A Graph-Theoretic Approach to Virtual Access Point Correlation

John D. Roth*, Jeremy Martin†, and Travis Mayberry†
*Naval Postgraduate School, Email: jdroth@nps.edu †United States Naval Academy, Email: {jmartin, mayberry}@usna.edu

***Abstract*—The wireless boundaries of networks are becoming increasingly important from a security standpoint as the proliferation of 802.11 WiFi technology increases. Concurrently, the complexity of 802.11 access point implementation is rapidly outpacing the standardization process. The result is that nascent wireless functionality management is left up to the individual provider's implementation, which creates new vulnerabilities in wireless networks. One such functional improvement to 802.11 is the virtual access point (VAP), a method of broadcasting logically separate networks from the same physical equipment. Network reconnaissance benefits from VAP identification, not only because network topology is a primary aim of such reconnaissance, but because the knowledge that a secure network and an insecure network are both being broadcast from the same physical equipment is tactically relevant information. In this work, we present a novel graph-theoretic approach to VAP identification which leverages a body of research concerned with establishing community structure. We apply our approach to both synthetic data and a large corpus of real-world data to demonstrate its efficacy. In most real-world cases, near-perfect blind identification is possible highlighting the effectiveness of our proposed VAP identification algorithm.**

I. Introduction

A common requirement when designing enterprise wireless networks is to support authenticated users in parallel with unauthenticated guest users. This often takes the form of two separate networks with separate Service Set IDentifiers (SSIDs) (i.e., for instance *MyNetwork* and *MyNetwork_Guest*). This has been accomplished in the past by putting multiple Access Points (APs) together, each one hosting a separate SSID. Motivated by a desire to eliminate redundant hardware and reduce deployment costs, a feature known as Virtual Access Points (VAPs) is often included in modern enterprise APs (and sometimes consumer-grade APs as well).

Traditionally, a wireless AP hosts a single wireless network configured with specific settings, including channel, SSID, and encryption. However, APs with VAP functionality can host multiple wireless networks, each with its own SSID and security settings.

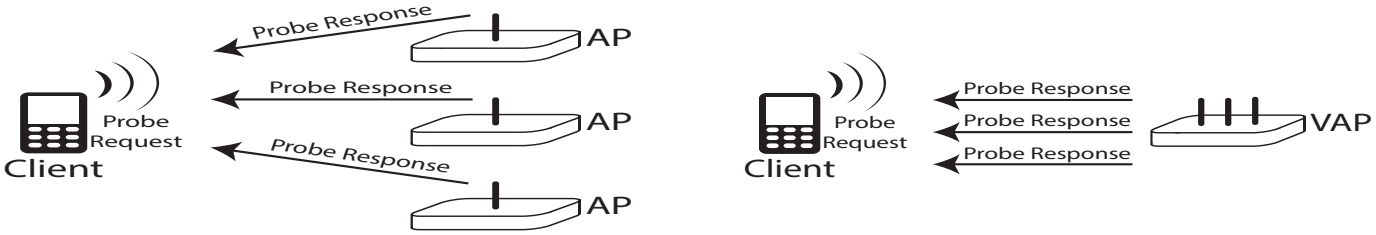
Since multiple VAPs run on a single physical AP, the least secure network being offered by an AP may be used by an attacker as an attack vector to gain access to the hardware. After connecting to an unsecured network, it may be possible to take control of the device or escalate access to one of the secured networks. It has been reported that many APs allow access to the internal configuration page from guest networks [1]. This access could be combined with poor AP configuration, default administrator passwords, or local software vulnerabilities to gain control of the AP [2].

Because of this useful added attack vector, an interesting research question arises: is there a systematic way to determine if two VAPs with different SSIDs and Basic Service Set IDentifiers (BSSIDs) are being hosted by the same physical AP? Although some networks are configured with straightforward naming conventions (i.e., *MyNetwork* and *MyNetwork_Guest*), this is not always the case. Moreover, there are often more than two VAPs hosted on a single device, and sometimes even VAPs with hidden SSIDs, further complicating the VAP ecosystem.

Related work in the literature is largely geared towards using VAP functionality to improve user experience in 802.11 networks. Specifically, work has been done to manage user mobility in WiFi networks through usage of VAPs [3], [4]. Additionally, indoor localization is another genre of research that has specifically addressed VAPs [5], [6]. Due to multipath artifacts during collection researchers have largely settled on fingerprint methods as the most viable solution for indoor localization. Fingerprint methods utilize a radio-frequency (RF) signature recorded at various locations in the area of interest. However, because the VAPs all transmit from the same physical hardware their resulting signatures are nearly identical, and thus redundant. Indoor localization methods seek to identify these redundancies and remove them in order to save in computational cost. The authors in [5] did this by defining a graph of which the vertices represented BSSIDs. Edges, or similarities, were defined by a correlation coefficient calculated from signal strength. A clique analysis was then performed to identify potential sources of redundant information (i.e., VAPs).

While the authors in [5] were more tolerant of false positive results, we seek greater precision to aid in a more surgical approach to network reconnaissance and attack vector identification. To this end, we explore methods of defining similarity other than signal strength by exploiting unencrypted attributes in 802.11 available to a passive listener. Specifically, in this paper, we use data from broad-scale 802.11 collection to:

- develop a novel multi-dimensional graph-theoretic framework for VAP correlation,
- develop a new voting map f to project the multi-dimensional representation into a space suitable for VAP correlation,
- demonstrate the effectiveness of this method on both synthetic and real-world large data sets, and
- conduct an extensive survey of 802.11 parameters over 600 GBs of real-world data that motivates our approach and illuminates current trends in VAP deployment.



(a) Traditional AP probe request/response operation

(b) VAP-enabled probe request/response operation

Fig. 1

First we identify several features of wireless traffic between APs and clients that can be used to indicate that two or more VAPs are colocated on the same physical device. Then, we represent traffic between clients and APs as an undirected graph, with the weights of each edge being determined by these features. After applying standard graph partitioning techniques, this ultimately leaves us with connected subgraphs that represent VAPs.

We go on to experimentally validate our approach against both simulated data and real-world WiFi traffic. When tested on these data, we achieve precision of .99 and recall of 1.0, showing that our approach is both accurate and robust.

II. 802.11 Access Ecology

A. AP Discovery

The IEEE 802.11 standard affords client devices the ability to discover nearby APs using active or passive scanning techniques [7]. The simplest form of discovery is passive scanning, where a device rotates through each 802.11 channel waiting to receive beacon frames from nearby APs, which are broadcast at regular intervals. Upon receiving a beacon frame, the device will update its available networks list.

Instead of just waiting for the beacons, active scanning can be done by transmitting probe request frames as the client rotates through the available channels. As the name suggests, these frames request nearby APs to respond and notify the device that they are within range. A probe request frame may be sent as a directed probe, targeting a single AP, or a broadcast targeting all nearby APs. A broadcast probe contains an empty SSID attribute (known as a *wildcard* SSID), while a directed probe has a specific SSID or *name* of an AP. APs in range receiving the probe will reply to the probe request with a probe response frame as depicted in Figure 1.

The use of directed probe requests, has over time been purposely limited due to growing privacy and tracking concerns [8], [9]. Directed probes are typically only observed in practice when a device is searching for APs with *hidden* SSIDs [10]. This is necessary because hidden SSID networks will not respond to broadcast probe requests. An AP configured with a hidden SSID also removes the SSID value when transmitting beacon frames. Ultimately, the use of hidden SSIDs has generally been found to provide little obfuscation value, as multiple techniques exist to recover the AP's hidden SSID [11]. Regardless, they remain in use and can cause devices to be subject to unnecessary privacy leaks [10].

B. VAP Framework

In the original 802.11 standard, ratified in 1999, an AP consisted of a single BSSID and SSID. Strictly interpreted, the standard mandates the use of a single SSID per BSSID. With the proliferation of 802.11 enabled devices in today's environment, there exists a growing requirement for more robust network designs. Consideration of congested channel environments, where various network providers wish to provide network services, highlights an exemplar case for AP configurations allowing multiple SSIDs. The inherent benefits are two-fold: a reduction in infrastructure cost, and an efficient use of the limited channel spectrum. This multi-SSID AP configuration is often described as a VAP. In Figure 1, we contrast the VAP architecture to the traditional single-SSID architecture.

There exists no industry standard delineating an official name, configuration, or implementation for VAPs. Often vendor data-sheets refer to the capability as a *multi-SSID* option, in others, the capability is labeled as a *VAP*. For our purposes we make the following definition.

Definition 1. *A VAP is a logical segmentation of a single physical AP at the Media Access Control (MAC) layer.*

Each VAP is thereby represented by a separate SSID and a unique BSSID (cf. Figure 1). Each distinct VAP on a device may be configured with different MAC settings, Internet Protocol (IP) settings, security settings, network ranges, Dynamic Host Configuration Protocol (DHCP) pools, etc., allowing for a wide range of network designs applicable to numerous scenarios. Of note, each VAP on a single device shares the same wireless channel and physical layer properties.

III. VAP Similarity Features

In this section, we present some possible metrics for determining whether two or more BSSIDs are actually hosted on the same physical AP. From this set of possible features, we will later select those which maximize our chance of correctly correlating BSSIDs to a single device. The feature space will then define a metric of closeness between BSSIDs represented by a graph $\mathbb{G}$.

Consider a set of BSSIDs that occur in observed 802.11 traffic and let ν_i represent the i^{th} BSSID. Each ν_i will then represent a vertex in $\mathbb{G}$. Further, let the set of all M observed BSSIDs originating from the same device be $\mathcal{V} \equiv \{\{\nu_1\}, \{\nu_2\}, \ldots, \{\nu_M\}\}$.

We find that two management frames are inherently predisposed to VAP correlation: beacon and probe response frames. These management frames are by design unencrypted as they support the network discovery process. Beacons are transmitted at a default rate of 100 ms [7] providing a steady stream of potential data from which to correlate VAPs. Probe responses, while not observed at the same rate of occurrence, are effectively always available in environments where client devices are present. Additionally, probe request frames describe natural *edges* in that they represent client-AP interaction. This forms a legitimate framework for community structure in the resulting graph $\mathbb{G}$. Conversely, beacon frames are connectionless and elicit no explicit response, thus do not carry as much inherent information as probe responses. We therefore focus the remainder of our study on the analysis of probe response frames.

The remainder of this section considers in order Information Elements (IEs), MAC address structure, and probe response reception time as possible features useful for VAP correlation.

A. Management Frame Information Elements

Probe response frames contain various configuration and device specific details within IE fields. The IE attributes are made up of both mandatory and optional parameters utilized as part of the AP discovery and selection process. We inspect a variety of these fields for potential use as feature sets in our study.

1) Beacon Interval: This mandatory two-byte IE represents the number of Time Units (TUs) between successive beacon frame transmissions [7]. The beacon interval is commonly set to a default of 100 TU but can be manually configured by a network administrator.

Proposition 1. *Vertex ν_i is not similar to vertex ν_j if*

$$\beta_i \neq \beta_j, \tag{1}$$

where β_i is the beacon interval set for ν_i.

It follows as possible, although not certain, that $\beta_i = \beta_j$ if $\{\{\nu_i\}, \{\nu_j\}\} \subset \mathcal{V}$. However, because the default β is rarely changed, it is much more likely that $\beta_i \neq \beta_j$ if $\{\{\nu_i\}, \{\nu_j\}\} \not\subset \mathcal{V}$. Beacon interval is, therefore, a better discriminator of what VAPs are *not* correlated than a correlation feature.

2) Timestamp: An eight-byte timestamp field contains a value depicting a synchronization timer of the frame's source AP [7]. The purpose of this field is to provide a mechanism to ensure synchronization across all devices on the network.

Proposition 2. *Vertex ν_i is similar to vertex ν_j if*

$$|\tau_i - \tau_j| = \Delta\tau_{ij} < \epsilon_\tau, \tag{2}$$

where $\tau_i = \hat{\tau}_i - t_r$ and $\hat{\tau}_i$ is the timestamp of ν_i and t_r is the time of reception.

Naturally, all VAPs of a distinct device maintain a shared timing function. We therefore expect the timestamp IE to be a strong correlation feature as it specifically represents the number of microseconds the AP has been active [12]. It will therefore always be the case that $|\tau_j - \tau_i| = \Delta\tau_{ij} < \epsilon_\tau$ for VAPs on the same device, but not necessarily true that $|\tau_j - \tau_i| = \Delta\tau_{ij} \geq \epsilon_\tau$ for VAPs on different devices. For instance, if a local area looses power, once power is restored it is likely that all devices in that area will have a similar $\hat{\tau}$ and therefore similar τ. We have purposely defined this similarity feature in terms of ϵ_τ in order to account for a time-varying propagation channel and variable transmit/receive queuing delay.

3) Sequence Number: A twelve-bit field indicates the current sequence number of a frame [7]. Each subsequent frame sent by a wireless device increments the sequence number modulo 4096.

Proposition 3. *Vertex ν_i is similar to vertex ν_j if*

$$|s_i - s_j| = \Delta s_{ij} < \epsilon_s, \tag{3}$$

where s_i and s_j are the sequence numbers attached to probe responses from ν_i and ν_j respectively.

We assume packets sent from the same device will be numbered sequentially regardless of logical separation of higher-layer entities, thus this proposition assumes that VAP probe responses happen approximately (to within ϵ_s) sequentially. The expectation of approximate sequence as opposed to a strict sequence guards against the probability of missed packets. Of note, one difficulty associated with this approach is that the packet numbering happens modulo 4096. It can then be verified that if 29 different APs all receive a probe request and $\epsilon_s = 5$ there is a 50% chance of VAPs being incorrectly correlated via sequence number.

4) Vendor Specific IEs: In addition to mandatory IE fields vendors often include proprietary fields commonly called Vendor Specific IEs. Our review of these fields indicate the use of a *VAP* ID attribute within several prominent manufacturers' probe response frames.

Proposition 4. *Vertex ν_i is similar to vertex ν_j if*

$$v_i = v_j, \tag{4}$$

where v_i is the VAP ID of ν_i.

Using the derived identifier, we are able to trivially link the VAPs to a distinct physical device. However, similarity as defined by this proposition, does not always follow since the VAP ID field is not mandatory and therefore not used by all manufacturers.

5) Device Signature: Lastly, we consider a device signature approach [10], [13], where the set of IEs within a probe request are combined to construct a *device signature* $\mathbf{s}_i = [s_1, s_2, \ldots, s_N]^T$ for ν_i where s_i is the i^{th} IE. Example IEs include:

Proposition 5. *Vertex ν_i is similar to vertex ν_j if*

$$\boldsymbol{s}_i = \boldsymbol{s}_j, \tag{5}$$

where $\boldsymbol{s}_i$ is the signature of v_i.

The expectation then is that the VAPs of a distinct device share the same device signature.

Algorithm 1 Proposed VAP-Identification Scheme

for $k \in$ FeatureSet **do**
  $\hat{\mathbb{G}} = \prod_k \mathbf{A}_k$
end for
$f : \hat{\mathbb{G}} \rightarrow \mathbb{G} = \cup_i \mathbb{G}_{K_i}$
for $i \in \mathbb{G}$ **do**
  $\mathbf{V} \leftarrow$modularityPartition($\mathbb{G}_{K_i}$)
end for

B. MAC Address Structure

Proposition 6. *Vertex ν_i is similar to vertex ν_j if the middle four octets of the MAC addresses belonging to ν_i and ν_j are identical.*

It is well-known that the organizationally unique identifier (OUI) of a MAC address identifies a particular vendor. However, since an AP may modify the locally assigned bit when enabling VAPs, the second and third octets can additionally provide a more robust correlator of colocated VAPs. Further, by observation of empirical data, it is very likely that an implementation of a MAC address scheme on a single piece of hardware will increment only the last octet, leaving the middle four octets unchanged. However, because this is entirely by uncodified convention, it will not necessarily always hold.

C. Reception Time

Proposition 7. *Vertex ν_i is similar to vertex ν_j if*

$$|t_i - t_j| = \Delta t_{ij} < \epsilon_t \tag{6}$$

where t_i and t_j are the times of reception at client u_k of a probe response from ν_i and ν_j respectively.

This proposition captures the idea that probe responses from VAPs resident on the same device should be received, queued, and returned to the requester as probe responses at a similar time for each VAP. While it is conceivable that two probe responses could reach a client u_k at a similar time by chance, especially in dense AP environments where propagation time differences between devices are small, we submit reception time as a possible vendor-agnostic feature for correlating VAPs.

IV. Graph Theoretic Approach

In this section, we present our graph theoretic approach to VAP identification (outlined in Algorithm 1) which leverages similarity features discussed in the previous section. We begin by formally defining vertex adjacency, and then go on to take that definition and use it in a graph-partitioning approach to finding community structure (i.e., correlating VAPs to multi-BSSID APs).

A. Vertex Adjacency

Let the i^{th} client be represented by u_i such that the set of all C observed clients be $\mathbf{u} = [u_1, u_2, \ldots, u_C]^T$. Finally, let $\mathcal{V}_i^{\{K\}}$ represent the i^{th} subset of BSSID(s) originating from the same device where $K \geq 1$ is the number of BSSID(s) associated with device u_i. Therefore, if $K > 1$ then each $\nu_i \in \mathcal{V}_i$ is considered a VAP, and if $K = 1$, the singleton member is the only BSSID associated with the AP.

Definition 2. *Vertex ν_i is adjacent to vertex ν_j iff ν_i and ν_j transmit a probe response to client u_k and $i \neq j$.*

This definition captures the idea of structure in that they have both interacted with the same client. In the case of a device assigned multiple BSSIDs (i.e., VAPs), we expect this to be the case since upon reception of one probe request from a single client, the VAPs will send K probe responses $\forall \nu_i \in \mathcal{V}$. This behavior will create $\binom{K}{2}$ edges and thus the complete graph $\mathbb{G}_K$. The graph structure of $\mathbb{G}$ is represented by an adjacency matrix $\mathbf{A}$ defined by

$$\mathbf{A} = \begin{bmatrix} 0 & \omega_{1,2} & \cdots & \omega_{1,M} \\ \omega_{2,1} & 0 & \cdots & \omega_{2,M} \\ \vdots & & \ddots & \vdots \\ \omega_{M,1} & \omega_{M,2} & \cdots & 0 \end{bmatrix}, \tag{7}$$

where $\omega_{i,j} = \omega_{j,i} \geq 0$ iff ν_i is adjacent to ν_j and $\omega_{i,j} = 0$ otherwise.[1]

B. Multi-Dimensional Representation

We may then define a multi-dimensional graph $\hat{\mathbb{G}}$ for each of the similarity metrics via

$$\hat{\mathbb{G}} = \prod_k \mathbf{A}_k. \tag{8}$$

Alternatively, the various graphs $\mathbf{A}_k$ may be viewed as multiple dimensions of the overall graph $\hat{\mathbb{G}}$, a perspective we hold throughout the remainder of this paper. Optimal projection of the information from each of these dimensions into a single dimension, represented by graph $\mathbb{G}$, is accomplished by the map $f : \hat{\mathbb{G}} \rightarrow \mathbb{G}$. f is then defined as a voting function as follows.

Let the empirical mean of $\mathbf{A}_k$ be represented by

$$\hat{\mu}_k = \frac{1}{e_k} \sum_{ij} \mathbf{A}_{k_{\{i,j\}}}, \tag{9}$$

where e_k is the number of edges represented in $\mathbf{A}_k$, and $\{i, j\}$ is the row/column index of $\mathbf{A}_k$. Next, if the edge $\{\nu_i, \nu_j\}$ exists in $\mathbf{A}_k$, let the vote for each edge $\{\nu_i, \nu_j\}$ in $\mathbb{G}$ to be connected be

$$w_{ij} = \sum_k \frac{C_k}{\mu_k} \hat{\mu}_k, \tag{10}$$

where C_k is a scaling constant and μ_k is the expected value of the nonzero edge weights in $\mathbf{A}_k$. Otherwise, if the edge $\{\nu_i, \nu_j\}$ does not exist in $\mathbf{A}_k$, let the vote for each edge in $\{\nu_i, \nu_j\}$ in $\mathbb{G}$ to *not* be connected be

$$\tilde{w}_{ij} = \sum_k \frac{C_k}{\mu_k} \hat{\mu}_k. \tag{11}$$

[1]Note that Definition 2 precludes self edges, thus the diagonal of $\mathbf{A}$ will always be zero.

By normalizing each of the empirically calculated means $\hat{\mu}_k$ by the actual expected mean μ_k, we provide a method of comparing the relative certainty or weight from one similarity feature with another. The constant C_k provides a heuristic means of adjusting the relative voting weight of each of the K features based on the amount of information that feature carries.

Finally, $\mathbb{G}$ is constructed via

$$\mathbf{A}'_{ij} = \begin{cases} 1 & \text{if } w_{ij} \geq \tilde{w}_{ij} \\ 0 & \text{o.w.} \end{cases} . \tag{12}$$

Thus, the final $\mathbb{G}$ is undirected, unweighted, and symmetric.

We also allow for certain features which prove good discriminators, but have poor correlation properties. This information is included in f through a series of hadamard products

$$\mathbf{A} = \mathbf{A}_1 \odot \mathbf{A}_2 \odot \cdots \odot \mathbf{A}_N \odot \mathbf{A}', \tag{13}$$

where there are N discriminators and $\mathbf{A}'$ is the result of (12).

C. Graph Partitioning

In an ideal scenario, using the above definitions of similarity and voting map f, the resulting $\mathbb{G}$ would be disconnected such that $\mathbb{G} = \cup_i \mathbb{G}_{K_i}$, where each connected portion of the overall graph $\mathbb{G}_{K_i}$ is the complete graph on K_i vertices and $\mathbb{G}_{K_i} \cap \mathbb{G}_{K_j} = \{\varnothing\}, \forall i \neq j$. With this result, each $\mathbb{G}_{K_i} \equiv \mathcal{V}_i^{\{K\}}$ can be interpreted as all $\nu_j \in \mathbb{G}_{K_i}$ being part of the same VAP and the method of VAP identification then ends here. However, it is possible that the above method will produce false edges between VAP communities. To guard against this, we use a divisive graph partitioning approach introduced in [14].

Divisive methods of clustering and graph partitioning have a rich history in the literature [15]–[18]. We focus specifically on the modularity approach [14] due to its relative success among peer methods for finding natural community structure in complex networks. Implicitly, at the heart of all approaches to graph partitioning is a comparison of the graph in its partitioned state against a null model. The null model in the modularity approach is a random graph where the expected degree of each vertex is equal to the actual degree of each vertex. For a given partition, one can find if the number of edges within the newly partitioned communities exceeds the expected number of edges. An affirmative result indicates a desired partition and a negative result provides a natural stopping point to the method, something lacking in other popular graph partitioning tools. Finding the globally optimum partition requires exhaustive trial; however, good approximate means of calculating suboptimal partitions can be done through eigen analysis of the unpartitioned graph [14]. The method also suffers in that it is an inherently successive approach since each graph is only made into two partitions at each step. This creates a method susceptible to the local optimum trap. Nevertheless, it has shown to be robust in the face of its own shortcomings especially when applied to real-world data [15]. Because the voting map $f : \hat{\mathbb{G}} \rightarrow \cup_i \mathbb{G}_{K_i}$ reduces the multi-dimensional representation of similarity to relatively small disconnected subgraphs, and because of the modularity method's proven success in the literature we find it appropriate for our end.

We therefore apply the modularity-based graph partitioning method as a last step in order to guard against incorrect incidental correlations. The result of the overall proposed VAP-identification scheme is $\mathbf{V}$, an unordered set representing the estimated BSSID relationships.

V. Methodology

Over the course of approximately two years (January 2015 to December 2016), we captured unencrypted 802.11 device traffic using inexpensive commodity hardware and open-source software. We primarily use a Nexus 5 Android phone running Kismet *PcapCapture* paired with an AWUS036H 802.11b/g Alfa card. We additionally employ several Raspberry Pi devices running Kismet with three individual wireless cards. Our corpus encompasses approximately 9,000 individual packet captures. The collection contains over 600 GBs of 802.11 traffic, consisting of over 2.8 million unique devices.

A. Ethical Considerations

Our collection methodology is entirely passive, leveraging data that is by design unencrypted. At no time did we perform active actions to stimulate or alter normal network behavior. Our intent is to show the ease with which one can build a similar capability with low-cost off-the-shelf equipment. However, given the nature of our data collection, we consulted with our Institutional Review Board (IRB).

The primary concerns of the IRB centered on: i) the information collected; and ii) whether the experiment collects data "about whom" or "about what." Because we limit our analysis to 802.11 management frames, we do not observe Personally Identifiable Information (PII). Further, humans are incidental to our experimentation as our interest is in wireless device layer-2 MAC addresses, or "what." Again, we have no way to map MAC addresses to individuals.

Finally, in consideration of beneficence and respect for persons, our work presents no expectation of harm, while the concomitant opportunity for network measurement and security provides a societal benefit. Our experiment was determined to not be human subject research and approved by our IRB.

B. Feature Set Assessment

In order to evaluate the efficacy of our feature sets and the overall results of our algorithm we required a method for deriving a *Truth Table*. To this end, we chose to use a subset of devices with which we can extract a VAP ID. We derived VAP identifiers using our laboratory equipment to reverse engineer the proprietary vendor specific IE data fields contained within probe request frames derived from various Juniper, Ubiquiti, and Cisco APs. We write custom Wireshark dissectors in order to efficiently retrieve the VAP identifiers.

First, we evaluate the efficacy of each similarity feature by parsing a subset of our real-world 802.11 collection where the VAP identifiers of Juniper and Ubiquiti are present.

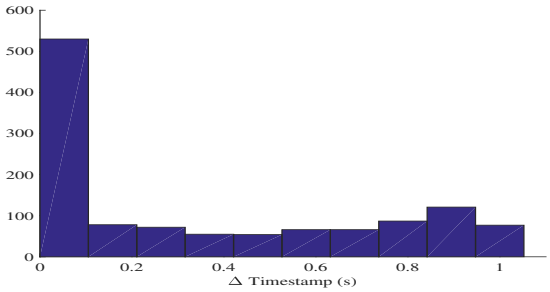


Fig. 2: Δ of timestamp values of successive probe responses

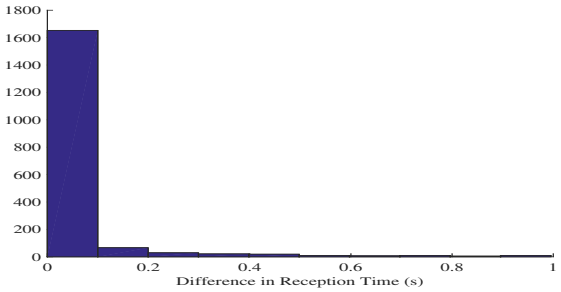


Fig. 3: Reception time Δ between successive probe responses

1) VAP identifiers: A logical question arises, if we have VAP identifiers (ID) then why is their a need for additional correlation analysis? The answer is three-fold: i) the VAP ID parameter is not universally implemented by manufacturers, ii) when used by a manufacturer, not all models utilize the field, and iii) reverse-engineering each vendor-specific IE requires extensive analysis, of which is not a feasible large scale process. As such we limit the use of the VAP ID feature as a means to measure the precision and recall of our graph partitioning results.

2) Beacon Interval: The first IE we evaluate is the beacon interval attribute β. Using our sample set we observe that all VAPs for a single device share the same beacon interval $\beta_i = \beta_j, \forall\{\{\nu_i\},\{\nu_j\}\} \in \mathcal{V}$. This indicates that the beacon interval lends well to use as a discriminator (i.e., VAPs should have the same beacon interval). However, we observe that the diversity of beacon interval values is particularly limited. In our sample set, ≈98% of devices had a beacon interval value of 100 (i.e., $\beta_i = \beta_j$ even when $\{\{\nu_i\},\{\nu_j\}\} \notin \mathcal{V}$), and a range of only 10 distinct values. Therefore, the beacon interval acts as a strong discriminator and weak correlator.

3) Timestamp: Next, we evaluate the timestamp attribute. We observe that timestamp $\hat{\tau}$ is vendor agnostic and remains consistent across VAPs on a distinct device. The following example highlights the value of the timestamp attribute.

Example 1 (No Timestamp)**.** *The following two MAC addresses share similar edges and have the following feature sets (t, s, β), in which result in a false positive correlation:*

$\underline{t}$	$\underline{s}$	$\underline{\beta}$
43.266601000	3721	100
43.285206000	3727	100

Example 2 (With Timestamp)**.** *Now, using the same two MAC addresses, sharing similar edges, and now having included the timestamp we have the feature sets (t, s, β, τ), accurately discriminating when $\tau_i \neq \tau_j$:*

$\underline{t}$	$\underline{s}$	$\underline{\beta}$	$\underline{\tau}$
43.266601000	3721	100	594964.733399
43.285206000	3727	100	232689.714794

We test this attribute across the entirety of our dataset and observe a trivial number of devices where the timestamp value is zero. We expect this is a configuration error, as the parameter is used as a synchronization function and should always be set.

In order to better define closeness of vertices in terms of timestamp values, we analyze a dataset of 1,206 successive probe responses and compare their timestamp difference distribution in Figure 2. The result is highly uniform with the exception of a large proportion of the data lying below 100 ms. We then define a representative edge weighting scheme

$$\omega_{ij,\tau} = 5 \cdot \mathbf{1}_{[0,0.1]}(\Delta\tau_{ij}) + \mathbf{1}_{(0.1,1.3]}(\Delta\tau_{ij}), \qquad (14)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

4) Sequence Number: We evaluate the sequence number attribute s, using our derived VAP identifiers we observe that Juniper devices maintain a shared sequence number across VAPs whereas some Ubiquiti models do not. Similarly, in our lab setting we observe that some Cisco and Aruba devices use a shared sequence number while others do not. We remove the sequence number feature from our algorithm due to ambiguity in VAP sequence number implementations across vendors and models as well as the high probability of collisions[2] within dense wireless environments.

5) Device Signature: We test the device signature methodology using the derived Ubiquiti/Juniper *Truth Tables* and within our lab environment. In both cases, the device signature method often fails to correlate the VAPs. These results are not surprising due to the nature of the VAP architecture. Device signatures are derived from the IEs of the frame which indicate the VAP-specific settings and parameters. The inherent motivation for VAP-based implementations is to allow for customizable and diverse network settings. As such, it follows that device signatures *will* likely be different: $\mathbf{s}_i \neq \mathbf{s}_j$ for $\{\{\nu_i\},\{\nu_j\}\} \in \mathcal{V}$. We therefore rule out signature analysis as a useful feature.

6) Reception Time: This feature is closely coupled to the community structure of our algorithm. Only devices with a perceived closeness as a function of reception time and having shared edges are ever assessed as a possible correlated VAP.

To better define vertex similarity in terms of reception time, we analyze a large multi-vendor dataset consisting of 1,890 identified probe response pairs from the same device and present the results in Figure 3. The data presented as exponential with a mean value of 140 ms with a maximum value at about 1 sec. We can then define an edge weighting function as

$$\omega_{ij,t} = \lambda e^{-\lambda\Delta t_{ij}} \mathbf{1}_{[0,1]}(\Delta t), \qquad (15)$$

where $1/\lambda$ =140 ms.

[2]Recall that the sequence number increments modulo 4096 .

7) MAC Address Structure: Within our Ubiquiti and Juniper test set we observe the following MAC address allocation schemas for VAPs:

1) bytes [1-5] remain constant, byte [6] incremented by 2 and
2) byte [1] local bit set, bytes [2-6] remain constant

We note that while these results follow the convention that the middle four bytes of the MAC address are the same, we do not have evidence that this holds true across all vendors. Additionally, consider the following example of two devices (which do follow the middle-four-byte-MAC convention), taken from real-world data, each with two configured VAPs:

Example 3 (Middle-four-byte-MAC Convention)**.**

```
78:19:F7:73:7E:01
78:19:F7:73:7E:03
78:19:F7:73:7E:C0
78:19:F7:73:7E:C2
```

In this case, the first two MAC addresses are VAPs from the same device and the last two MAC addresses are VAPs from a different device. A graph based on the middle-four-byte convention would be $\mathbb{G} \cong \mathbb{K}_4$. This would result in all four VAPs identified as belonging to a single AP. Due to a lack of standardization, and what appears to be a manufacturer and model defined schema that would lead to false positives in our data set, we chose to remove the MAC address structure from the feature set.

C. Simulated Results

In order to determine an optimal methodology for VAP identification, we used a simulated environment to generate packet capture (PCAP) files. In each simulation, some number of APs are deployed uniformly over an area of 100 m $\times$ 100 m. Some number of clients are then generated and move through the area from a randomly generated point of origin. As each client comes within range of an AP its probe request(s) trigger(s) probe response(s). In order to capture the random nature of the wireless channel, connectivity is modeled as random variable R such that the probability of reception due to shadowing, as a function of distance d, is given by

$$p_R(d) = 1 - \Phi\left(\frac{d-\mu}{\sigma}\right), \tag{16}$$

where $\mu = 30$ m and $\sigma = 3$ m. Here, $\Phi(\cdot)$ is the cumulative distribution function of a Gaussian random variable with mean μ and variance σ.

D. Feature Efficacy

We first seek to evaluate the efficacy of the various features indicative of a VAP outlined previously in this section. To this end, Monte-Carlo trials of the aforementioned simulation are done only with individual correlative dimensions of $\hat{\mathbb{G}}$ only (i.e., there is no information fusion and β is not included since it is used only as a discriminator).

After the graph $\hat{\mathbb{G}}$ is built using only one dimension (i.e., only one feature $k \subset \{\{t\}, \{\tau\}\}$) it is partitioned as per Algorithm 1 for $k = 1$. The results of each individual effort

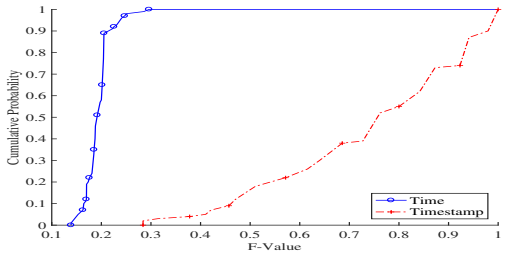


Fig. 4: VAP features evaluated through Monte-Carlo trials

are presented in Figure 4 over 1,000 trials each. Efficacy is measured as F score, a function of both precision and recall, where an F score of 1 denotes perfect VAP identification for the given data. For our application, precision is defined as a ratio of the number of correct graph edges identified to the total number of graph edges identified. Recall is defined as the ratio of the number of correct graph edges to the total number of true graph edges.

As is evident from the results in Figure 4, the ability of timestamp to correlate VAPs far outweighs that of reception time. It therefore makes sense that the timestamp dimension is weighted higher in the voting map f (i.e., $C_\tau > C_t$, cf. (10)–(11)). However, it is also seen that reception time does have some correlative properties. It therefore has intrinsic value which this identification scheme profits from. Despite the small overall value added by time-based correlation, consider the following two cases. First, we have observed vendor traffic that has elected to not broadcast the timestamp field. If this is the case and reception time is not considered, then the results would clearly be inconclusive. Second, if a building or local area experiences a power surge, then it would be that all APs in that area would share similar timestamps as they would all restart at similar times. This is a second example of where a second discriminator would be valuable. Finally, we submit that reception time is a particularly powerful feature in that it is intrinsically linked to propagation time and thus largely bound by physics and not software. For the remainder of the results we let $C_\tau = 2C_t$.

We further restrict our algorithm such that if beacon interval or arrival time do not identify an edge then an edge is not allowed. This allows for discrimination in β and simultaneously acknowledges that if two probe responses do not arrive within ϵ_t then they cannot be correlated. This operation is integrated into the map f by

$$\mathbf{A} = \mathbf{A}_t \odot \mathbf{A}_\beta \odot f(\hat{\mathbb{G}}). \tag{17}$$

E. Multi-Dimensional Projection of Feature Information

Here, we seek to collapse the multi-dimensional graph into one dimension, via the previously discussed map $f : \hat{\mathbb{G}} \to \mathbb{G}$ suitable for the graph partitioning algorithm presented in Section IV. The most straightforward route to project all of the information in one dimension would be to average the edge weights across each of the K dimensions via

$$\omega_{i,j} = \frac{1}{K}\sum_k \omega_{k_{\{i,j\}}}. \tag{18}$$

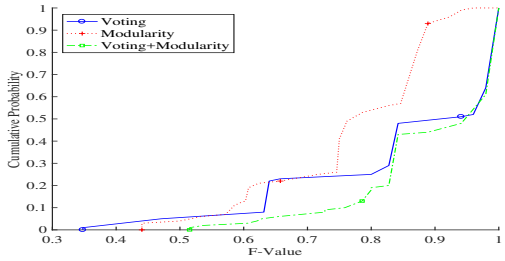


Fig. 5: The performance of the voting map f is compared against taking the mean along each dimension (18) and then partitioning the resulting graph via the modularity-based method.

However, this reduction may not yield an optimal result since, as we have just seen, different layers present a variable amount of correlative power. As an alternative, we propose a voting scheme where each dimension casts a "vote" as to whether or not an edge exists between two vertices. As a result, the final graph is not weighted (as would be the case in the first method of edge weight averaging) since the vote is for whether vertices are similar or not (cf. Equations (9)-(12)).

In order to determine the efficacy of voting, we compare it in another Monte-Carlo study. Here, $\hat{\mathbb{G}}$ is collapsed either via the voting map f or by taking the mean of the graph edges, as in (18), and then performing a modularity-based partition. In both cases, the timestamp is weighted twice as heavily as is reception time (cf. the results in Figure 4). The results of this study are presented in Figure 5 and suggest that both proposed methods are equally as effective at VAP identification with a slight preference given to the voting map. However, when the modularity-based partitioning is applied to the graph projected by voting, even more improvement in performance is realized. It is with this method of voting to reduce dimensionality, followed by a modularity-based partitioning, that we propose to best identify VAP structure.

VI. Results

In this section, we present the results of the final VAP identification algorithm as outlined in the previous sections. We first evaluate the algorithm on small vendor-specific data where the vendor-ID field can provide a means to positively verify results. Next, we apply the algorithm to a much larger multi-vendor dataset to evaluate performance in a more likely real-world application.

A. Vendor Specific Data

Here we use real-world data to validate the algorithm's efficacy. These data are grouped into three sets by vendor: Ubiquiti, Cisco, and Juniper. The test data and results are characterized in Table I. These data show a wide variety of community structure and also demonstrate the results over short and long observation periods. As can be seen in the results, the algorithm is able to correctly identify all of the VAPs in the Ubiquiti and Cisco data. In the Juniper data, only one edge is not identified due to imperfections in the wireless channel.

TABLE I: Test Data and Results by Vendor

	Ubiquiti	**Cisco**	**Juniper**
Probe Responses	1446	226	215
BSSIDs	25	7	32
Single-BSSID Devices	2	1	5
Two-BSSID Devices	1	3	3
Three-BSSID Devices	7	0	7
F Score	**1**	**1**	**0.9787**

B. Multi-Vendor Data

In a real-world implementation, data would likely contain multiple vendors. Additionally, our algorithm provides significant value by automating results over a large dataset as opposed to a traffic analysis approach which is manpower intensive but works well on small datasets. We therefore, move to consideration of two large sets of multi-vendor data.

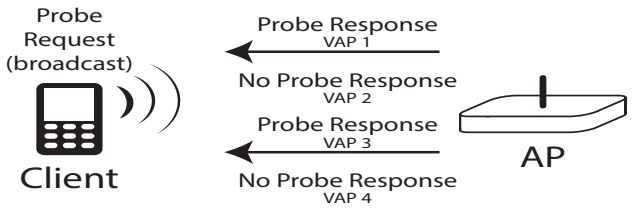
The first set contains 62 different physical devices. Of these APs, the algorithm identified 14 as VAP-enabled. Of the 14 VAP-enabled devices ten are correctly identified and four are each missing exactly one observed BSSIDs (i.e., for these four devices all BSSIDs which belong to the VAP were correctly correlated with the exception of a single hidden SSID node). The remaining 48 single-BSSID devices were also correctly identified. With this data set, the VAPs were all identified with (17) and did not require further graph partitioning. We attribute this to the heterogeneous nature of the underlying data which is a result of a less dense distribution of APs.

The second set contains 504 APs, of which the algorithm identified 101 as VAP-enabled. These data contained a more dense AP distribution and thus invoked the partitioning step in the case of two VAPs. Interestingly, the partitioning step was correctly invoked in one case, while the other resulted in an over-partitioning. As such, the 101 VAP-enabled devices identified by our algorithm were in actuality 100 devices. This represents the single instance in either test set where we fail to identify all non-hidden SSID nodes for a given physical device. As was the case in our first test, we fail to correlate the hidden SSID-enabled VAPs. For each test we achieve precision of 1.0 and .99 respectively. When allowing for the exception of the hidden SSIDs, we achieve a recall of 1.0 for both data sets.

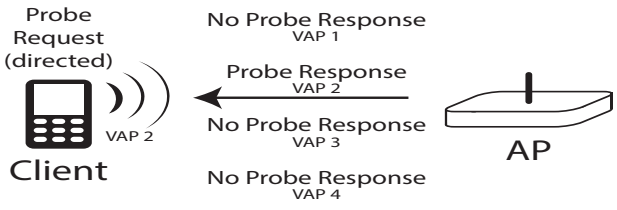
C. Hidden SSID Problem

The use of a hidden SSIDs creates two problems in our community structure-based methodology for VAP correlation analysis. First as depicted in Figure 6a, when a client transmits a broadcast probe request, a probe response is not elicited from any hidden SSIDs configured VAPs. In the case of a directed probe request, illustrated in Figure 6b, we observe a single probe response frame issued from the AP of interest. We gain no insight into the community structure of the physical device as only the queried SSID associated VAP responds.

Similarly, while not observed within these data sets, imperfections in the collection environment where a single probe response is dropped from a set of probe responses will inherently lead to a lack of community structure. In this case, the node itself is missing from the data set entirely.



(a) Hidden SSID Behavior - Broadcast Probe Request



(b) Hidden SSID Behavior - Directed Probe Request

Fig. 6

D. MAC Address Structure

Initially we assumed that the MAC address allocation structure of VAPs would follow a consistent method, thereby allowing for use as a correlation classifier. However, after evaluating our algorithm's results we find this not to be the case, even within a single vendors implementation. The following example serves to illustrate the complexity of one vendors VAP MAC address structure. The first octet has had the locally assigned bit set, the second has been replaced by the sixth, the third and fifth are swapped, and the fourth remains constant:

Example 4 (Byte stucture)**.** *The following two VAP MAC addresses are observed with feature sets that meet all requirements for similarity yet have different middle four octets:*

```
90:72:40:23:0D:EC
92:EC:0D:23:40:70
```

Furthermore, the allocation schemes observed within our dataset (revealed while using our correlation construct) were surprisingly diverse. Across vendor types, and across a single manufacturer's product line we identified 13 unique MAC address schemes. We posit that our novel correlation methodology can be utilized to classify these schemes and organize by device model. This classification could be used for extending our methodology in the future.

VII. Conclusion

In summary, we have presented a graph theoretic algorithm for VAP identification. The algorithm took a multi-dimensional set of data represented by $\hat{\mathbb{G}}$ and reduced it to one dimension $\mathbb{G}$ via a novel voting map f. Several possible features in the probe response framework were evaluated for use in the algorithm where reception time and timestamp were found to be strong correlative features. Beacon interval was also found to be useful for edge discrimination as was reception time. The algorithm was found to be accurate on both small vendor specific data and large multi-vendor data. A strength of the algorithm was revealed in that idiosyncratic byte structures were identified that may have confounded a more traditional traffic-analysis approach to VAP identification. Our algorithm has shown how automated large data analysis can be successful through graph-theoretic methods.

VIII. Acknowledgments

We thank Erik C. Rye, Michael Koppel, Lamont Brown, and Danny Flack for early feedback and contributions.

Views and conclusions are those of the authors and should not be interpreted as representing the official policies or position of the U.S. government.

References

[1] C. Hoffman, "Warning: Guest Mode on Many Wi-Fi Routers Isn't Secure," Jun 2015. [Online]. Available: https://tinyurl.com/n8b4q34

[2] A. Tsow, M. Jakobsson, L. Yang, and S. Wetzel, "Warkitting: The Drive-by Subversion of Wireless Home Routers," *Journal of Digital Forensic Practice*, vol. 1, no. 3, pp. 179–192, 2006.

[3] Y. Grunenberger and F. Rousseau, "Virtual Access Points for Transparent Mobility in Wireless LANs," in *IEEE Wireless Commun. Networking Conf.*, 2010, pp. 1–6.

[4] T. Hamaguchi, T. Komata, T. Nagai, and H. Shigeno, "A Framework of Better Deployment for WLAN Access Point Using Virtualization Technique," in *IEEE 24th Int. Conf. Advanced Info. Networking Applications Workshops*, 2010, pp. 968–973.

[5] S. He, T. Hu, and S.-H. Gary Chan, "Toward practical deployment of fingerprint-based indoor localization," *IEEE Pervasive Comput.*, vol. 16, no. 2, pp. 76–83, March 2017.

[6] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise Indoor Localization Using Smart Phones," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 787–790.

[7] IEEE Std 802.11-2007, "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," pp. C1–1184, Jun. 2007.

[8] T. Kropeit, "Dont Trust Open Hotspots: Wi-Fi Hacker Detection and Privacy Protection via Smartphone," 2015.

[9] M. Ghering and E. Poll, "Evil Twin Vulnerabilities in Wi-Fi Networks," 2016.

[10] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A Study of MAC Address Randomization in Mobile Devices and When it Fails," *Proceedings on Privacy Enhancing Technologies*, vol. 4, pp. 268–286, 2017.

[11] V. Ramachandran, *Backtrack 5 Wireless Penetration Testing: Beginner's Guide*. Packt Publishing Ltd, 2011.

[12] D. A. Westcott, D. D. Coleman, B. Miller, and P. Mackenzie, *CWAP Certified Wireless Analysis Professional Official Study Guide: Exam PW0-270*, 1st ed. Alameda, CA, USA: SYBEX Inc., 2011.

[13] D. Gentry and A. Pennarun, "Passive Taxonomy of WiFi Clients Using MLME Frame Contents," *arXiv preprint arXiv:1608.01725*, 2016.

[14] M. E. Newman, "Finding Community Structure in Networks Using the Eigenvectors of Matrices," *Physical Review E*, vol. 74, no. 3, 2006.

[15] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The Bottlenose Dolphin Community of Doubtful Sound Features a Large Proportion of Long-lasting Associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[16] J. Kim and K.-H. Cho, "Robustness Analysis of Network Modularity," *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 4, pp. 348–357, Dec. 2016.

[17] M. E. Newman, "Equivalence Between Modularity Optimization and Maximum Likelihood Methods for Community Detection," *Physical Review E*, vol. 94, 2016.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Journal Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.