for each host, it is observed that the possible dual-homed hosts are 10.10.13.89 and 10.10.13.100. The confidence intervals for hosts 10.10.13.6 and 10.10.13.89 are shown in Figure11 and Figure 12; the confidence intervals of the designated hosts are in blue while the values for clock skew for all hosts on the network in red. As can be seen in Figure 11, the confidence interval for host 10.10.13.6 only includes the value of its own clock skew. In Figure 12, the hosts represented by the IP addresses of 10.10.13.89 and 10.10.13.100 fall within confidence interval the confidence interval for 10.10.13.89, while the other hosts remain outside of these bounds. After comparing the data in Table 3 to Figure 11 and Figure 12, these results confirm the initial network setup where the hosts represented by the IP addresses 10.10.13.89 and 10.10.13.100 were from the same host.

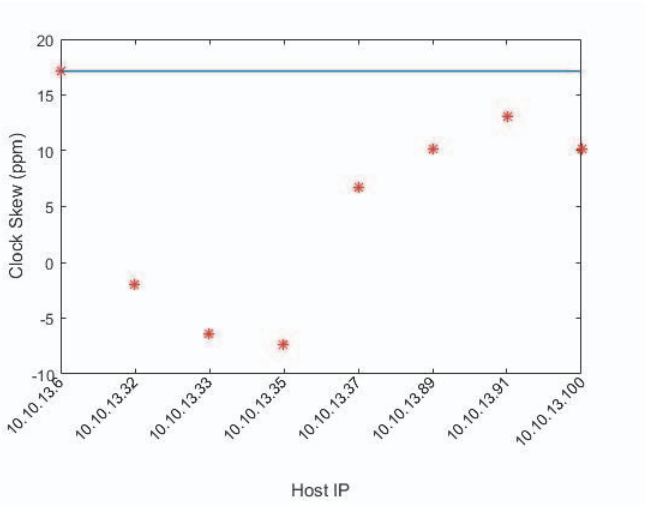

Fig. 11. Confidence Interval of 10.10.13.6 Compared to the Mean Value of All Clock Skews Calculated

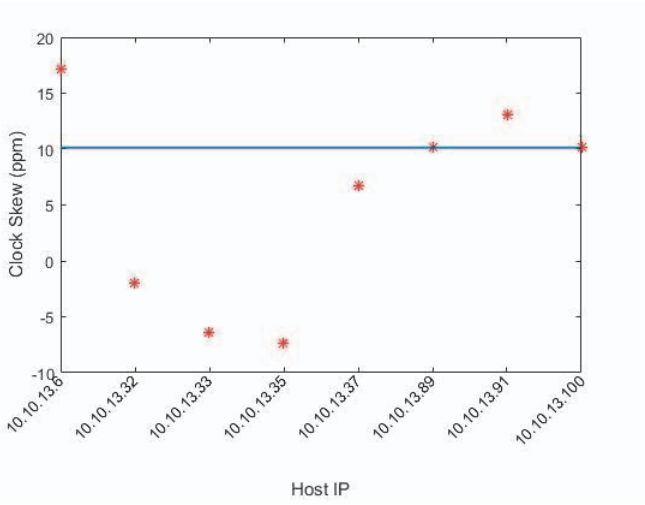

Fig. 12. Confidence Interval of 10.10.13.89 Compared to the Mean Value of All Clock Skews Calculated

TABLE III. UPPER AND LOWER BOUNDS OF THE 95% CONFIDENCE INTERVAL OF EACH HOST'S CLOCK SKEW

	Host							
	10.10.13.6	10.10.13.32	10.10.13.33	10.10.13.35	10.10.13.37	10.10.13.89	10.10.13.91	10.10.13.100
Upper Bound CI	17.147	-1.860	-6.276	-7.184	6.757	10.171	13.085	10.176
Mean Value	17.126	-1.953	-6.405	-7.313	6.700	10.132	13.020	10.140
Lower Bound CI	17.104	-2.045	-6.534	-7.441	6.643	10.093	12.955	10.104

D. Multi-Homed Host

The final validation of the proposed scheme was to add a host with three interfaces to the network and attempt its detection. A Raspberry Pi was connected to the network using its standard built in Ethernet connection as well as with two USB to Ethernet adapters. These interfaces were assigned with the IP addresses of 10.10.13.89, 10.10.13.91, and 10.10.13.100. As in the previous sections, the clock skew for all hosts on the network were calculated, and the proposed scheme was used to correlate any possible multi-home connections. As seen in Figure 13, there are now three parallel lines for the solutions to (5), suggesting that these IP addresses are from the multi-homed host.

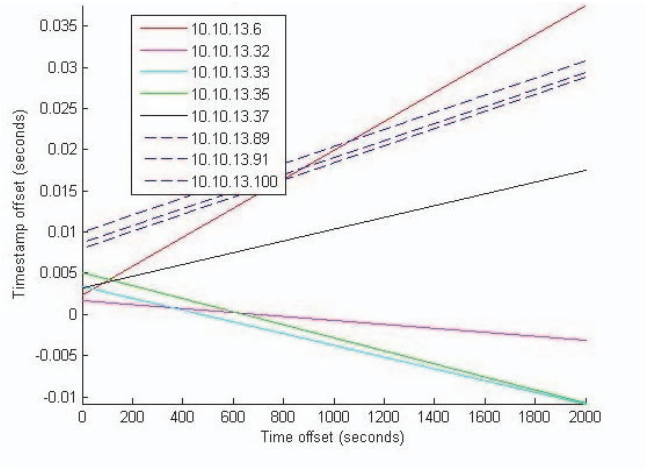

Fig. 13. Upper Bound Solution of All Hosts over a Single Trial after the Three Connections Were Made to the Network from One Host

This is confirmed when their mean values are compared to each other's confidence intervals as was done in previous sections.

V. CONCLUSION

A scheme to use the clock skew of a device as an identifier that is independent of the interface the device used to connect to the network was developed and tested in this work. Since the clock skew of a host stays relatively constant over time [5] and is independent of the interface used [6], it was proposed that this can be used to correlate traffic that appears to be coming from different source IP addresses as traffic from the same host.

The concept of using clock skew as a unique identifier for a host has been suggested and tested in literature, but this idea has not been utilized in attempting to detect a host on a network using multiple interfaces. These concepts and methods were used to create a model to detect a multi-homed host from a designated fingerprinter. This information can then be used by the controller in a SDN to create new flow rules and isolate a possible multi-homed host for further investigation and to mitigate security risks. The ability for a designated host to act as a fingerprinter and determine the clock skews of each host on its subnet based on information from its own internal clock and TCP timestamp information was demonstrated in this research. Based on this information, it was shown using analyses of the confidence intervals of a device's clock skew compared to the calculated mean clock skew of all other devices on the network that the traffic from IP addresses that originated from the same host can be correlated to one another. Finally, it was shown that it was possible to use this scheme to detect a device on the network using three distinct interfaces.

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," Proceedings of the IEEE, vol. 103, pp. 14–76, 2015.

[2] E. Byres, "Dual homed machines are the juiciest targets," Tofino Security, 2010. https://www.tofinosecurity.com/blog/dual-homed-machines-are-juiciest-targets

[3] H. Bigdoli, Handbook of Information: Security, Threats, Vulnerabilities, Prevention, Detection and Management. Hoboken, NJ: John Wiley and Sons, Inc., 2006.

[4] T. J. Klevinsky, S. Laliberte, and A. Gupta, Hack I.T.: Security through Penetration Testing. Boston, MA: Pearson Education, Inc., 2002.

[5] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," IEEE Transactions on Dependable and Secure Computing, vol. 2, pp. 93–108, 2005.

[6] L. Polcak, J. Jirasek, and P. Matousek, "Comment on Remote Physical Device Fingerprinting," IEEE Transactions on Dependable and Secure Computing, vol. 11, pp. 494–496, 2014.

[7] G. Wicherski, F. Weingarten, and U. Meyer, "IP agnostic real-time traffic filtering and host identification using TCP timestamps," in IEEE 38th Conference on Local Computer Networks, 2013, pp. 647–654

[8] B. J. Martin, "Detecting a Multi-Homed Device Using Clock Skew," M.S. thesis, Electrical and Computer Engineering, Monterey, CA: Naval Postgraduate School, 2016.

[9] W. Jianping V. M. Vokkarane, R. Jothi, Xiangtong Qi, B. Raghavachari, and J. P. Jue, "Dual-homing protection in IP-over-WDM networks," Journal of Lightwave Technology, vol. 23, pp. 3111–3124, 2005.

[10] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," IEEE Communications, vol. 51, pp. 36–43, 2013.

[11] [K. Bakshi, "Considerations for software defined networking (SDN): Approaches and use cases," in IEEE Aerospace Conference, 2013, pp. 1–9.

[12] J. L. Johnson, "Software Defined Network Monitoring Scheme Using Spectral Graph Theory and Phantom Nodes," M.S. thesis, Electrical and

Computer Engineering, Monterey, CA: Naval Postgraduate School, 2014.

[13] S. Zander and S. J. Murdoch, “An improved clockskew measurement technique for revealing hidden services.” In USENIX Security Symposium, 2008, pp. 211–226.

[14] F. Lanze, A. Panchenko, B. Braatz, and A. Zinnen, “Clock skew based remote device fingerprinting demystified,” in IEEE Global Communications Conference, 2012, pp. 813–819.

[15] D. L. Mills, “Network Time Protocol (Version 3): Specification, Implementation, and Analysis,” RFC 1305, 1992.

[16] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, http://www.rfc-editor.org/info/rfc793.

[17] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", RFC 1323, DOI 10.17487/RFC1323, May 1992, http://www.rfc-editor.org/info/rfc1323.

[18] H. Kikuchi, Y. Tominaga, and Y. Tanaka, “Remote host fingerprinting based on clock skew,” in International Symposium on Communications and Information Technologies, 2008, pp. 225–227.

[19] C. W. Therrien, Discrete Random Signals and Statistical Signal Processing. New York: Prentice Hall, 1992.

[20] M. Tummala and C. Therrien, Probability and Random Processes for Electrical and Computer Engineers. Boca Raton, FL: CRC Press, 2012.

[21] T. C. Parker, “Spectral graph theory analysis of software-defined networks to improve performance and security,” Ph.D. dissertation, Electrical and Computer Engineering, Monterey, CA: Naval Postgraduate School, 2015.