MEDIUM ACCESS CONTROL PROTOCOL RECOGNITION FOR ADAPTIVE WIRELESS COMMUNICATIONS NETWORKS

Margaret Rooney
Advisor: Dr. Mark Hinders
Applied Science Department, William & Mary

Abstract

The ability to distinguish between transmissions from wireless communications networks using different medium access protocols can facilitate the incorporation of a cognitive radio into an existing network, promote air-to-air and ground-to-air communications for on-demand mobility and commercial aircraft, or can improve efforts to disrupt enemy communications effectively. The way in which users access the electromagnetic spectrum provides one of the most prominent distinctions between reservation based and contention based medium access control protocols. We exploit the regular timing of transmissions from networks utilizing a reservation based time-division multiple access (TDMA) protocol to differentiate between transmissions governed by TDMA and by contention based carrier sense multiple access (CSMA) protocols. Our approach leverages modular arithmetic to identify periodicity in transmission timings, and an unsupervised k-means algorithm to generate distinct TDMA and CSMA clusters. A variety of supervised machine learning algorithms are then explored to build a protocol classifier. We next develop an automated clustering algorithm to run on a set of center frequencies extracted from a noisy network trace to determine whether the network has a multi-channel or single-channel architecture. Such information can be used to determine whether the network is employing a frequency division multiple access protocol to access the electromagnetic spectrum.

1. Introduction

Increased use of the electromagnetic spectrum in recent years has led to the development of new technologies with the ability to assess spectrum usage and to adjust transmission parameters intelligently to take advantage of unused frequency bands. Such technology can be incorporated into air-to-air and air-to-ground aeronautical communication systems for efficient transmission of sensor data both in NASA's On-Demand Air Mobility and Urban Air Mobility aircraft and in conventional aircraft.[1–3] For adaptive nodes to utilize vacant spectrum channels efficiently without causing unintended interference to other users, it is beneficial to determine how other networks are accessing a particular channel. The method by which nodes of a communications network share frequency channels is referred to as the medium access control (MAC) protocol. Knowledge of a network's MAC protocol can facilitate the incorporation of a cognitive radio into an existing network, or can improve efforts to disrupt enemy communications. Such knowledge can be attained by employing machine learning techniques to analyze transmissions of the network under consideration. In this paper, we present a two-stage machine learning approach to reservation based MAC protocol recognition. First, an unsupervised k-means clustering algorithm is employed to partition the dataset into reservation-based protocol and contention-based protocol clusters. These groupings then become labels for the data, and a variety of supervised machine learning algorithms are explored to generate a MAC protocol classifier. The next part of our work focuses on developing an automated clustering algorithm which can be run on a set of center frequencies extracted from a noisy network trace in order to obtain an estimate of the set of center frequencies actually utilized by the network. Such information can then be used to determine whether or not the network is occupying multiple channels of the electromagnetic spectrum.

2. Related Work

The majority of the literature concerning medium access control protocols for wireless communication networks centers around the development and design of efficient time-slotted, random access, and hybrid protocols. Lai et al.[4] design medium access protocols for cognitive users to opportunistically access the spectrum in the absence of primary users. Yahya and Ben-Othman[5] discuss MAC protocols for wireless sensor networks, including their design and the various advantages and disadvantages associated with each. Este et al.[6] investigate the implementation of the support vector machines algorithm to identify traffic emanating from specific applications, while Soysal and Schmidt[7] perform internet traffic classification using flow traces. There exist numerous surveys providing detailed discussion of MAC protocol design for both wireless sensor networks and cognitive radio networks.[8–13] A number of authors have investigated the use of machine learning for improved MAC protocols, for primary user detection, and in cognitive radio networks.[14–19] The publications of Hu et al.[20,21] and Yang et al.[22] present supervised machine learning approaches to MAC identification. In their works, the authors use received power and channel state features combined with a support vector machines model to distinguish between protocols.

Estimating the number of inherent clusters in a dataset is an integral part of the clustering process.[23,24] A variety of methods for cluster evaluation have been proposed in the literature, some of which rely on visual techniques like the elbow method[25,26] and silhouette plots,[26] and some of which utilize metrics such as the gap statistic.[23] Many researchers have focused on determining the optimal number of clusters from a prespecified range of cluster values, and almost exclusively consider multi-dimensional datasets. Pelleg and Moore[27] propose a k-means based clustering algorithm that uses Bayesian Information Criteria to estimate the ideal number of clusters from a predefined range of potential clusters. Kryszczuk and Hurley[28] estimate the number of clusters in a dataset using various cluster validity indices. Dudoit and Fridlyand[29] use resampling and prediction accuracy to estimate the number of clusters in a dataset for improved tumor classification. Others have proposed algorithms that use iterative methods to estimate the number of clusters. Yao and Choi[30] developed a clustering algorithm suited to grouping web-pages of similar contents that measures average inter-cluster similarity against a constant value. Rosenberger and Chehdi[31] proposed an automatic clustering algorithm for image segmentation based on k-means which evaluates the number of clusters at each partition using the average dispersion measure. The algorithm we have developed uses k-means clustering and an intra-cluster variance based metric to automatically determine the inherent number of clusters in a one-dimensional dataset of transmission center frequencies.

3. Medium access control protocols

The way in which nodes of a network access the electromagnetic spectrum is referred to as the MAC protocol. Our MAC protocol recognition algorithm first focuses on distinguishing between reservation based time division multiple access (TDMA) and contention based carrier-sense multiple access (CSMA) protocols, then determines whether the network is occupying multiple frequency channels, indicative of a frequency division multiple access (FDMA) protocol. Figures 1 and 2 show examples of networks governed by these protocols.

3.1. Time-division multiple access

Time-division multiple access is a MAC protocol that allows multiple users to transmit on a single channel without collisions. This is accomplished through segmenting time into a series of repeating frames that are further divided into individual time slots. Each time slot is assigned to a network user so that only one user may transmit at any given time. Time slots may be re-assigned to accommodate new users entering the network. Applications include radio, cellular, and satellite systems.

3.2. Carrier-sense multiple access

Carrier-sense multiple access is a MAC protocol in which users access the spectrum randomly and opportunistically. To avoid collisions, users transmit only when the channel seems vacant. If another transmission is in progress, the user waits until the ongoing transmission is complete before using the channel. Since CSMA is contention based and therefore is not constructed of rigid time slots, users can transmit packets of varying sizes whenever the channel is available. Applications include Wi-Fi and radio systems.

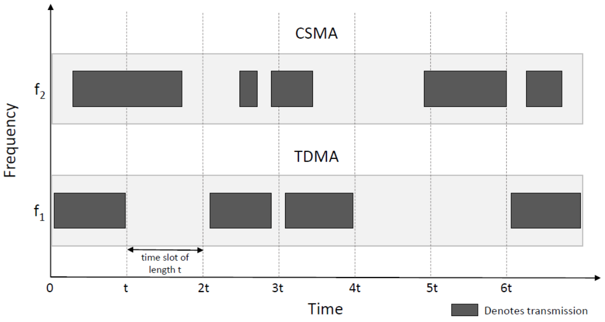

Figure 1: Illustration of TDMA and CSMA MAC protocols. Frequency channel f_1 uses TDMA. It is divided into time slots of length t, and each transmission (represented by a solid grey box) fits within the boundaries of its defined time slot. Frequency channel f_2 uses CSMA, where transmissions are of varying lengths and occur at irregular intervals.

3.3. Frequency division multiple access

Frequency division multiple access is characterized by its use of numerous transmission and reception channels.[32] Each node in the network transmits packets on a designated channel while receiving packets from other nodes on each of the other channels. FDMA networks can be implemented in conjunction with other medium access methods to create hybrid network protocols. TDMA-FDMA networks maintain time-slotted transmission schedules in which each time slot is associated with a different center frequency. CSMA-FDMA networks allow nodes to access the spectrum randomly while supporting a wide variety of packet lengths. However, CSMA-FDMA differs from traditional CSMA protocols in that CSMA-FDMA networks have individual transmission channels for every node in the network.

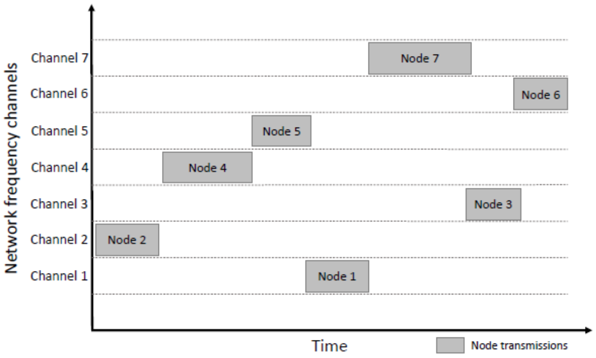

Figure 2: Illustration of a network using FDMA. The network is composed of seven nodes each transmitting on a separate channel. Nodes monitor each of the channels to receive packets from the rest of the network.

4. Machine learning

We have developed a MAC protocol recognition algorithm that integrates both unsupervised and supervised machine learning techniques. Since the data used for our algorithm development were unlabelled, we began by employing an unsupervised clustering method to partition the preprocessed data into two groups: reservation based and random access protocols. These clusters became class labels, and a classifier was trained to identify new inputs as TDMA or CSMA governed transmissions. Next, a k-means clustering algorithm was run on the set of center frequencies recorded for all transmissions to determine the number of channels utilized by the network. The following sections provide brief descriptions of the machine learning methods used in various iterations of the MAC recognition algorithm.

4.1. Unsupervised learning

4.1.1. k-Means clustering

The k-means clustering algorithm partitions an unlabelled dataset $X = x_1, x_2, \ldots, x_n$ into k clusters, such that the variance within each cluster is at a minimum. The process of determining an optimal partition of the data begins by randomly assigning data points to one of k groupings. The mean, or centroid, of each cluster is calculated and then used to compute the variance of the cluster. A data point x_i initially belonging to cluster c_j is re-assigned to cluster c_k if $d(x_i, c_j) > d(x_i, c_k)$, where d is the distance between two points according to the user-specified distance metric. The new cluster centroids and variance are then calculated. Clusters are altered in this way until each new cluster centroid differs from the previous one only very slightly. At this point, the intra-cluster sum-of-squares

$$\sum_{i=1}^{n}\sum_{j=1}^{k}\min_{\mu_j \in C}(\|x_i - \mu_j\|)^2 \tag{1}$$

has been minimized. Here, the data are segmented into k clusters, each with centroid $\mu \in C$.

4.2. Supervised learning

4.2.1. Support Vector Machines

Support vector machines (SVMs) are supervised machine learning models commonly used for classifying high-dimensional data. The SVM classification algorithm seeks to establish a maximum margin decision hyperplane between linearly separable classes. For datasets that are not linearly separable, the hyperplane can be found by using a kernel function to project the data into a higher dimension where the classes become linearly separable. The optimal hyperplane maximizes the minimum distance between the hyperplane and the points in the training dataset $x_i \in \mathbb{R}^n$ by solving the quadratic problem

$$\min_{w,b} \frac{1}{2}\|w\|_2^2 \tag{2}$$

subject to

$$y_i(w^T x_i + b) \geq 1, \ \forall i \in \mathbb{N} \tag{3}$$

where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are primal variables. The three most commonly used kernel functions are the linear kernel (4), polynomial kernel (5), and radial basis function kernel (6).

$$K(x, x_i) = x_i^T x \tag{4}$$

$$K(x, x_i) = (x_i^T x + c)^m, \ c > 0, \ m \in \mathbb{Z} \tag{5}$$

$$K(x, x_i) = e^{-\gamma \|x_i - x\|_2^2}, \ \gamma \geq 0 \tag{6}$$

4.2.2. Naïve Bayes

Naïve Bayes classifiers are supervised machine learning algorithms that can provide probabilistic outputs rather than hard decisions of class membership. Bayes classifiers consider each input feature independently, rather than relating combinations of features to a certain class. The algorithm is based on Bayes Theorem, which states that

$$p(c|x) = \frac{p(x|c)p(c)}{p(x)} \tag{7}$$

to compute the posterior probability of class c given predictor x, or $p(c|x)$. The prior probability of a class is denoted $p(c)$, the prior probability of a predictor is $p(x)$, and the probability of predictor x given class c is written $p(x|c)$.

Since all features are independent of the class, for some set of features $X = x_1, x_2, \ldots x_n$ and class c,

$$p(x_1, x_2, \ldots, x_n|c) = \prod_{i=1}^{n} p(x_i|c). \tag{8}$$

Then, for two classes C_1 and C_2, the Naïve Bayes classifier is defined as

$$f_{NB}(X) = \frac{p(C_1)}{p(C_2)} \prod_{i=1}^{n} \frac{p(x_i|C_1)}{p(x_i|C_2)}, \tag{9}$$

where

$$E \in C_1 \iff \frac{p(C_1)}{p(C_2)} \geq 1. \tag{10}$$

4.2.3. k-Nearest Neighbors

The k-nearest neighbors (kNN) classifier assigns a class label to an input by examining class membership of the neighboring data points in the n-dimensional feature space. A variety of distance metrics can be used to evaluate the distance between the input and neighboring data points. This simple algorithm requires no explicit training step, and only the number of neighbors to consider and the desired distance metric need be specified by the user.

5. Data sets

The datasets used to develop, evaluate, and refine the algorithm were generated by software defined radio (SDR) testbeds and EMANE network simulation software.[33] These datasets included TDMA, CSMA, FDMA, TDMA-FDMA, and CSMA-FDMA networks transmitting over the 2.0-2.1 GHz and 2.4 - 5.0 GHz frequency bands. Traces of packet transmissions were collected for each network. Included in the fifteen recorded trace features are transmit time in microseconds, packet length in bytes, source and target node identification, and center frequency in GHz. Therefore, a collection of n transmissions for some network N is recorded in trace T_N as follows:

$$T_N = \begin{bmatrix} transTime_1 \ packetLen_1 \cdots \ centFreq_1 \ \cdots \\ \vdots \\ transTime_n \ packetLen_n \cdots \ centFreq_n \ \cdots \end{bmatrix}$$

6. TDMA/CSMA classification

Each trace contains a record of the transmit time of every packet sent in the network, so the transmit time feature for trace T can be written as $T = t_1, t_2, ..., t_n$ for a trace containing n transmission events. This feature was used to calculate the differences between consecutive transmissions

$$T_d = \begin{bmatrix} t_2 \\ \vdots \\ t_n \end{bmatrix} - \begin{bmatrix} t_1 \\ \vdots \\ t_{n-1} \end{bmatrix} = \begin{bmatrix} td_1 \\ \vdots \\ td_{n-1} \end{bmatrix}$$

which became the basis of the feature vector generation stage for the TDMA/CSMA classifier.

6.1. Feature vector generation

In number theory, modular arithmetic is defined by a modulus $N > 1$ and all integers $r \in [0, N-1]$ such that any integer taken modulo N is congruent to some $r \in [0, N-1]$. The congruence class of an integer k modulo N can be determined by writing k as

$$k = m * N + r, \tag{11}$$

where $m, r \in \mathbb{Z}$ and $0 < r < N$. Then,

$$r \equiv k(mod \ N), \tag{12}$$

and so k is in the same congruence class as r.

The goal of this work is to exploit the regular timing of TDMA transmissions to facilitate differentiation between TDMA and CSMA protocols. Ideally, the differences between transmission times of TDMA emissions should be integer multiples of the predetermined time slot length, τ. Thus for any TDMA transmission time difference td_i and modulus τ,

$$td_i \ mod \ \tau \equiv 0. \tag{13}$$

In reality, a variety of factors prevent TDMA transmission time differences from being exact multiples of the time slot duration. To account for noise, the modulo value of a transmit time difference td_i and the time slot length τ is normalized with respect to τ so that near-integer multiples of τ are treated as integer values. So, for

$$td_i \ mod \ \tau \equiv r, \tag{14}$$

the normalized transmit time difference modulo value, r_{norm}, is calculated as

$$r_{norm} = \begin{cases} \frac{r}{\tau} & r < 0.5\tau \\ \frac{\tau - r}{\tau} & r \geq 0.5\tau \end{cases} \tag{15}$$

Therefore, for any td, $r_{norm} \in [0, 0.5]$. If the majority of normalized values of transmit time differences modulo τ are approximately zero, this indicates that transmissions frequently occurred at regular intervals, and so are likely TDMA. If the normalized values of transmit time differences modulo τ are dispersed fairly evenly throughout the interval $[0, 0.5]$, this then indicates that transmissions occurred at random intervals, characteristic of CSMA protocols.

The feature vector generated for each trace contains two elements: the mean of the normalized transmit time difference modulo values, and the variance of the normalized transmit time difference modulo values. Thus the feature vector for trace T_i containing n transmission events is $[\mu_i, \sigma_i^2]$, where the mean of the normalized transmit time difference modulo values is calculated as

$$\mu_i = \frac{1}{n-1} \sum_{j=1}^{n-1} r_{norm,j} \tag{16}$$

and the variance of the normalized transmit time difference modulo values is calculated as

$$\sigma_i^2 = \frac{1}{n-1} \sum_{j=1}^{n-1} (r_{norm,j} - \mu_i)^2. \tag{17}$$

6.2. Time slot length estimation

In general, due to the slotted structure of a TDMA network, some portion of the transmissions will inevitably occur in consecutive time slots regardless of the amount of traffic. The percentage of transmissions that occur in consecutive time slots will be high for congested networks, where vacant time slots are rare, but will be low for uncongested networks where few consecutive time slots are used in each frame. For CSMA traces, the choice of a potential time slot duration to use as the modulus has little effect on the outcome of the modular arithmetic-heavy feature generation, since the randomness of transmission times ensures that modulo values are fairly equally spread throughout the interval $[0, 0.5]$.

Initially, the time slot length for a specific trace was estimated as the minimum transmission time difference. However, this did not consistently result in an accurate estimation of the time slot length since in some TDMA traces, the minimum transmission time difference was much less than the time slot length due to noise. Therefore, the time slot duration used as the modulus in the feature vector generation was estimated individually for each trace by averaging a small percentage of the shortest transmission time durations. A range of values between 5% and 10% of the shortest transmission time durations were tested, with 6% repeatedly producing the best approximation of the time slot length for the entire spectrum of network traffic levels. Using such a small percentage of the shortest transmission time durations to estimate the time slot length worked equally well for both highly congested networks and uncongested networks, where oftentimes less than 20% of transmissions occurred in consecutive time slots. In most cases, the estimated time slot length was within 2% of the actual time slot duration for TDMA traces.

6.3. Machine learning for classification

After generating feature vectors for each of the roughly 160 TDMA and 160 CSMA traces, the unlabelled dataset was fed into a k-means clustering algorithm that partitioned the dataset into two distinct clusters. One cluster, centered near the origin, was composed of traces with low means and low variances. The second cluster was composed mainly of traces with a mean value of about 0.25 and a variance of about 0.0200.

The cluster indices generated by the k-means clustering algorithm were used to label the full dataset, which was then split into training and test sets.

Since the k-means clusters accurately divided the data into TDMA and CSMA clusters with the exception of only three data points, these cluster indices were well-suited to being used as supervised data labels. Plots of the k-means groupings and the actual groupings are provided in figures 3 and 4. Various training/test splits were imposed to assess the performance of each type of model. The accuracy of each classifier was calculated as the difference between the total cases and the misclassified cases divided by the total number of cases. Each classifier performed well for the varying training/test data splits, with all accuracies exceeding 90%.

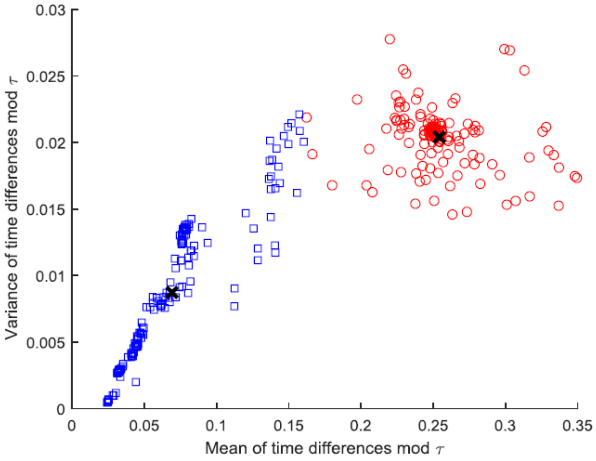

Figure 3: Plot of the two clusters obtained as a result of running the k-means clustering algorithm on a dataset composed of 325 data points. The centroid of the cluster near the origin is located at (0.0692, 0.0087), and that of the cluster in the top right of the plot is located at (0.2542, 0.0204).

6.4. Classifier accuracy

The classifiers were trained and evaluated on three different training/test splits: 70/30, 50/50, and 30/70. For all training/test splits, each of the five classifiers accurately predicted the class membership of the majority of the test data, with only a few misclassified cases. Results are contained in Table 1.

7. FDMA recognition

In order to determine the number of channels, the second component of the algorithm runs a k-means clustering algorithm on the set of center frequencies recorded for all transmissions of the network. In theory, the optimal number of clusters to generate should be equivalent to the number of channels occupied by a network. The center frequency data is initially partitioned into two clusters, and in each subsequent implementation of k-means the number of clusters increases by one until the number of channels can be determined confidently. This optimal number of clusters is found by identifying the minimum number of clusters that explains the majority of variance within the dataset.

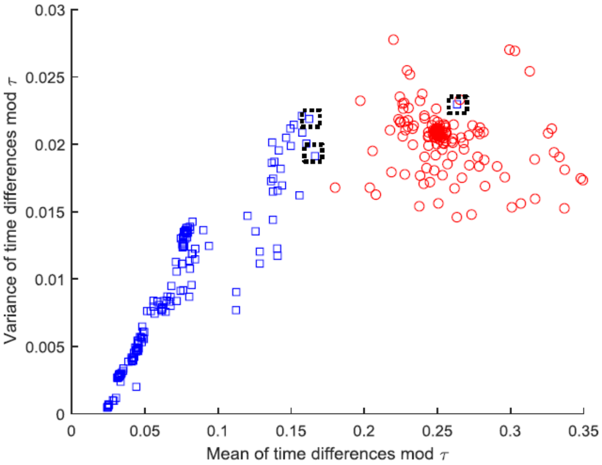

Figure 4: Plot of true TDMA and CSMA groupings. TDMA traces are represented by squares, and CSMA traces are represented by circles. A comparison between the ground truth clusters and the k-means clusters (Fig. 3) shows only three misclassified data points, outlined by dashed boxes.

In every iteration of the k-means algorithm, the variance of each cluster c_i with $i \in [1, k]$, denoted σ_i^2, is calculated as

$$\sigma_i^2 = \frac{1}{m-1} \sum_{j=1}^{m} |x_j - \mu_i| \tag{18}$$

where m is the number of datapoints x in cluster i, and μ is the the mean of cluster i. These variances are summed to give the total value of the intra-cluster variances for a set of clusters (19).

$$\sigma_{tot,k}^2 = \sum_{i=1}^{k} \sigma_i^2 \tag{19}$$

This sum is normalized through division by the variance of the entire set of center frequencies σ_{all}^2 and is then subtracted from one so that the resulting value var_k corresponds to the amount of the total variance within the data that can be explained by segmenting the data into k clusters, represented by (20).

$$var_k = 1 - \frac{\sigma_{tot,k}^2}{\sigma_{all}^2} \tag{20}$$

Train/test split	Classifier	% Accuracy	Misclassified	Train data	Test data
70/30	kNN	97.98	2	227	99
	Naïve Bayes	97.98	2		
	SVM (linear kernel)	94.95	5		
	SVM (polynomial kernel)	98.99	1		
	SVM (rbf kernel)	98.99	1		
50/50	kNN	98.77	2	163	163
	Naïve Bayes	98.77	2		
	SVM (linear kernel)	98.16	3		
	SVM (polynomial kernel)	99.39	1		
	SVM (rbf kernel)	98.16	3		
30/70	kNN	98.68	3	99	227
	Naïve Bayes	97.81	5		
	SVM (linear kernel)	92.54	17		
	SVM (polynomial kernel)	99.56	1		
	SVM (rbf kernel)	97.81	5		

Table 1: Accuracies of each of the classifiers for various training/test data splits.

The var_k values for each k are recorded in a k-by-2 matrix. The first two columns of Table 2 provide an example of such a matrix.

k	var_k	r_k
1	0	—
2	0.5084	2.9196
3	0.6826	1.9096
4	0.7737	2.2212
5	0.8148	0.6962
6	0.8738	2.5006
7	0.8974	0.9519
8	0.9221	0.9690
9	0.9477	1.0771
10	0.9714	0.9503
11	0.9964	168.2766

Table 2: An example of the amount of variance within data that can be explained by k clusters for an 11-node TDMA-FDMA network is contained in column 2. Column 3 contains the slope ratios computed for the same network.

Plotting the results produces a curve from which the optimal number of clusters can be identified by locating the point at which the slopes of successive segments begin to approximate zero. Figure 5 shows an example of such a plot.

The points on the plot are of the form (k, var_k), where k corresponds to the number of clusters and var_k is calculated using (20). The slopes of the line segments are labelled according to the endpoints of the line, so that the slope of the segment connecting points $(k-1, var_{k-1})$ and (k, var_k) is denoted $L_{k-1,k}$. For each $i \in [2, k]$, the slope ratio shown in (21) is computed. Table 2 provides an example of slope ratios calculated for a plot of the (k, var_k) values computed for an 11-node TDMA-FDMA network.

$$r_i = \frac{L_{i-1,i}}{L_{i,i+1}} \tag{21}$$

The largest slope ratio, $r_{max} = max(r_i),\ i \in [2, k]$, nearly always corresponds to the point where subsequent segments have a slope of approximately zero. When r_{max} exceeds a user-defined threshold, the algorithm stops increasing the number of clusters on which to run the k-means algorithm, and the optimal number of channels is identified as the number of clusters corresponding to the maximum slope ratio r_{max}. Since the maximum slope ratio is usually significantly greater than each of the other slope ratios, the choice of the threshold is flexible. If the maximum ratio remains below the threshold after a set number of iterations of the k-means algorithm, the optimal number of clusters is assumed to be one.

After having determined the number of channels the network is likely using to transmit information, the k-means algorithm is run a final time on the collection of center frequency data with the k-value set equal to the estimated number of channels. This last iteration of the k-means algorithm is used to create a new list of center frequencies for the network trace by replacing the cluster index of each transmission event with the associated cluster centroid. The new set of center frequencies can be used to learn which nodes are communicating on each center frequency.

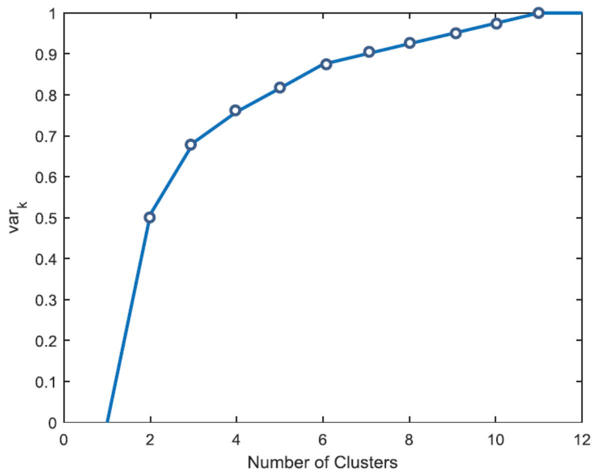

Figure 5: Example of a plot of the total variance within the data that can be explained by segmenting the data into k clusters. The optimal number of clusters is chosen by identifying the point on the curve at which slopes of succeeding segments are approximately zero. The curve above was generated using the trace from an 11-node TDMA-FDMA network.

8. Results and discussion

The modular arithmetic method used to generate machine learning features from network traffic was developed under the assumption that only packet transmission times could be reliably detected. Since no additional features are required as inputs to the algorithm, this method provides a way to recognize the MAC protocol of a network blindly. All classifiers trained on the feature vectors created using the modular arithmetic method described in Section 6.1 achieved an accuracy of over 90%. Only data points midway between the cluster concentrations in the feature space were misclassified.

Although each of the classifiers accurately predicted the correct class of nearly every test data point, two of the three classification algorithms, SVM and kNN, produce a hard decision on class membership rather than calculating the probability that a data point belongs to a certain class. Therefore, an advantage of the Naïve Bayes classifier is the option of a probabilistic output, which provides some idea of the certainty with which a class is assigned to each input.

Table 3 contains details of the few cases misclassified by the Naïve Bayes model for each training/test data split. For the Naïve Bayes model trained on 30% of the full dataset, since the misclassified cases were nearer the concentration of CSMA training data than the concentration of TDMA training data, it was unsurprising that they were all incorrectly classified as CSMA traces. The Naïve Bayes classifier trained on 50% of the entire dataset misclassified only two of the test data points. All training data points surrounding the two incorrectly labelled points were TDMA, so these misclassifications are not surprising. The data points misclassified by the Naïve Bayes model trained on 70% of the complete dataset fell on the border between the TDMA and CSMA classes in the feature space, shown in Fig. 6. Since the data points are midway between the clusters and the CSMA class probability is only marginally higher than the TDMA class probability for each, these misclassifications are reasonable.

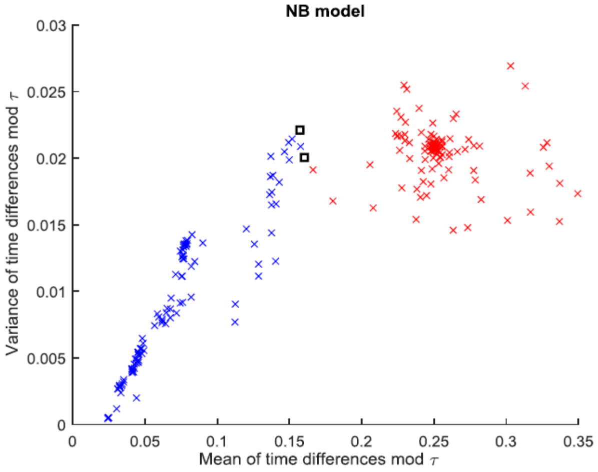

Figure 6: Plot of training data and two cases misclassified by the Naïve Bayes model trained on 70% of the entire dataset. Misclassified points are represented by squares.

The FDMA recognition component of the algorithm was generated and refined using data from a variety of different protocol scenarios, including TDMA, CSMA, FDMA, TDMA-FDMA, and CSMA-FDMA networks. All networks utilized between one and eleven channels, with each channel supporting either a single node or the entirety of the network's nodes. The algorithm accurately estimated the center frequencies of nearly all channels for each FDMA-based network in the dataset, with differences from actual center frequencies less than several hundred kHz.

Since the main component of the frequency clustering algorithm uses an unsupervised machine learning technique, no training data were required to generate the algorithm. Therefore, all datasets were used to evaluate the accuracy of the channel estimation. The frequency clustering algorithm was tested on 60 sets of network traces, 30 of which were

Train/test Split	P(TDMA)	P(CSMA)	Pred. class	Actual class	Mean feature	Var feature
30/70	0.1098	0.8911	CSMA	TDMA	0.1603	0.0200
	0.2377	0.7623	CSMA	TDMA	0.1518	0.0214
	0.4749	0.5241	CSMA	TDMA	0.1466	0.0205
	0.1102	0.8898	CSMA	TDMA	0.1494	0.0221
	0.3246	0.6754	CSMA	TDMA	0.1494	0.0212
50/50	0.6895	0.3105	TDMA	CSMA	0.1623	0.0219
	0.7161	0.2839	TDMA	CSMA	0.1664	0.0191
70/30	0.4145	0.5855	CSMA	TDMA	0.1573	0.0221
	0.3935	0.6065	CSMA	TDMA	0.1603	0.0200

Table 3: Class membership probabilities, actual and predicted class, and feature values for all data points misclassified by the Naïve Bayes classifier for all training/test data splits.

from multi-channel FDMA networks and 30 of which were collected from single channel non-FDMA networks. Across all network traces, the algorithm was tasked with identifying a total of 170 center frequencies from noisy traces. This evaluation was repeated numerous times for differing levels of added noise. The reasonable range of noise to introduce to the center frequencies of the simulated data was determined through a Monte Carlo simulation, which indicated that detection error generally does not exceed 0.002% for a 1 MHz signal. To test for accuracy, the algorithm was run on the entire set of traces ten times, each time introducing a different amount of error into the set of center frequencies. The amount of noise ranged between 0-0.1% of the channel bandwidth, which corresponded to between -5 and 5 kHz. The accuracy was calculated as the fraction of the 170 center frequencies that were correctly identified. A plot of the accuracy for all amounts of detection error is shown in Fig. 7. The algorithm consistently identified over 95% of the center frequencies.

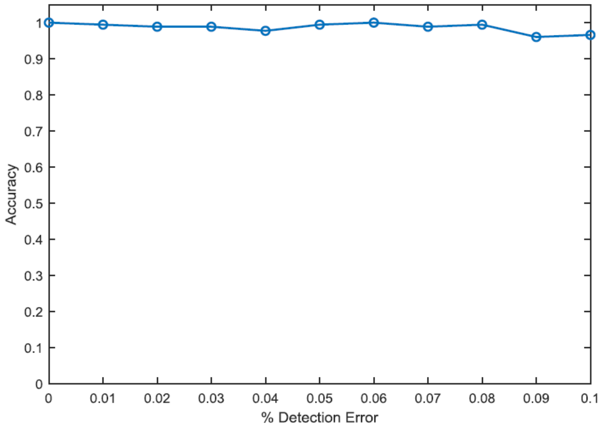

Figure 7: Plot illustrating the accuracy of the frequency clustering algorithm for varying amounts of detection error introduced into the set of center frequencies.

9. Conclusions

This work presents a MAC protocol recognition algorithm to differentiate between transmissions governed by various protocols. Such knowledge can be used by sensors in both vertical takeoff and landing and conventional aircraft to adapt transmission parameters to optimize use of the available spectrum. We used a modular arithmetic-based method to extract features from the transmission times of packets and developed a two-stage machine learning approach to TDMA/CSMA recognition that first uses a k-means clustering algorithm to partition the data into two clusters. Then, we used these groupings to label the data and explored a variety of supervised machine learning algorithms to generate a set of classifiers, all of which achieved an accuracy of over 90%. The channel clustering component of the algorithm was developed for the purpose of recognizing whether a network is employing an FDMA based protocol. It uses an unsupervised k-means clustering algorithm on one dimensional noisy center frequency data to estimate the actual center frequencies which a network is using to transmit packets, and uses that list of frequencies to determine the number of nodes transmitting on each channel. The accuracy of the algorithm consistently exceeded 95% for differing levels of detection error, and was successful in distinguishing between FDMA and non-FDMA based networks. Future work will include extending the MAC recognition algorithm to accommodate a broader range of protocols and finding a more efficient way to recognize non-FDMA networks.

Acknowledgements

The work presented here was funded in part by the Virginia Space Grant Consortium and by the Naval Research Laboratory in Washington, D.C.

References

[1] Ponnu Jacob, Rajendra Prasad Sirigina, AS Madhukumar, and Vinod Achutavarrier Prasad. Cognitive radio for aeronautical communications: A survey. *IEEE Access*, 4:3417–3443, 2016.

[2] B Holmes, R Parker, D Stanley, P McHugh, L Garrow, P Masson, and J Olcott. Nasa strategic framework for ondemand air mobility. 2017.

[3] Nasa embraces urban air mobility, calls for market study.

[4] Lifeng Lai, Hesham El Gamal, Hai Jiang, and H Vincent Poor. Cognitive medium access: Exploration, exploitation, and competition. *IEEE Transactions on Mobile Computing*, 10(2):239–253, 2011.

[5] Bashir Yahya and Jalel Ben-Othman. Towards a classification of energy aware mac protocols for wireless sensor networks. *Wireless Communications and Mobile Computing*, 9(12):1572–1607, 2009.

[6] Alice Este, Francesco Gringoli, and Luca Salgarelli. Support vector machines for tcp traffic classification. *Computer Networks*, 53(14):2476–2490, 2009.

[7] Murat Soysal and Ece Guran Schmidt. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation*, 67(6):451–467, 2010.

[8] José Marinho and Edmundo Monteiro. Cognitive radio: survey on communication protocols, spectrum decision issues, and future research directions. *Wireless networks*, 18(2):147–164, 2012.

[9] Liljana Gavrilovska, Daniel Denkovski, Valentin Rakovic, and Marko Angjelicinoski. Medium access control protocols in cognitive radio networks. In *Cognitive Radio and Networking for Heterogeneous Wireless Networks*, pages 109–149. Springer, 2015.

[10] Kurtis Kredo II and Prasant Mohapatra. Medium access control in wireless sensor networks. *Computer networks*, 51(4):961–994, 2007.

[11] Jie Xiang, Yan Zhang, and Tor Skeie. Medium access control protocols in cognitive radio networks. *Wireless Communications and Mobile Computing*, 10(1):31–49, 2010.

[12] Yi Zhi Zhao, Chunyan Miao, Maode Ma, Jing Bing Zhang, and Cyril Leung. A survey and projection on medium access control protocols for wireless sensor networks. *ACM Computing Surveys (csuR)*, 45(1):7, 2012.

[13] Claudia Cormio and Kaushik R Chowdhury. A survey on mac protocols for cognitive radio networks. *Ad Hoc Networks*, 7(7):1315–1329, 2009.

[14] Mohammad Abu Alsheikh, Shaowei Lin, Dusit Niyato, and Hwee-Pink Tan. Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4):1996–2018, 2014.

[15] Charles Clancy, Joe Hecker, Erich Stuntebeck, and Tim O'Shea. Applications of machine learning to cognitive radio networks. *IEEE Wireless Communications*, 14(4), 2007.

[16] Karaputugala Madushan Thilina, Kae Won Choi, Nazmus Saquib, and Ekram Hossain. Pattern classification techniques for cooperative spectrum sensing in cognitive radio networks: Svm and w-knn approaches. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1260–1265. IEEE, 2012.

[17] Mario Bkassiny, Yang Li, and Sudharman K Jayaweera. A survey on machine-learning techniques in cognitive radios. *IEEE Communications Surveys & Tutorials*, 15(3):1136–1159, 2013.

[18] T Charles Clancy, Awais Khawar, and Timothy R Newman. Robust signal classification using unsupervised learning. *IEEE Transactions on Wireless Communications*, 10(4):1289–1299, 2011.

[19] Zhu Han, Rong Zheng, and H Vincent Poor. Repeated auctions with bayesian nonparametric learning for spectrum access in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 10(3):890–900, 2011.

[20] Sanqing Hu, Yu-Dong Yao, and Zhuo Yang. Mac protocol identification approach for implement smart cognitive radio. In *Communications (ICC), 2012 IEEE International Conference on*, pages 5608–5612. IEEE, 2012.

[21] Sanqing Hu, Yu-Dong Yao, and Zhuo Yang. Mac protocol identification using support vector machines for cognitive radio networks. *IEEE Wireless Communications*, 21(1):52–60, 2014.

[22] Zhuo Yang, Yu-Dong Yao, Sheng Chen, Haibo He, and Di Zheng. Mac protocol classification in a cognitive radio network. In *Wireless and Optical Communications Conference (WOCC), 2010 19th Annual*, pages 1–5. IEEE, 2010.

[23] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[24] André Hardy. An examination of procedures for determining the number of clusters in a data set. In *New approaches in classification and data analysis*, pages 178–185. Springer, 1994.

[25] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[26] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[27] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

[28] Krzysztof Kryszczuk and Paul Hurley. Estimation of the number of clusters using multiple clustering validity indices. In *International Workshop on Multiple Classifier Systems*, pages 114–123. Springer, 2010.

[29] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome biology*, 3(7):research0036–1, 2002.

[30] Zhongmei Yao. Automatically discovering the number of clusters in web page datasets. In *Proceedings of the 2005 International Conference on Data Mining, DMIN 2005*, 2005.

[31] Christophe Rosenberger and Kacem Chehdi. Unsupervised clustering method with optimal estimation of the number of clusters: Application to image segmentation. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 1, pages 656–659. IEEE, 2000.

[32] GholamHossein EkbataniFard. Multi-channel medium access control protocols for wireless sensor networks: A survey. *Journal of Advances in Computer Research*, 2011.

[33] Extendable mobile ad-hoc network emulator (emane).