Topology Reconstruction and Characterisation of Wireless Ad Hoc Networks

Jon Arnold*†, Nigel Bean†, Miro Kraetzl*, Matthew Roughan† and Matthew Sorell‡

*Defence Science and Technology Organisation
PO Box 1500, Edinburgh, S.A. 5111, Australia
Email: {jon.arnold, miro.kraetzl}@dsto.defence.gov.au

†School of Mathematical Sciences
University of Adelaide, SA 5005, Australia
Email: {nigel.bean, matthew.roughan}@adelaide.edu.au

‡School of Electrical and Electronic Engineering
University of Adelaide, SA 5005, Australia
Email: matthew.sorell@adelaide.edu.au

Abstract— **Wireless ad hoc networks provide a useful communications infrastructure for the mobile battlefield. In this paper we apply and develop passive radio frequency signal strength monitoring and packet transmission time profiling techniques, to characterise and reconstruct an encrypted wireless network's topology. We show that by using signal strength measurements from three or more wireless probes and by assuming the use of carrier sense multiple access with collision avoidance, for physical layer control, we can produce a representation of a wireless network's logical topology and in some cases reconstruct the physical topology. Smoothed Kalman filtering is used to track the reconstructed topology over time, and in conjunction with a weighted least squares template fitting technique, enables the profiling of the individual network nodes and the characterisation of their transmissions.**

Keywords: Wireless Networks, Wireless Ad Hoc Protocols, 802.11, Monitoring, Template Fitting, NS-2.

I. Introduction

Wireless Ad Hoc Networks (WAHNs) continue to be the focus of many research efforts, with a large number of protocols having been developed or under development. A possible application for WAHNs and the one focused on here, is the use of WAHN protocols to provide networked mobile battlefield communications. From an adversaries point of view, the communications provided by a battlefield WAHN could provide a valuable source of intelligence. The number of communicating nodes, their location and the nodes which provide the most communications, are all valuable sources of information which could be used to change the course of a battle. We wish to investigate just how much information can be gleaned from a WAHN, even when encryption techniques are used to secure both the protocol and data.

The problem we consider here, is to determine how to reconstruct this secure network's topology, without decoding the transmissions and without knowing unique node identifiers such as IP addresses. The approach makes minimal assumptions about the WAHN. In one component of the algorithm we exploit the specific behaviour of the IEEE 802.11 MAC protocol, but the algorithm is generic, and could be redesigned to exploit the behaviour of most wireless MAC protocols.

The paper's main contributions are techniques that produce logical graph representations of WAHNs over time, and a method that profiles and characterises their nodes using only passive, time synchronised Radio Frequency (RF) signal strength measurements. These measurements are gathered by wireless probes at different known locations, and are used to calculate the origin of each transmission [1], [2]. A density based clustering technique [3], over fixed successive time intervals, in conjunction with inferences based on the MAC protocol provides the topology information.

Kalman filtering is used to track each individual node's movement. We show that by using a simple linear dynamics model, smoothed Kalman filters are capable of tracking the node movement. We also show that by using the predicted location of each node, we can associate nodes recursively across the successive logical graphs and can therefore track a significant proportion of the topology over time.

Finally we investigate if it is possible to classify transmitting nodes as sources, sinks or relays of data. An adversary may wish to deny communications on the battlefield and therefore attempt to disrupt communications going through a crucial network relay node. Similarly, for intelligence gathering purposes an adversary may wish to determine which node generates the majority of traffic, this could be a command or surveillance node in the WAHN.

We show that by using the successive time measurements between each transmission, we can generate timing histograms that represent the transmission protocols used by particular nodes. These histograms characterise the nodes' transmission profiles for successive logical graphs. Using a least squares approach, we fit these node histograms to a known set of transmission protocol timing histograms and consequently are able to classify them as sources, sinks or relays.

Section II describes the RF propagation environment and the NS-2 simulation framework that was used to generate the wireless data for probing. The simulated probes are used for node localisation using multilateration in Section III. The positional estimates are clustered using a density based technique in Section IV, where subgraphs are produced representing the networks topology by incrementally clustering the location estimates. Kalman filtering is used for tracking node movement in Section V. Section VI develops the node profiling using template fitting, and we conclude and propose future work in Section VII.

II. Preliminaries

Estimating a WAHN's topology from RF signal strength measurements is fraught with much uncertainty. The RF propagation environment can vary markedly, especially with node movement. Fading effects caused by the propagation path or the channel bandwidth can result in large variances in the received signal strength. However, when it is assumed that the network being studied is fully encrypted and that only passive RF measurements can be undertaken, the received signal strength measurements and their respective timings provide a good means for network characterisation.

A free space propagation model was used to simulate the RF propagation loss, with the received signal strength calculated at each probe using the distance d between it and the associated nodes. Typically the received power decays proportionally to d^{-n} where n is the path loss exponent. Rappaport [4] shows that n varies between 2 and 4, and for the free space model $n = 2$. It has also been shown that variations in the received power measurements in dB, can be modeled by a Gaussian distribution and that the standard deviation σ, of the received power can be as low as 4 and as high as 12 [4], [5]. For our simulations, the exact distance from the probes to the known node locations was calculated to provide a perfect measurement. Gaussian noise with a mean of zero and a variance of 20, was added to this to simulate mildly noisy free space propagation distance estimates. Although this channel model is overly simplified, it's adequate for developing the topology reconstruction methods that are this paper's main contribution. A subsequent paper will report on more detailed channel fading and time difference of arrival (TDOA) simulations.

NS-2 was used to generate the simulated WAHN transmissions, as it provides multiple ad hoc protocol implementations, simulated applications and the ability to include mobility. It also allows for the simulation of simplified RF propagation channels using characteristics of typical 802.11b WLAN hardware. A 30 second, ten node WAHN simulation, implementing the Ad-hoc On-Demand Distance Vector (AODV) protocol was developed to investigate this topology reconstruction problem. Figure 1 shows the simulation's node movement, with the node's positions plotted at one second intervals.

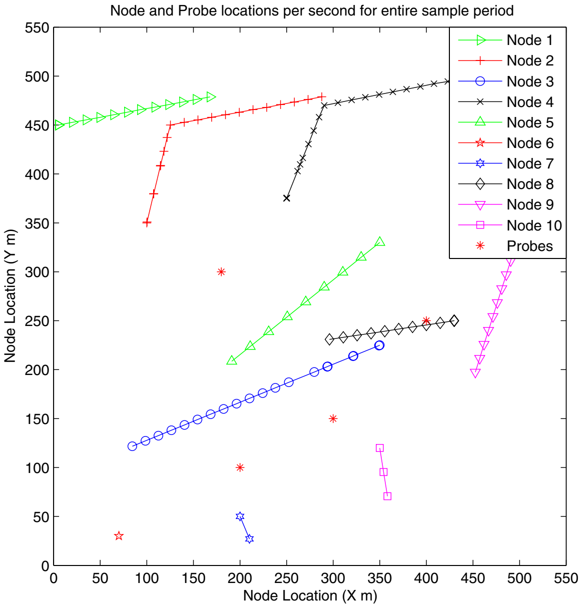


Fig. 1. Node tracks and probe locations plotted every second.

Three separate transmission sessions were implemented, two UDP constant bit rate sessions and one FTP session over TCP, each starting and stopping at different times. The Node's movement was structured to ensure that routing changes had to occur.

The trace files produced from the NS-2 simulations provided the data for experimentation, which was filtered to include only the packet level communication timing and node location data. Three or more simulated networked probes with positions (x_p, y_p) were used to obtain the measurements of the received signal strength from the NS-2 data in Matlab; they are represented as asterisks in Figure 1.

III. Node Localisation

To calculate an estimated node location, the noisy distance measurements were compared to a carrier sensing threshold distance ($CSthresh$) that is calculated according to the sensitivity of a standard 802.11b receiver plus 6 dB of loss. The additional loss is included to allow for receiver variations. If $CSthresh$ is exceeded for at least three measurement probes, then a Minimum Mean Square Estimate (MMSE) for each node location (x_{est}, y_{est}) can be generated. This is often termed multilateration in the WAHN location literature. To solve the MMSE we need n location estimates, where $n \geq 3$. These are calculated by using the probe distance measurements d_i

$$d_i = \sqrt{(x_i - x_{est})^2 + (y_i - y_{est})^2}, i = 1, \dots, n. \quad (1)$$

Squaring and rearranging these terms yields the following equation for each probe measurement

$$-x_i^2 - y_i^2 = x_{est}^2 + y_{est}^2 - 2(x_{est}x_i + y_{est}y_i) - d_i^2. \quad (2)$$

For n such probe equations the $x_{est}^2 + y_{est}^2$ can be removed by subtracting the p_n probe equation from the set, with the resultant being in the form of $v = \beta A$, where

$$v = \begin{bmatrix} -x_1^2 - y_1^2 + x_{p_n}^2 + y_{p_n}^2 - d_n^2 + d_1^2 \\ -x_2^2 - y_2^2 + x_{p_n}^2 + y_{p_n}^2 - d_n^2 + d_2^2 \\ \vdots \\ -x_{n-1}^2 - y_{n-1}^2 + x_{p_n}^2 + y_{p_n}^2 - d_n^2 + d_{n-1}^2 \end{bmatrix},$$

the unknown node's location estimate is

$$A = \begin{bmatrix} x_{est} \\ y_{est} \end{bmatrix},$$

$$\beta = \begin{bmatrix} 2(x_{p_n} - x_1) & 2(y_{p_n} - y_1) \\ 2(x_{p_n} - x_2) & 2(y_{p_n} - y_2) \\ \vdots & \vdots \\ 2(x_{p_n} - x_{n-1}) & 2(y_{p_n} - y_{n-1}) \end{bmatrix}.$$

A is solved using the Moore-Penrose generalised matrix inverse solution for the MMSE [6]

$$A = (\beta^T \beta)^{-1} \beta^T v. \tag{3}$$

Figure 2 plots the actual node locations for every transmission without noise and the estimated node locations from (3), for every transmission that is measured by three or more probes. This data represents the full 30 seconds of simulated AODV communications that produced nearly 32,000 separate RF transmissions. The estimated location plot shows the anticipated increase in errors in the location estimates at the edges of the probes' detection ranges (nodes 1, 2 and 4), and no estimates for node 1's location in the early stages of the simulation, as less than three probes can measure its transmissions.

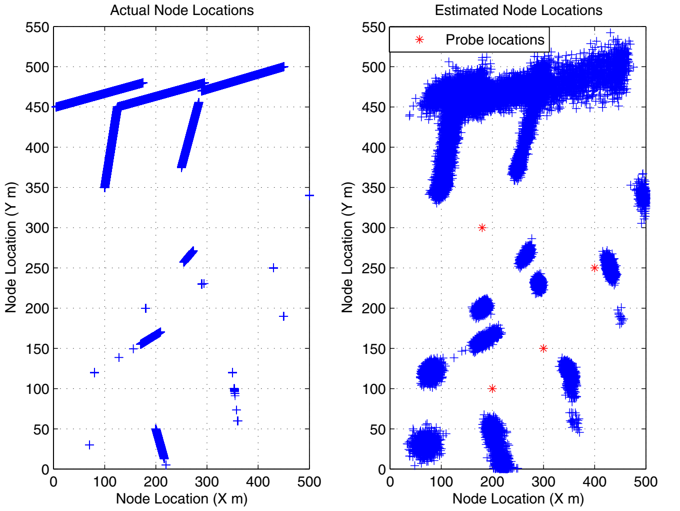


Fig. 2. Actual node transmissions, the probe locations and the MMSE node locations.

IV. Density Based Clustering

A. Clustering using DBSCAN

The location estimates generated by solving the MMSE above provide an estimate for every RF transmission that is received simultaneously by at least three probes. Consequently, thousands of estimates can be generated every second, particularly when there is considerable node movement, or when data packets have to traverse the WAHN. The noise in these estimates requires some form of clustering of the results, so that a centroid for each of the node's locations can be generated.

The clustering algorithm used to group each node estimate must be able to cope with arbitrary cluster shapes to allow for node movement. It also needs to be able to cluster the estimates with minimal knowledge of the network dynamics. With the large number of location estimates the processing must also be as efficient as possible. These requirements suggest that a density based clustering algorithm would be most appropriate, consequently the DBSCAN algorithm by Ester *et al.* [3], was used.

DBSCAN works by clustering points within neighbourhoods that are defined by a radius Eps (the Euclidean distance between the points) and a minimum number of points $MinPts$, within the neighbourhood. If the density of points within the radius Eps exceeds the $MinPts$ threshold, then a neighbourhood is formed and a core point is created. The cluster grows by differentiating between core and border points. Core points are essentially within the centre of the cluster or neighbourhood and are within the radius of multiple border points. Border points are within a neighbourhood but cannot themselves be core points as they don't exceed the $MinPts$ threshold. The algorithm seeds itself with a random starting point and determines if that point can form a core point. If so, the algorithm loops about this point building the cluster by creating and merging neighbourhoods that satisfy the Eps and $MinPts$ parameters. This continues until all the data points are classified. Points that don't fit within clusters are labeled as noise.

The values for Eps and $MinPts$ were determined empirically, although a knowledge of what can be expected for the thinnest cluster does help as a starting point. Ester *et al.* suggest simple heuristics for choosing these values, however for this data set, investigating the separation of transmission clusters versus clustering time for different values of Eps and $MinPts$ produced the best results. Three–dimensional clustering in space and time was used. The distance between points was calculated using a scaled Euclidean distance from the first time record to every other time record, and the Euclidean distance for each X and Y location, for every node to every other node for all transmissions, given by

$$Dist = \sqrt{\alpha(\triangle T)^2 + (\triangle X)^2 + (\triangle Y)^2}. \tag{4}$$

The time scaling represented by α, was incorporated to proportionally balance the magnitude differences between the distance measurements and the time measurements. The inclusion of time clustering allowed for better discrimination between clusters, compared to using only the nodes' locations. It also allowed for an increased Eps radius and a smaller $MinPts$. The clustered results for the entire data set is depicted in Figure 3. Of particular interest is the clustering error for nodes 1 and 2, where even with clustering over time, the clusters have merged to form a single large cluster. The merging of these two separate tracks cannot be prevented using only three dimensions.

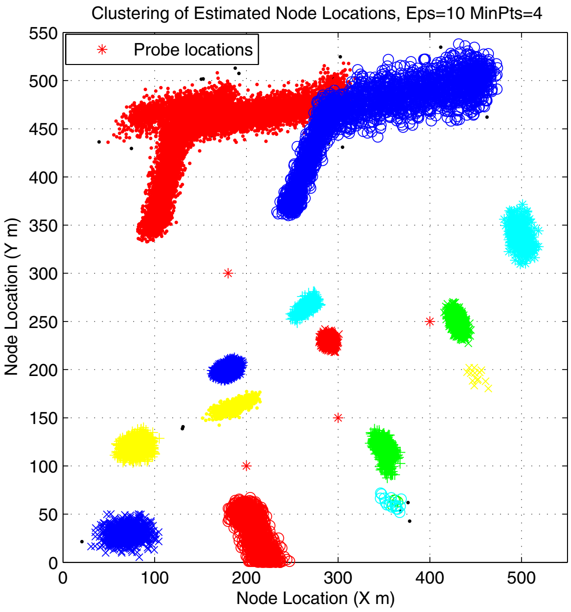


Fig. 3. Clustering of Estimated Node locations.

B. Generating Subgraphs from the Clustered Transmissions

Having generated the clustered node estimates we are now in the position where we can generate graphs that represent the logical connections between the nodes. Aggregating all the transmission data into one graph is inappropriate when there is node movement, so the clustering and consequent graphs need to be generated over sequential, uniform time intervals $\triangle t$. To generate the adjacency matrix for each time interval, we use the underlying 802.11 MAC protocol that governs the RF transmissions. As the MAC protocol implements Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA), we know that there must be a Request-To-Send (RTS) and Clear-To-Send (CTS) handshake between communicating nodes. Assuming this, in conjunction with the maximum propagation range between nodes, which is based on the receiver threshold (the amount of received signal strength that must be exceeded for error free demodulation of the RF transmissions), we infer links/edges between successive transmitting nodes. Simulations have shown that the vast majority of edges are correctly inferred using this method.

Figure 4 shows the subgraphs produced from clustering two consecutive one second transmission intervals. By producing clusters and the associated subgraphs over a reduced time interval we also reap the benefit of much faster clustering. The run time is of O(nlog(n)), where n represents the number of points to be clustered. It also helps prevent clusters from merging, particularly when the minimum number of points for clusters is kept small. For this data it prevents the merging of clusters as occurred in Figure 3 for nodes 1 and 2.

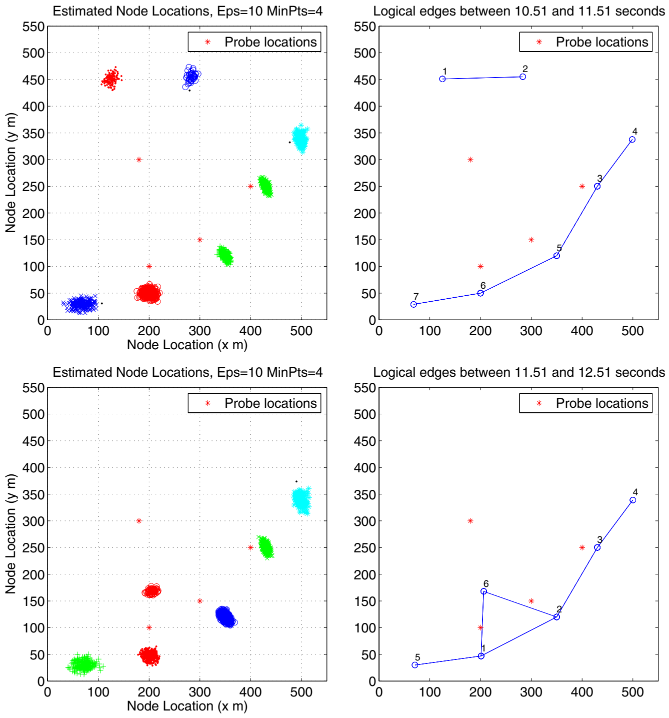


Fig. 4. Two consecutive subclusterings of the estimated node locations and their respective subgraphs.

V. Kalman Tracking of Node Locations

The subgraphs produced in the preceding section, provide only a single average or centroid point for the estimated location of each unknown node. This location gives no indication of the accuracy of the estimate. We therefore assume that the distributions for our node location estimates are normally distributed as,

$$f(X) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}, \qquad (5)$$

where p is the number of variables, Σ is the covariance matrix, and μ is the vector of the expected values of X, namely the mean locations of the x and y coordinates, or the centroids of the clusters from section IV.

As our nodes' locations are assumed normally distributed, we can use a simple Kalman filter [7], [8] to track the nodes' movements from graph to graph. We need to uniquely label the nodes across successive subgraphs to allow for the reconstruction of the WAHN topology and for node characterisation.

The Kalman filter is essentially a state space algorithm used for estimating the track of an object, using its past and current measurements, in this context, the node's location. A process model is used to represent the dynamics of the tracked node, and assuming linear dynamics, the state of the target is represented as

$$x_{t_{i+1}} = F_{t_i} x_{t_i} + w_{t_i}, \tag{6}$$

where the target state at time t_{i+1} is predicted using the current state x_{t_i}, and the state transition matrix F_{t_i} describes the target dynamics from time t_i to t_{i+1}. Here w_{t_i} represents the process noise and is modeled as a zero mean Gaussian distribution with known covariance Q_{t_i}. This predicted state is compared to the measured state using a measurement model

$$z_{t_i} = H_{t_i} x_{t_i} + v_{t_i}, \tag{7}$$

where z_{t_i} is the measurement vector, H_{t_i} is the measurement matrix, and v_{t_i} is the measurement noise modeled as a zero mean Gaussian with known covariance R_{t_i}.

The Kalman filter is implemented using a two step process, a prediction phase based on equation (6) where the noise is introduced by calculating the error covariance $P_{t_{i+1}}$ for the estimates

$$\begin{aligned} \hat{x}^-_{t_{i+1}} &= F_{t_i} \hat{x}_{t_i}, \\ P^-_{t_{i+1}} &= F_{t_i} P_{t_i} F^T_{t_i} + Q_{t_i}, \end{aligned} \tag{8}$$

and a correction phase, based on the observed deviation from the prediction.

The filter is effectively tuned using the Kalman gain K_{t_i} which minimises the mean square error between the predicted state and the measured state. The corrected estimate $\hat{x}_{t_{i+1}}$ and its covariance $P_{t_{i+1}}$ are calculated as

$$\begin{aligned} K_{t_i} &= P^-_{t_{i+1}} H^T_{t_i} (H_{t_i} P^-_{t_{i+1}} H^T_{t_i} + R_{t_i})^{-1}, \\ \hat{x}_{t_{i+1}} &= \hat{x}^-_{t_{i+1}} + K_{t_i} (z_{t_i} - H_{t_i} \hat{x}^-_{t_i}), \\ P_{t_{i+1}} &= (1 - K_{t_i} H_{t_i}) P^-_{t_{i+1}}. \end{aligned} \tag{9}$$

Ideally the choice of the process model should accurately represent the expected node movement. However to simplify the problem, a constant velocity model has been implemented to represent the node movement such that $X = [x, y, \dot{x}, \dot{y}]^T$, where $\dot{x}$ and $\dot{y}$ represent the x and y node velocities respectively. Simulation has shown that this is adequate for tracking the node movement for the NS-2 data. Consequently the transition and process noise covariance matrices are given as

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{10}$$

$$Q = \sigma^2 \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix}, \tag{11}$$

where $T = t_{i+1} - t_i$ is the update interval, and $\sigma^2 = 20$ is the covariance for the continuous time process noise.

The measurement matrix and its covariance, also assumed independent, are similarly set to

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \; R = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}. \tag{12}$$

A separate Kalman filter is required to track each individual node, and is initialised as above. Unique node labeling is required for tracking nodes over successive time intervals, and it is here where the Kalman filter's corrected predictions $\hat{x}_{t_{i+1}}$ are used for labeling. For the first sample period any detected nodes are uniquely labeled; in subsequent periods each detected node is compared to the **predicted** locations from the previous set of unique node locations. The previously corrected location $\hat{x}_{t_{i+1}}$ is used to predict the current estimated location $\hat{x}^-_{t_{i+1}}$ which is then compared to the current measured node's location. If the error between the current measured location and the predicted location falls within a gating window, derived from the maximum movement rate, then the node is relabeled, otherwise a new unique label is created.

This process is repeated for every sample period with each unique node location stored and sequentially tracked. On completion of the online tracking the Rauch-Tung-Striebel (RTS) algorithm [9], [10] is used to recursively smooth the estimated node locations. This provides a reduced error between the estimated node location and the measured node location. The smoothed Kalman filtered tracks, are plotted in Figure 5.

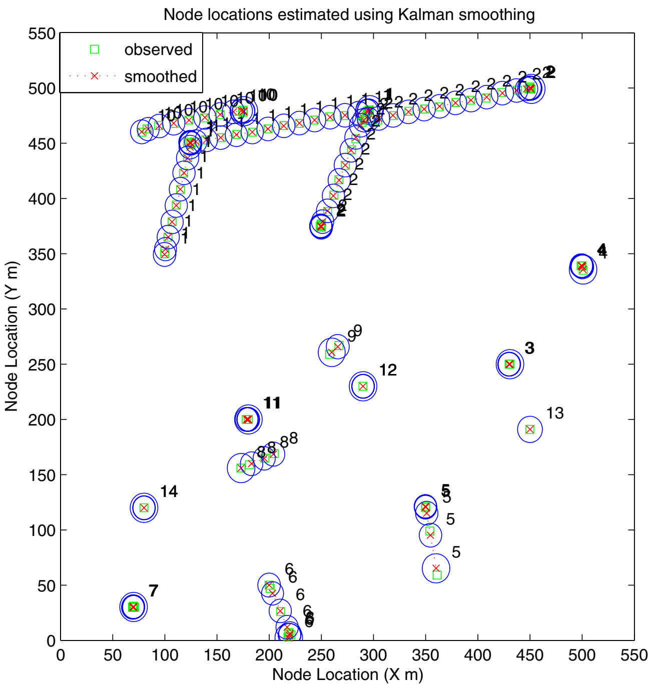


Fig. 5. Smoothed Kalman filtered node locations.

Also plotted for each smoothed location estimate is an ellipsoid that represents its 90% confidence interval. This is determined by calculating the Mahalanobis radius that encloses 90% of the probability mass. The Mahalanobis distance

$$MD = (X - \mu)^T \Sigma^{-1} (X - \mu), \tag{13}$$

is the term in the exponential of the normal distribution. It is determined by evaluating the inverse cumulative distribution function of the χ^2 distribution up to the desired confidence value.

The major and minor axes of the covariance ellipsoids are provided by the eigenvectors of the covariance matrix, and their lengths by the square roots of their respective eigenvalues. The ellipsoids provide a good representation of the uncertainty of the location estimate, and also an indication of potential node labeling errors when different node ellipses overlap. The effect of the Kalman smoothing is also evident, shown by the reduced ellipsoid diameters.

Comparing Figure 5 with the actual node movement in Figure 1, it is evident that the Kalman tracking has successfully tracked the nodes that have regular transmissions across successive time intervals. For example, nodes 1, 2, 4, 6, 7 and 10 are correctly tracked and uniquely labeled. However fourteen unique node labels have been created, when there should be ten. This is a consequence of nodes 3, 5, 8 and 9 moving a significant distance without transmitting. This results in extra unique labels being created for these nodes. Without additional data for association, it is very difficult to track this movement without transmissions, particularly if the predicted node movement vectors cross.

VI. Profiling Nodes using Template Fitting

To help with the data association problem, and to investigate the possibility of discriminating between source, sink and relay nodes, the node's transmission time distributions were analysed. This involved using the same NS-2 AODV simulation trace file with its "macTrace" function enabled. This reports all the MAC layer transmissions for the simulation and their associated transmission start times.

The node's transmission timings were taken as the differential between start times of consecutive transmissions; this is obviously only an approximation, but is shown below, to work extremely well for our template fitting.

Histograms were produced that represent the node's transmission time distributions, where the estimated timing is filtered such that any differential times that exceed 1 ms are removed. We do this to avoid the cases when there are no transmissions for a long time. The 1 ms upper limit, is selected to capture the effects of the exponential back-off implemented by the 802.11 MAC protocol. It captures both virtual and real RF collisions and the transmissions used to establish and maintain routing (AODV, ARP, etc.) The combination of these effects is plotted in Figure 6. The additional annotations on the plot represent a few selected bins, showing, the elapsed time from the start of the transmission to the next transmission, the number of times a transmission of the same length occurred, and the associated NS-2 transmission labels, in these cases MAC layer acknowledgements.

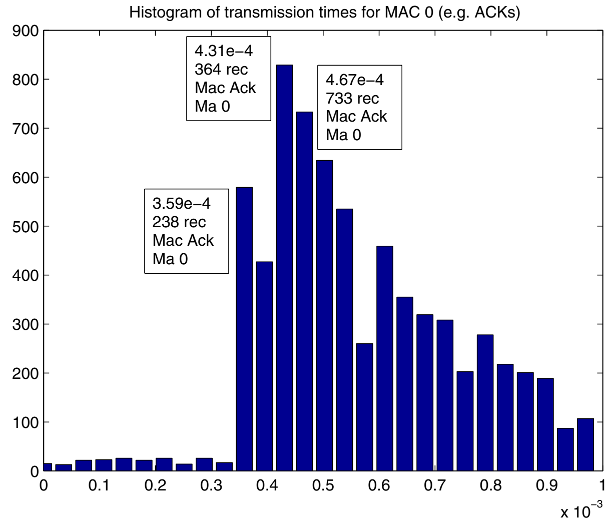
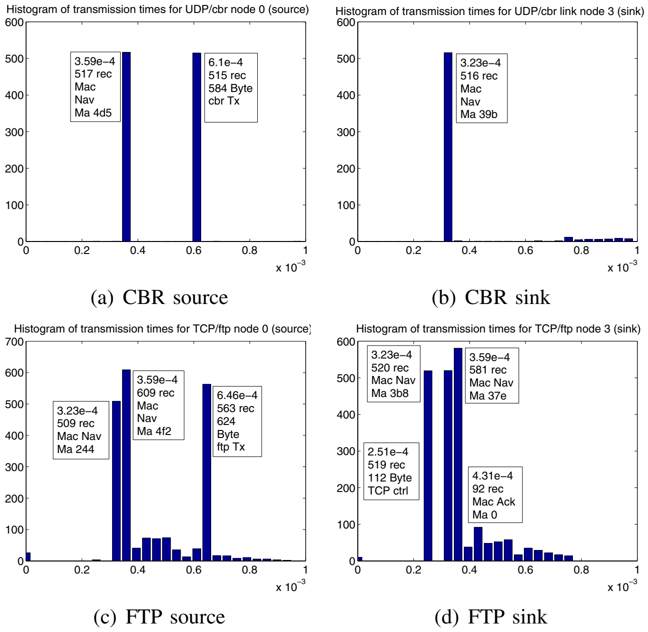


Fig. 6. Transmission time distribution for MAC.

To investigate the possibility of distinguishing between nodes acting as sources, sinks or relays a number of simulations were undertaken. Routing paths were established that simulated either a UDP, constant bit rate transfer between a source, relay and sink, or an FTP session over a TCP path in the same configuration. Figure 7 depicts the source and sink node transmission time histograms for both these scenarios.



(a) CBR source

(b) CBR sink

(c) FTP source

(d) FTP sink

Fig. 7. Histograms for node transmission times for either a single UDP or TCP link.

It clearly demonstrates that it is possible to discriminate between both protocol and transmission types. The annotations on the histograms show the packet transfers, the associated MAC, RTS/CTS and the ACK transmissions.

To automatically classify nodes, based on their transmission time distributions, a template fitting technique for node profiling was developed. Normalised templates were produced for nodes acting as sources or sinks for both of the transmission scenarios described above. Templates are not needed for relay nodes as their transmissions are simply a combination of those for sources and sinks.

Given that we have a set of measured data $y = [y_1, y_2, \ldots, y_n]^T$, for each transmitting node, we need to compare this using a maximum likelihood test, to our template set $\{T_1, T_2, \ldots, T_N\}$, so that we can assign the node a given transmission profile. The classical least squares approach [11], [12] is used to fit a linear model of the scaled template plus noise to the measured observations

$$y = x + n. \tag{14}$$

The noise component n is assumed normal, and y has a covariance matrix $\Sigma = Cov(y_p, y_q)$. The model component x is equal to the matrix H scaled by Θ, where $\Theta = [a_1, a_2, \ldots, a_N]^T$ and

$$x = H\Theta, \tag{15}$$

where,

$$H = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ t_1^{(1)} & t_1^{(2)} & \cdots & t_1^{(n)} \\ t_2^{(1)} & t_2^{(2)} & \cdots & t_2^{(n)} \\ \vdots & \vdots & \vdots & \vdots \\ t_N^{(1)} & t_N^{(2)} & \cdots & t_N^{(n)} \end{bmatrix}^T . \tag{16}$$

The first row of H comprises $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$, a vector that represents the exponential back-off implemented by the 802.11 MAC protocol already described. The template data $T = [t_N^{(1)}, t_N^{(2)}, \cdots, t_N^{(n)}]^T$ represents the values for each bin in the N template histogram set. The model for y has as before, a normal distribution whose density function in multivariate form is

$$f_\Theta = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(y-H\Theta)^T \Sigma^{-1} (y-H\Theta)}. \tag{17}$$

f_Θ is maximised when the exponential error term is minimised. Therefore for a given estimate of Θ, the weighted squared error between y and our model $H\Theta$ equals

$$e = (y - H\Theta)^T \Sigma^{-1} (y - H\Theta). \tag{18}$$

The Maximum Likelihood Estimator (MLE) for Θ is then

$$\hat{\Theta} = (H^T \Sigma^{-1} H)^{-1} H^T \Sigma^{-1} y, \tag{19}$$

which is simplified with the assumption that the covariance terms $Cov\{\Sigma_p, \Sigma_q\}$ are zero when p≠q, and thus Σ is a diagonal matrix with the terms equal to the variance of the observations $\sigma^2 = Var(y_n)$.

To this point we have generated a likelihood fit of our templates to our observation data, but we do not have a measure of confidence for the quality of the fit. The χ^2 fitting technique described in [11, page 653] is used to provide the quantitative measure we require. Our log likelihood ratio already has the χ^2 form

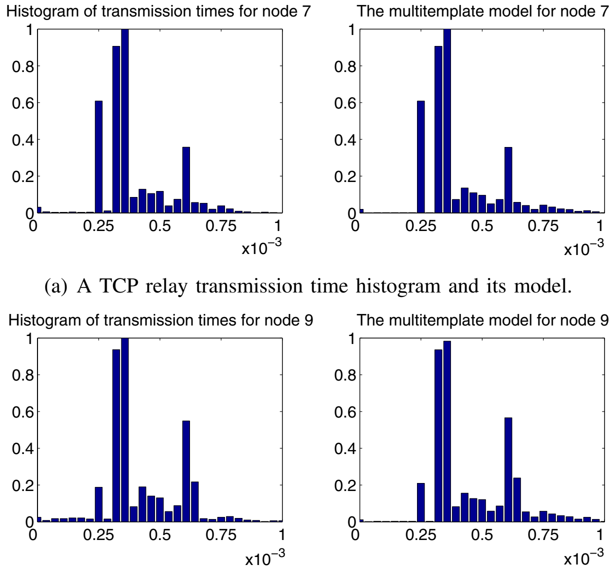
$$\chi^2 = \sum_{n=1}^{N} \frac{1}{\sigma^2} (y - H\Theta)^2. \tag{20}$$

Using this value, we can calculate the probability Q, of this occurring by chance as

$$Q = \Gamma_q \left(\frac{N}{2}, \frac{\chi^2}{2} \right), \tag{21}$$

where Γ_q is the incomplete Gamma function. To produce a confidence over a desired range, we scale the value of the variance σ_n^2 by λ. In our case, we desire a fit between 0 and 100%, so we set our probability to be $100Q$ and scale σ_n^2 appropriately.

Figure 8 shows the fitting results for two nodes, one a TCP relay and one a relay for both TCP and UDP packets. The first histogram of each set in the figure represents the actual node transmissions. The second histogram is the fitted model of the transmissions created using the five template set.



(a) A TCP relay transmission time histogram and its model.

(b) A TCP/UDP relay transmission histogram and its model.

Fig. 8. Comparison of original transmission time histograms and their associated modeled histograms.

It shows that by using only the TCP and UDP source and sink templates and the MAC RTS/CTS template, we have been able to accurately model the distributions, even when they are combinations of all the templates. Applying the χ^2 test to the noisy measured data resulted in a “Goodness of Fit” as calculated using the scaled Q, of 100% for each node. The incomplete Gamma function has a very sharp, “Step Function” characteristic and the consequence of this is that it is very difficult to get a moderate rating. This results in either a good or bad rating and as such the quality of the variance estimates for each bin is critical.

Figure 9 shows an aggregated plot of the final number of unique nodes, their logical edge connections from the inferred adjacency matrix, and their classification from the final sample of data.

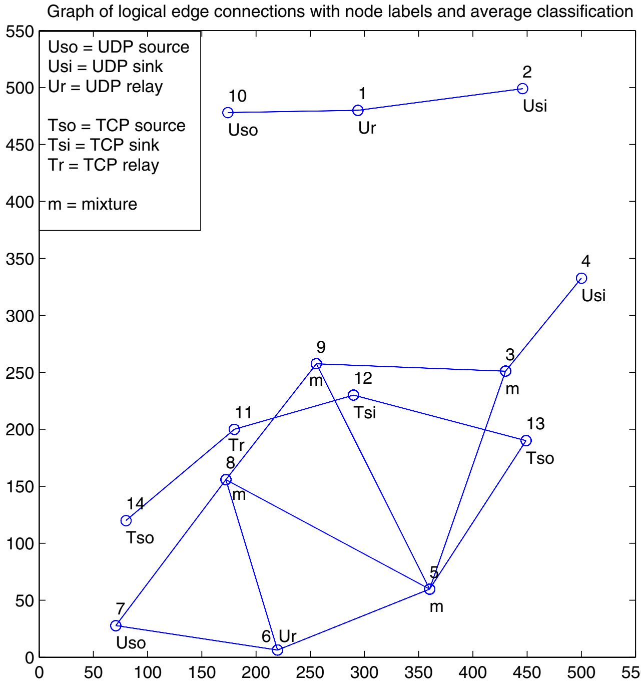


Fig. 9. Graph of unique nodes, their logical edge connections, and their average classification.

The node classifications are determined by the relative percentage fits of each template in the template set. If for example the average template fit for node 10 has a UDP source component of 85% or greater, then it is classified "Uso", a UDP source. An 85% threshold is also used for the UDP sink and the TCP source and sink templates. Nodes are classified as relays when their source and sink template fits are for the same protocol, and within ±10% of each other. For example, a fit comprising of 45% TCP source and 55% TCP sink components, is classified as a TCP relay. The mixture classification represents nodes that can't be classified into one of the other six categories.

The classification for every node in Figure 9 is correct, and the inferred logical connections between the nodes are also correct. The results do however highlight the need for further work in aggregating the data to aid in node labeling. Firstly it appears that it may be possible to improve the unique node labeling, by fusing the tracking data with the classification data. However this is not straight forward. High mobility can cause continuous routing changes which means that some nodes can only be classified as mixtures of transmission types. Also the sampling period for the transmission clustering, can greatly affect the classification. If this period is short enough, then for the majority of cases nodes are characterised correctly. If however the sampling period is chosen poorly, incorrect classifications can be made. An example of this type of error is shown by the labeling of nodes 4 and 13 which are in fact the same, node 9 from Figure 1. The node for the majority of the simulation is in fact a UDP sink. However for one very short period it acts as a TCP relay, but this occurs across a sample period and the transmission is classified as a TCP source. The classification is correct, but for only a very small sample of data. An inspection of the variance for this node's location shows that it is 100 times worse then all the other nodes, so it could be regarded as an outlier. However it does show that more work is required.

VII. Conclusion

This paper has demonstrated, that with a basic understanding of the 802.11 MAC protocol and by using received signal strength measurements, wireless probes can reconstruct a wireless network's topology. It has also been shown that nodes can be characterised using a simple template fitting approach. Only a single template is required for modeling the network and data link layers and separate templates for respective, source and sink transport layer protocols, to accurately model a transmitting node.

Future work will concentrate on fusing the collected data and developing probing schemes that can optimally reconstruct the wireless network's topology. More complex propagation models and TDOA techniques are already being simulated, to evaluate the developed techniques in less benign environments.

References

[1] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *MobiHoc*, pp. 201–212, ACM, 2003.

[2] N. Patwari, J. Costa, and A. Hero, *Locatization In Sensor Networks*, ch. Learning signal strength from sensor location and connectivity. Springer-Verlag, 2006.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Second International Conference on Knowledge Discovery and Data Mining*, (Portland, Oregon), pp. 226–231, AAAI Press, 1996.

[4] T. S. Rappaport, *Wireless Communications Principles & Practice*. New Jersey: Prentice-Hall, 1996.

[5] N. Patwari, A. O. H. III, M. Perkins, N. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Processing*, vol. 51, pp. 2137–2148, August 2003.

[6] E. W. Weisstein, "Moore-Penrose matrix inverse." From MathWorld–A Wolfram Web Resource.

[7] Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*. Academic Press, Orlando, FL USA, 1988.

[8] R. Brown and P. Y. C. Hwang, *Introduction to random signals and applied Kalman filtering*. NY: Wiley, 1997.

[9] H. E. Rauch, "Solutions to the linear smoothing problem," *IEEE Trans. Auto. Control*, vol. AC-8, p. 371, 1963.

[10] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, p. 1445, 1965.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran*. Cambridge University Press, 2nd ed., 1992.

[12] L. L. Scharf, *Statistical Signal Processing*. Addison-Wesley, 1991.