

Increasing throughput in IEEE 802.11 by optimal selection of backoff parameters

B.M.Parker

Southampton Statistical Sciences Research Institute,
University of Southampton, UK

J.Schormans

School of Electronic Engineering and Computer Sciences,
Queen Mary University of London, United Kingdom

S.G.Gilmour

Southampton Statistical Sciences Research Institute,
University of Southampton, UK

January 10, 2014

Abstract

The IEEE 802.11 standard uses Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) to avoid multiple devices simultaneous transmitting on a shared transmission medium. In this paper, Bianchi's model for IEEE 802.11 is studied and we suggest some important improvements. Firstly, we expand the state space of the Markov chain to model the evolution of a network, instead of a single device. Secondly we relax the assumption that the network must be saturated. Thirdly, we extend the model to allow for heterogeneous

devices with different transmission profiles.

We use this new model to perform Monte Carlo simulation to discover the impact of the minimum and maximum contention window times (CW_{\min} and CW_{\max}) in the standard on measures of throughput in a network. By exhaustive search over a parameter space, we find optimal values for these devices for any given network model, and show that the recommended parameters in the IEEE 802.11 standard are not optimal. We consider both average and minimum throughput, and show that increases in throughput of around 8% are possible for saturated networks, and that even greater improvements are possible for any case in which the traffic sources are not homogeneous, i.e. any real scenario.

1 Introduction

In communications networks, a Medium Access Control (MAC) policy determines how access to a single transmission medium is shared between devices. Where there is no central media access controller, and each device determines its own media access, this sharing algorithm is a distributed coordination function (DCF).

Carrier Sense Multiple Access with Collision Detection (CSMA-CD) is a DCF used in Ethernet (IEEE 802.3); if a device transmits a frame of information and detects another device transmitting at the same time, it will transmit a jam signal, and then wait for a random amount of time before

sending that frame again. This random time is determined by the truncated binary exponential backoff algorithm. and parameters determining the minimum and maximum contention window times (CW_{\min} and CW_{\max}). The device will initially backoff for a period uniformly at random in the discrete interval $[0, CW_{\min}]$. If another collision occurs, the device will backoff for a period uniformly in $[0, 2CW_{\min}]$, and so on, until we reach some maximum backoff period $[0, 2^k CW_{\min}] = [0, CW_{\max}]$.

Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) is similar to CSMA-CD, but if a device wishes to transmit it will monitor the channel and, if it finds the channel busy, waits for a random interval before retransmission, again determined by the truncated binary exponential backoff algorithm. This monitoring before transmission seeks to avoid collisions.

The IEEE 802.11 standard[16] specifies parameters CW_{\min} and CW_{\max} which control time intervals between devices retransmitting in a CSMA-CA DCF. Further parameters, notably $TXOP_{\text{limit}}$, and $AIFS$, which we do not consider here, are also specified. Recommended values of these are given in the standard, are not mandatory, and differ according to the physical medium (PHY). There are many optional extensions within the 802.11 standard, and for a more full technical description we refer the reader to, for example, Lopez Toledo et al. [20]. However, extensions to the standard have so far not been widely implemented, and the standard IEEE 802.11 DCF is still relevant.

In this paper we show how sensitive the DCF is to the values of CW_{\min} and CW_{\max} and how to choose them in order to optimize the throughput

possible in the network.

Although we have focused here on the IEEE 802.11 standard, as it is both well used and studied, the CSMA-CA DCF is used in other standards such as IEEE 802.15.4 (Wireless Personal Area Networks). See for example Campbell et al.[7] which explains the use of CSMA-CA in IEEE 802.15.4 and compares these two standards. We seek to demonstrate through reference to IEEE 802.11 the importance of the parameter choice, although the methodology could be adapted to a different standard by changing the network model we discuss below.

The Bianchi model is fundamental to our research. Bianchi ([3],[4]) models the transmission state of a device as a Markov chain. He assumes a saturated model, i.e. the device always has untransmitted packets that it will transmit when it has an opportunity. He uses this model in order to find a theoretical throughput for the CSMA/CA DCF under saturation conditions. He assumes each device acts in the same way and looks at the Markov chain from the point of view of one device to determine the behaviour of the rest of the network, a classic decomposition approach. [5] proves this assumption is valid asymptotically for a large number of states.

Ziouva and Antonakopoulos[27] extend the Bianchi model to account for busy medium conditions for invoking backoff conditions.

Banchs and Vollerero[2] consider optimizing throughput in IEEE 802.11e, a wireless standard which allows different priorities for different classes of traffic, called Access Classes. By modelling each Access Class as a different

device they then consider each device evolving as a Markov chain as per Wu et al.[26]. A set of non-linear equations are solved to give the maximum throughput for any configuration parameters; the authors present optimal settings for the IEEE 802.11e parameters CW_{\min} , CW_{\max} , and $TXOP_{\text{limit}}$, and $AIFS$.

Lopez Toledo et al.[20] ascribe the throughput sensitivity in an IEEE 802.11 network to the choice of the CW_{\min} and CW_{\max} backoff parameters, and show that saturation throughput depends only on the number of devices in the network and these backoff parameters. They develop a sequential Monte Carlo Bayesian based estimator for the number of devices, and hence find optimal values for CW_{\min} and CW_{\max} , and also show how this could work in a distributed situation. They show through simulation that this optimization method is effective.

Sharma et al. [25] do not decouple the Markov chain, focusing on the whole series of interactions for a homogeneous network. They form a Markov chain with a state vector denoting the number of stations in each backoff stage; they show that this Markov chain stays close to what they call a typical state, from which estimates of parameters (e.g. throughput) are inferred. This Markov chain is in effect a summary of the information in the Markov chain used by Bianchi and by reducing the state space the authors can add some tractability and make a performance analysis.

Kong et al.[17] analyze IEEE 802.11e performance. Traffic with higher priority is assigned lower values of CW_{\min} and CW_{\max} , with the result that

higher priority traffic tends to be transmitted sooner. Their model includes a maximum number of attempted transmissions before a packet is discarded ($TXOP_{\text{limit}}$), and other parameters specified in IEEE 802.11e (CW_{min} , CW_{max} , and $AIFS$). Simulation shows that the choice of these parameters affects the network behaviour greatly, but a general pattern or method to find optimal parameter settings is not shown. Similar work is performed analytically later by Hwang et al[15], where the effect of different parameters on an IEEE 802.11e network are deduced by an analytic model and verified, under saturated conditions.

Criticisms of the Bianchi model include the fact that it is idealized, in the sense that packets are not lost, and that real behaviour in wireless networks such as packet retransmissions may change the behaviour and remove the independence assumed. There have been various attempts to adapt the model to make the behaviour more realistic; for example Alshaynour and Agarwal[1] add an extra dimension to the standard Markov chain model. Transmission losses are assumed which give a loss probability that is constant for each packet transmitted. As well as the Bianchi backoff state and backoff counter, a state which tracks the number of data frame retransmission attempts for that device have been added. These are assumed independent between devices, and an analysis is done in a similar way to Bianchi to find throughput, and also the packet loss percentage.

All the papers above assume saturation conditions for the network, and thus also for the Bianchi Markov chain model .

Malone et al.[22] allow non-saturated models by introducing “post-backoff” states which represent devices which have transmitted a packet, but have none waiting; the time until a packet arrives is determined by a parameter, which can vary between devices. This model goes some of the way to relaxing the saturation assumption, and allowing heterogeneous devices. They show how the value of CW_{\min} is crucial to network performance for their model. However, the model is still a per-device model of the network with a Markov chain for each device, and does not account for important interactions between devices.

The hypothesis that the probability of a collision is constant is investigated in [22]. In that article (section 5) it is found that observed collisions of transmissions vary depending on the backoff stage of the device; the authors argue that this is because a packet will be retransmitted only if other devices in the network are transmitting. Thus the experimental data suggests that the probability of successful transmission depends on behaviour of other devices in the network, and interaction between devices may not be ignored in an accurate model. In this same paper, the behaviour of the Bianchi model (in saturation only) is found to significantly under-predict throughput in a model, particularly for a small number of devices, although [10] comes to a different conclusion with the same data.

Conversely, experimental or simulation work has shown that the Bianchi model is indeed a relatively good approximation, at least for saturated networks. For example, Mare et al.[23] validate the performance of the model

experimentally, albeit only for two states. Both Markov chain analysis and experimental inference become harder as the number of devices increases.

A large amount of simulation work continues to assess the performance of the IEEE 802.11 standard and extensions, and more generally CSMA-CS DCFs, for various applications; this is generally done using a simulator of a network, or sometimes a real network, and does not use the model-based approach we present here, nor evaluate the entire state space. For example, recently Hajlaoui et al.[13] study how the MAC and physical layer choice affects throughput of voice transmission in IEEE 802.11n, showing that parameter choice is important here, but do not attempt to optimise. Deng et al.[9] study the effect of collisions of packets on safety-critical VANETS (proposed Vehicular Ad Hoc Networks) through means of simulation, and show the effect of CW_{\min} and CW_{\max} parameters, amongst others, on delays; here parameter choice could really cost lives, although again a systematic search is not performed.

The “idle sense” method is proposed in [14] to dynamically change the value of the CW parameters based on network performance (the number of observed consecutive idle slots seen by a device) to ensure fairness and increased throughput; average throughput per host significantly increased using this proposed DCF under simulations.

There have been many proposed algorithms for coordination functions for MAC in wireless networks; Chen et al.[8] take a game theoretic approach to designing medium access methods and present a method that leads to a Nash

equilibrium when considering each transmitting node as a player in a game. They suggest an implementation of a medium access policy which replaces the exponential backoff procedure of CSMA/CA and replaces it with a policy whereby the channel access probability and contention window is chosen by each node in an attempt to maximize some utility function. Recent work for heavy loads includes a game theoretic approach in [12].

2 Model

As in most previous research above, we model the evolution of the idle/transmission and backoff states of a particular device as a Markov chain in slotted time.

Indeed, a discrete time Markov chain is a very good model for this process. The IEEE 802.11 DCF specifies a number of time periods (Inter Frame Periods), one of which is the “Slot time”. If a device which is in a backoff state at the start of this slot time has not detected a transmission by the end of the slot time, it will decrement its backoff timer by one, and hence become one “slot time” closer to transmission. As all devices must monitor once in this time step if they want to transmit, we can use this time step as a transition time (“clock tick”) within a discrete time Markov chain. The state of each device in any period which is equal to the slot time can therefore usefully be represented as a binary variable representing the transmission status (transmitting or not transmitting).

Most previous research using the Bianchi model assumes that the evolution of the state of a particular device is independent of the evolution of the rest of the network, and moreover implicitly assumes that the behaviour of each device in the network is identical. For example, Bianchi[4] states that “The fundamental independence assumption . . . implies that each transmission ‘sees’ the system in the same state, i.e. the steady state”. We feel that an important feature in the CSMA-CA is the interaction between sources, as demonstrated experimentally in [21] as referred to above, and we therefore crucially model the evolution of the *entire network* sharing a single wireless access point as a Markov chain, and not each device separately.

As described, an assumption in most previous work is that each device is saturated, and will always try to gain control of the medium and transmit. We assume devices are either active or inactive, corresponding to a user having data to transmit or not, and is designed to represent a pattern of user transmission and not technical constraints, such as buffering in the NIC or elsewhere. This is similar to the model for one source developed by Pitts and Shepherd[24].

We assume that the transmission state of each device is determined by an on-off source. We let the probability of a device which is not transmitting receiving data it may wish to transmit as α , and the probability of a device, which is transmitting data, finishing transmission of the current frame and going to an idle state as β . If we know the time it takes the IEEE 802.11 backoff counter to decrement (our clock tick), we can easily parametrize α and

β in terms of this (see section 2.1.1). Setting $\alpha = 1$ in our model such that the device is never idle, it becomes very similar to that of Bianchi. Note we do not claim that the on-off transmission model is a good representation of standard traffic flow, however it allows us readily to demonstrate the importance of parameter settings for the IEEE 802.11 DCF in a more typical example where networks may be unsaturated.

We also, as per explicit or implicit assumptions in previous work, assume that networks have no hidden or exposed nodes, the channel behaviour is fixed, and each node can detect traffic from all other nodes in the network. These idealised assumptions allow us to assess the behaviour of the DCF.

2.1 Optimizing CW_{\min} and CW_{\max}

For simplicity in practical implementation we restrict the CW parameters to powers of 2. We define the maximum number of backoffs determined by our parameters to be $m = \log_2 \frac{CW_{\max}}{CW_{\min}}$, and the parameters $\phi = (CW_{\min}, m)$.

We assume we have n devices $1, \dots, n$ in our network. We let device i have a state $\mathbf{Y}_i = (\mathbf{S}_i, \mathbf{B}_i)$ where \mathbf{S}_i is the transmission state of the device and \mathbf{B}_i is the backoff counter for the device. The state space for \mathbf{S}_i is $\{-1, 0, 1, 2, \dots, m\}$ where -1 means a device is idle, 0 means it is transmitting, and $1, \dots, m$ represent the m possible backoff states. The backoff counter \mathbf{B}_i takes values between -1 (idle) and CW_{\max} . We define the whole state space to be \mathbb{Y} . We represent the state of the network at time t by the vector of devices $\mathbf{Y}(t)$. Let $a(t)$ be the number of devices that are transmitting at time

t , which is equal to the number of zeroes in $\mathbf{B}(t)$; if the number of devices is greater than two, let $a(t) = 2$. We represent the transition probabilities as

$$\mathbb{P}(\mathbf{Y}(t), \mathbf{Y}(t+1)) = \prod_{i=1}^n P_{a(t)}[\mathbf{Y}_i(t), \mathbf{Y}_i(t+1)],$$

where by splitting up the transition probabilities into these three possibilities, the transition of each device from one state to the next can now be found as

$$\begin{aligned} P_0[(-1, -1), (-1, -1)] &= 1 - \alpha, \\ P_0[(-1, -1), (0, 0)] &= \alpha, \\ P_0[(i, j), (i, j-1)] &= 1, \quad i \geq 1, j \geq 1, \\ P_1[(-1, -1), (-1, -1)] &= 1 - \alpha, \\ P_1[(-1, -1), (1, j)] &= \frac{\alpha}{\phi_1}, \quad 0 \leq j \leq \phi_1 - 1, \\ P_1[(0, 0), (-1, -1)] &= \beta, \\ P_1[(0, 0), (0, 0)] &= 1 - \beta, \\ P_1[(i, 0), (0, 0)] &= 1, \quad i \geq 1, \\ P_1[(i, j), (i, j)] &= 1, \quad i \geq 1, j \geq 1, \\ P_2[(-1, -1), (-1, -1)] &= 1 - \alpha, \\ P_2[(-1, -1), (1, j)] &= \frac{\alpha}{\phi_1}, \quad 0 \leq j \leq \phi_1 - 1, \\ P_2[(i, 0), (\min(i+1, m), j)] &= \phi_1(2^{-\min(i+1, m)}), \\ & \quad 0 \leq j \leq \phi_1(2^{\min(i+1, m)}) - 1, \\ P_2[(i, j), (i, j)] &= 1, \quad i \geq 1, j \geq 1. \end{aligned}$$

We form a transition matrix M of these transition probabilities. Note that a single element of M represents a transition between a vector describing the state of all devices in the network at a given time, and a vector describing the network state in the next time period.

We are interested in the long run proportion of time that the network spends in each state, which we represent by a stationary distribution π . We write the stationary distribution as $\pi(n, \boldsymbol{\theta}, \boldsymbol{\phi})$ when we wish to emphasize the dependence of π on the number of devices, n , and the traffic pattern, which we model with our parameters $\boldsymbol{\theta} = (\alpha, \beta)$.

2.1.1 Determining on-off parameters α and β

Recall that α is the probability that a device that is idle will seek to transmit at the beginning of any time slot, and that β is the probability that a transmitting device will become idle. We can therefore represent the desired throughput, which is the percentage of time that a device would seek to transmit in isolation, as $\frac{\alpha}{\alpha + \beta}$. If the bandwidth of the network is B , we denote the average bandwidth desired by device i as b_i . We therefore write

$$b_i = B \frac{\alpha}{\alpha + \beta}, \tag{1}$$

where in this section we assume each device is homogeneous. In any time slot, the device will come to the end of the current frame and stop transmitting with probability β . The number of slots used in any transmission will

therefore be geometrically distributed with mean $\frac{1}{\beta}$.

As described clearly in Bianchi[4], each transmission of a frame is followed by a Short Inter-frame Space (SIFS), an acknowledgement from the receiving device (ACK), and then a Distributed Inter-frame Space (DIFS). We find the mean time t_i that a device engages the medium whilst transmitting as

$$t_i = \frac{\text{Mean frame size}}{B} + \text{SIFS} + \text{ACK} + \text{DIFS}. \quad (2)$$

So, given the decrement interval of the backoff counter S , our clock tick, we can write

$$t_i = \frac{1}{\beta} S \quad (3)$$

and solve the simultaneous equations (1), (2), and (3), for α and β . N.B this method of determining α and β may become invalid for supersaturated networks, i.e. if $n \frac{\alpha}{\alpha+\beta} > 1$.

2.1.2 Optimality criteria

The system is successfully able to transmit when one (and only one) device is in state 0 (i.e. $a(t) = 1$). The success of a DCF is generally measured by a function of the throughput of each device, i.e. the proportion of time that a device is able to transmit for, which we call $u(i)$. Within a given DCF, it would be possible to engineer situations in which one device always has access to the medium, and other devices have no access.

We seek to maximize some function of the throughput of each device,

$u(i)$, expressed as

$$\psi^\kappa = \begin{cases} \frac{1}{n} \sum_{i=1}^n \frac{u(i)^{1-\kappa}}{1-\kappa}, & 0 \leq \kappa \leq \infty, \kappa \neq 1; \\ \frac{1}{n} \sum_{i=1}^n \log u(i), & \kappa = 1. \end{cases}$$

This is the alpha-fair criterion described first by Kunniyur and Srikant[18].

There are two criteria which we use to demonstrate our approach:

1. The total throughput for the system, corresponding to $\kappa = 0$:

$$\psi_T(n, \theta, \phi) = \sum_{y \in \mathbb{Y}, a(y)=1} \pi(n, \theta, \phi).$$

2. The minimum throughput for any device in the system, corresponding to $\kappa \rightarrow \infty$:

$$\psi_m(n, \theta, \phi) = \min_i \left[\sum_{y \in \mathbb{Y}, a(y)=1, y_i=0} \pi(n, \theta, \phi) \right].$$

Thus we wish to find

$$\phi_t^*(n, \theta) = \operatorname{argmax}_{\phi} \psi_T(n, \theta, \phi), \text{ and}$$

$$\phi_m^*(n, \theta) = \operatorname{argmax}_{\phi} \psi_m(n, \theta, \phi),$$

where ϕ^* represents the optimal design for that criterion.

2.2 Optimization procedure

There are two methods we consider for finding the optimal value of ϕ (i.e. optimal values of CW_{\min} and CW_{\max}).

1. Analytic method: Find $\pi(n, \theta, \phi)$ explicitly in terms of ϕ , by solving the equation $\pi M = \pi$ such that $\sum_i \pi_i = 1$ and use analytic techniques to find the solution.
2. Monte Carlo: Find an approximate solution for $\pi(n, \theta, \phi)$. For each potential value of ϕ in our parameter space, iterate the Markov chain over a long number of iterations. As the chain is ergodic (recurrent and irreducible), the empirical stationary distribution found through iteration converges to the true stationary distribution.

The first method provides an analytic solution, but the state space \mathbb{Y} is very large and the solution becomes intractable for a large number of devices and parameters, and for practical values of CW_{\min} and CW_{\max} . As described in Section 1 above, previous work has focused on approximating the Markov chain by concentrating on the evolution of one device, in order to approximate the stationary distribution; although an analytic solution can be found for the simpler model, it does not guarantee accuracy for a larger number of devices if the model approximation is inaccurate. We therefore proceed with the second method for a model which we believe is more correct as it more closely mimics the true behaviour in a network.

2.3 Strengths of the model

Our on-off traffic model is flexible; it allows for saturated and unsaturated networks. Almost all study has been of theoretical performance of saturated networks, whereas in reality wireless networks are often unsaturated. Setting $\alpha = 1$ however allows us to recover the Bianchi model. Moreover, we can generalize our model to heterogeneous behaviour of users, something not readily available in the Bianchi model.

The model of network evolution attempts to find a model which is parsimonious; we believe the model allows for a rapid simulation to find throughput (or other fairness criterion, such as minimum throughput) whilst preserving important features of the DCF (collisions). The Markov nature of the model leads to a rapid (Monte Carlo) simulation, with the result that optimal value for DCF parameters can be found. We do not claim that the results are more accurate or precise than a full simulation or experiment with real devices, however they allow us to quickly assess the performance of the system over the parameter space, which may allow practitioners to focus on which hardware experiments to run. In short, our model provides a balance between a complex experiment and a simpler model which may not describe the system performance well.

3 Homogeneous traffic results

We use an IEEE 802.11g network as our example, but the general principle will apply to any network which uses the CSMA-CA protocol. We assume we have a network which has a maximum bandwidth $B = 54Mbps$, a slot time $S = 9\mu s$, SIFS = $10\mu s$, ACK = $2\mu s$ and DIFS = $28\mu s$. Let us assume our mean packet size, including header and FCS, is 1000 bytes. Then from equation (2), we calculate the mean time that a device engages the transmission medium as $t_i = \frac{1000 \times 8}{54 \times 10^6} + 10 \times 10^{-6} + 2 \times 10^{-6} + 24 \times 10^{-6} = 188 \times 10^{-6} \approx 2 \times 10^{-4}s$.

We initially assume devices seek to transmit at 1.5Mbps, or 10% of the total available bandwidth, such that $b_i = 0.1$ for all i . Using $t_i = 2 \times 10^{-4}$ as derived above, we solve equations (1) and (3) to get $\alpha = 0.005$ and $\beta = 0.045$.

We used the second optimization procedure described in section 2.2, and performed an exhaustive search over the parameter space $\phi = \{(CW_{\min}, m)\}$, where $\log_2(CW_{\min}) \in \{1, 2, \dots, 10\}$, and $m \in \{1, 2, \dots, 10\}$, such that there were 100 candidate points for ϕ .

3.1 Assessing convergence

When simulating, we must determine how many iterations of the chain we need. We want our estimate, $\hat{\psi}$ of the throughput to be sufficiently close to the true value ψ to enable us to determine the optimum value of the unknown parameters ϕ ; i.e. when is $|\hat{\psi} - \psi| < \epsilon$ for some tolerance ϵ ,

or equivalently, when is $\text{Var}(\hat{\psi})$ small enough that we are confident in our results. (As the Markov chain is irreducible and aperiodic (ergodic), we know that the stationary distribution exists, and as ψ is a function of the stationary distribution, the variance of $\text{Var}(\hat{\psi}) \rightarrow 0$ asymptotically.) The number of iterations to reach convergence is known as the mixing time of the Markov chain, and can be difficult to work out in general, and specifically for the complicated Markov chain we have here. (See [19] for some background.)

Some techniques for assessing whether a Monte Carlo algorithm simulating a Markov chain has converged are presented in [6]. [11] suggests dividing our simulation into blocks each consisting of 100 iterations, and forming an estimate for ψ_T and ψ_m after each block; we consider only the last 100 blocks seen, and call the estimates found after the k -th block (after $100k$ iterations) $\hat{\psi}_T^{(k)}$ and $\hat{\psi}_m^{(k)}$. In order to test whether the chain has converged, we compare the two sub-sequences $\hat{\psi}^{(1)}, \dots, \hat{\psi}^{(10)}$ and $\hat{\psi}^{(51)}, \dots, \hat{\psi}^{(100)}$; we form our estimates $\hat{\psi}_A = \sum_{k=1}^{10} \frac{1}{10} \hat{\psi}^{(k)}$ and $\hat{\psi}_B = \sum_{k=51}^{100} \frac{1}{50} \hat{\psi}^{(k)}$, and calculate the sample autocovariances which we call S_A^2 and S_B^2 respectively for the two sub-sequences. We form the statistic $Z = \frac{\hat{\psi}_A - \hat{\psi}_B}{\sqrt{0.1S_A^2 + 0.5S_B^2}}$, and note that $Z \xrightarrow{D} N(0, 1)$ as $n \rightarrow \infty$.

More heuristically, as we perform more iterations the throughput vs parameters mesh plots seen in Figures 1 to 4 below get smoother. We can visually assess when the graphs are smooth enough to find useful results.

In practice we find a moderate number of iterations ($\approx 10^7$) are sufficient to make a good estimate of π , and find parameters which increase throughput

substantially as compared to the recommended values given in the standard.

3.2 Numerical results

For each candidate point ϕ , we performed 5×10^7 Monte Carlo iterations, and present a mesh plot of throughput

$$\psi_T(n, (0.005, 0.045)^T, \phi) = \sum_{y \in \mathbb{Y}, a(y)=1} \pi(n, \theta, \phi).$$

We present the results for selected numbers of devices as Figure 1, although plots for other numbers of devices are similar in feature. Each graph shows the raw throughput for each device (expressed as a percentage of available bandwidth) against phi_1 (x-axis), and phi_2 (y-axis), our parameterised values for CW_{\min} (x-axis), and CW_{\max} (y-axis) . By finding the maximum value of each graph, we can find the optimal values of parameters for that number of devices. We note some general features of the graph:

- The slight lack of smoothness in the graph is due to the stochastic nature of the Markov chain simulation, and the amount of jaggedness is proportional to the variance of the simulated process. Increasing the number of simulations beyond 5×10^7 can reduce this jaggedness further. The $n = 1$ graph of course has no conflict, so lets us get an idea of the (very small) variation caused by the use of Monte Carlo.
- A very high numbers of allowed backoffs generally reduces the through-

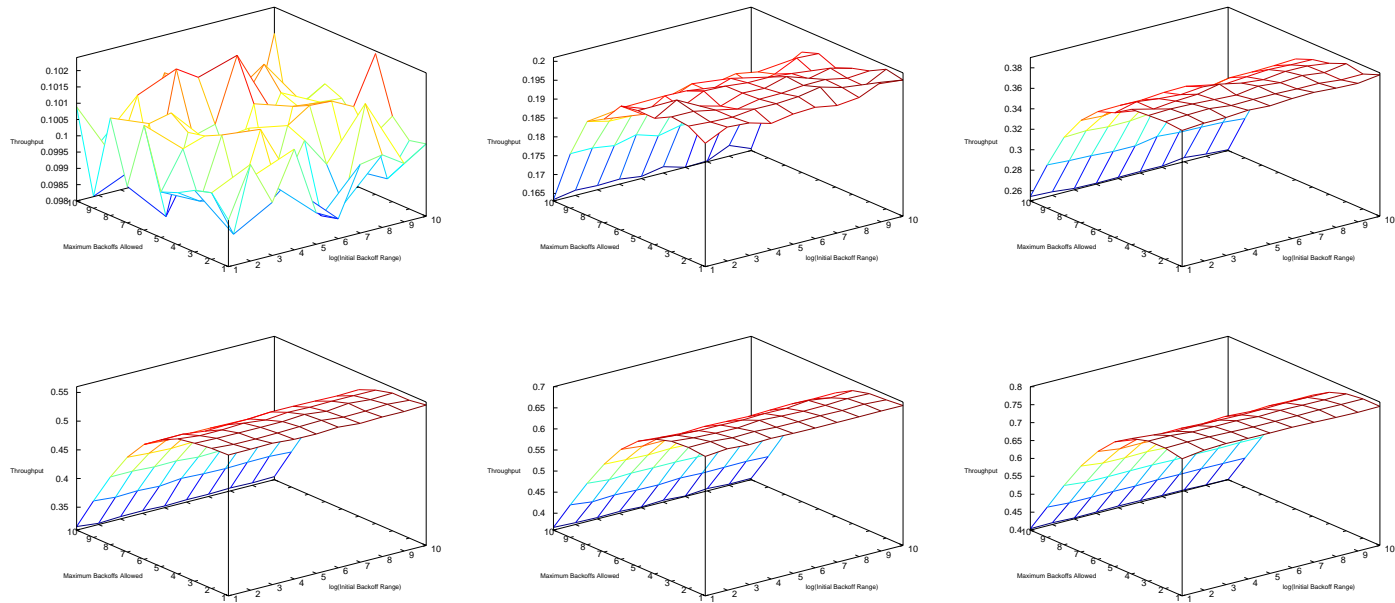


Figure 1: Raw throughput (z-axis) against varying CW_{\min} (x-axis), and CW_{\max} (y-axis). From top left, top middle, etc., to bottom right the number of devices is: 1,2,4,6,8,10

put considerably; this is because it is possible for one device which has data to transmit to have a large value of backoff counter, and be required to wait a long time even when the medium is available.

- A smaller number of backoffs seems in general to be optimal for maximizing throughput. In other words, apart from pathological cases, the ratio CW_{\max}/CW_{\min} is important in optimising throughput, and the absolute value of CW_{\min} is less important. Typically setting the maximum number of backoffs $CW_{\max}/CW_{\min} = 2$ is near optimal.
- Small values of both CW_{\min} and CW_{\max} give slightly worse results as more conflicts occur.

The recommended values for CW_{\min} and CW_{\max} in the IEEE 802.11 standard[16], which vary depending on the transition medium (PHY) used. Most mediums suggest setting CW_{\min} at either 15 or 31, and agree on $CW_{\max} = 1023$. This corresponds to $\phi = (4, 6)$ or $\phi = (5, 5)$ in our parametrization. If these values of ϕ were optimal, we would expect the highest values of throughput to be found here, however our results suggest that the optimum is not found at these levels.

For example, in Figure 1, the throughput under recommended values for the DSSS PHY in the ISM band recommends $CW_{\min} = 31$ and $CW_{\max} = 1023$, corresponding to $\phi = (5, 5)$. Our Markov chain simulation shows that this yields a raw throughput of 73.7%. If we use parameters $CW_{\min} = 2$ and $CW_{\max} = 8$, corresponding to $\phi = (1, 3)$ this would provide a throughput

Table 1: Percentage increase in raw throughput by choice of parameters

No. of devices	Throughput at $\phi = (5, 5)$	Max. Throughput	ϕ at max.	% increase in throughput at optimal ϕ
2	0.19759	0.20018	(1,9)	1.3103
3	0.28989	0.29511	(2,2)	1.7986
4	0.37553	0.38800	(1,10)	3.3216
5	0.45576	0.47476	(1,6)	4.1683
6	0.52839	0.55608	(1,6)	5.2401
7	0.59282	0.63168	(1,4)	6.5542
8	0.65104	0.69426	(2,1)	6.6383
9	0.69777	0.75035	(2,2)	7.5357
10	0.73652	0.79345	(1,3)	7.7297

of 79.3%. This corresponds to a net increase in throughput of 7.73%.

Table 1 shows the increase in throughput by choosing parameters for between two and ten devices. The benefit of choosing parameters for a larger number of devices is more pronounced, as there is less unused bandwidth and resolving conflicts well becomes more important.

3.3 Minimum throughput

We now consider maximizing the minimum throughput ψ_m of each transmitting device, and repeat the same procedure for the same candidate points for ϕ to find our optimal value of CW_{\min} and CW_{\max} .

We again plot the throughput $\psi_m(n, (0.005, 0.045)^T, \phi)$ for each potential value in our parameter space, although in this figure we plot the minimum throughput. These are displayed as Figure 2, again for selected n . The graphs are slightly less smooth after 5×10^7 iterations compared to those for

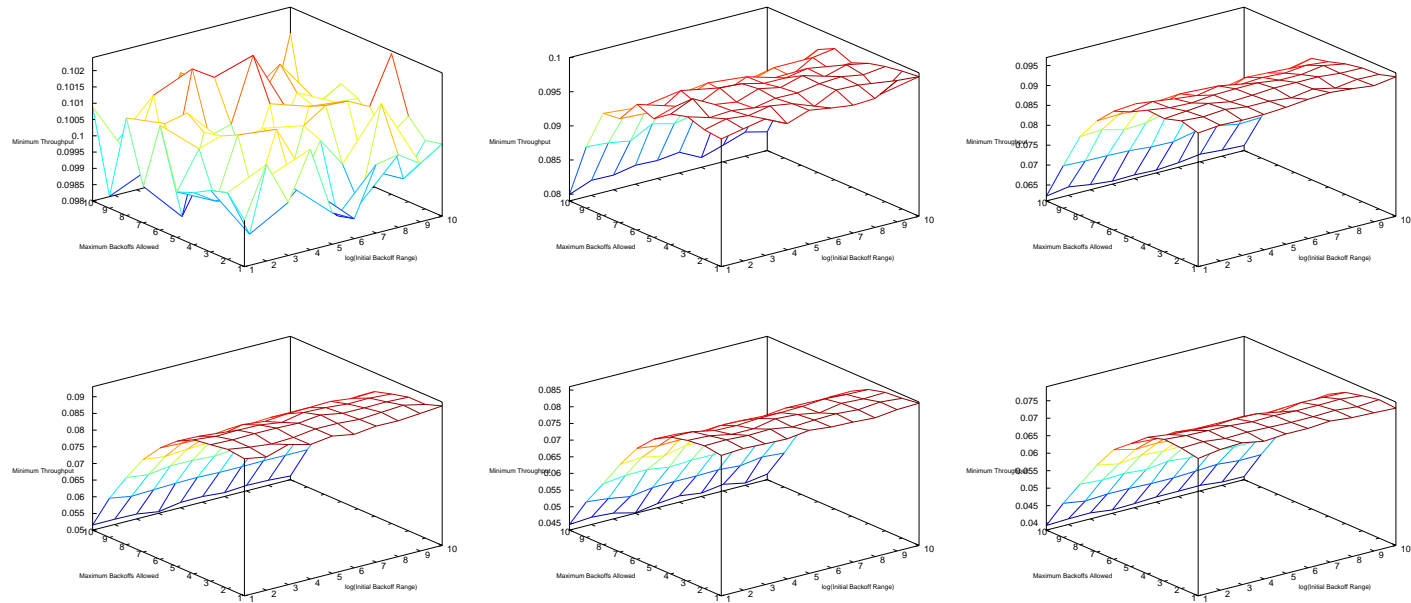


Figure 2: Minimum throughput (z-axis) against varying CW_{\min} (x-axis), and CW_{\max} (y-axis). From top left, top middle, etc., to bottom right the number of devices is: 1,2,4,6,8,10

Table 2: Percentage increase in minimum throughput by choice of parameters

No. of devices	Min. Throughput at $\phi = (5, 5)$	Max (min Throughput)	ϕ at max.	% increase in min. throughput at optimal ϕ
2	0.097989	0.099507	(4,2)	1.5499
3	0.096156	0.098193	(2,2)	2.1178
4	0.093403	0.096503	(1,7)	3.3185
5	0.089447	0.094466	(1,6)	5.6108
6	0.086831	0.092151	(1,10)	6.1272
7	0.08404	0.089405	(1,4)	6.3838
8	0.080182	0.086114	(2,2)	7.399
9	0.076725	0.082357	(2,3)	7.3406
10	0.072465	0.078708	(1,3)	8.6156

the raw throughput. This is because the variance of the minimum function (corresponding to $\kappa \rightarrow \infty$) is higher than that of the mean function ($\kappa = 0$), so we need more Monte Carlo iterations to get the same smoothness. Nevertheless, the graph clearly demonstrates that the recommended standard values corresponding to $\phi = (4, 6)$ or $\phi = (5, 5)$ in our parametrization are sub-optimal, and suggests that values around $\phi = (2, 2)$ would generally yield greater minimum throughput.

Table 2 shows the increase in the minimum throughput for any device by choosing parameters, for up to ten devices. Again we see that the minimum throughput can be increased more by choosing backoff parameters for a larger number of devices; for example the increase for choosing parameters with two devices is 1.55%, but with 10 devices is 8.62%.

4 Heterogeneous users

Up to now we have assumed that the behaviour of the users is homogeneous, i.e. that the traffic that each user has to transmit is governed by the same on-off probabilities θ . In reality, this assumption is not true; different users have different transmission patterns.

Consider a “household” model of IEEE 802.11, a typical small scale implementation where a small collection of users in a household engage in different activities using a wireless network; the standard assumption of homogeneous users is not appropriate, and we investigate in this section how the backoff parameters affect the throughput of the household.

We now extend our model such that the on state and off state of each user in the model is modelled by a vector $\theta = (\alpha, \beta)$, where now α_i is the probability of going from an idle state to a state where device i has a frame to transmit, and β_i is the probability of going from a state where device i is transmitting to an idle state.

We maintain our example of an IEEE 802.11g network as above, with the same parameters, and we consider a scenario where we have three types of users. User 1 is making long file transfers, and we let $\alpha_1 = 0.0025$ and $\beta_1 = 0.0225$. User 2 is engaged in a VoIP conversation, with many short exchanges of packets, so $\alpha_2 = 0.01$ and $\beta_2 = 0.09$. We let any other users experience an intermediate transfer length with $\alpha_i = 0.005$ and $\beta_i = 0.05, i \geq 3$ as before.

We plot our throughput graph again for the heterogeneous devices as

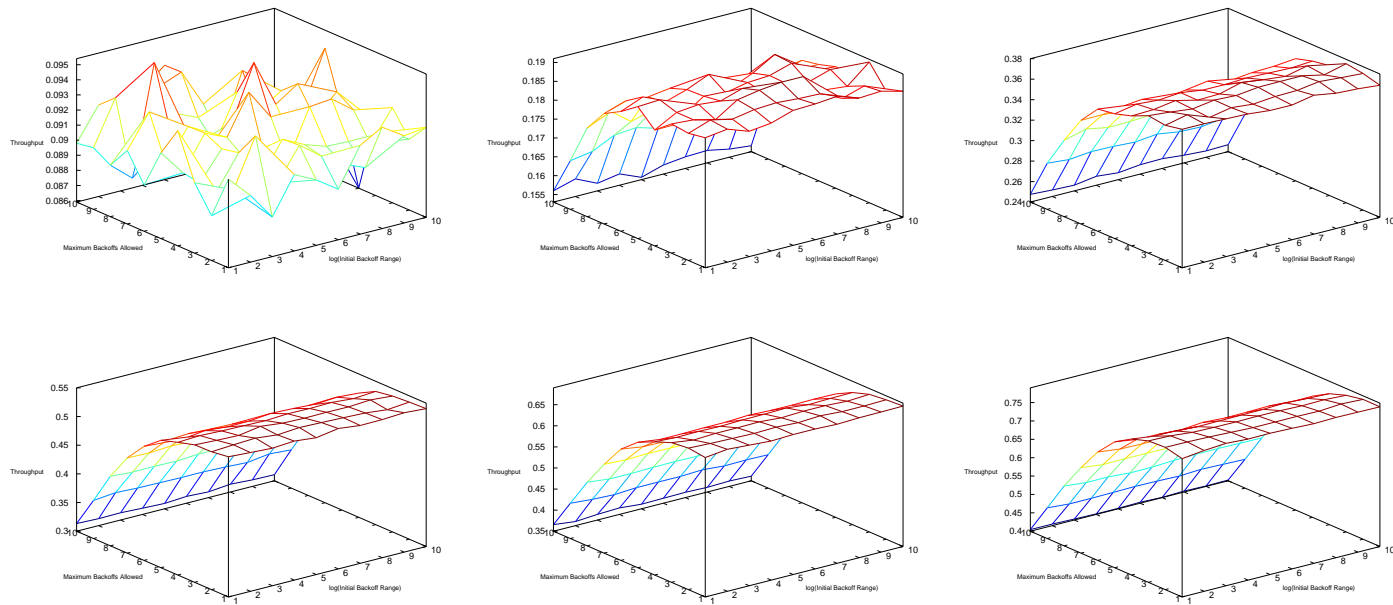


Figure 3: Raw throughput for heterogeneous devices (z-axis) against different parameters CW_{\min} (x-axis), and CW_{\max} (y-axis). From top left, top middle, etc., to bottom right the number of devices is: 1,2,4,6,8,10

Table 3: Percentage increase in throughput for heterogeneous devices by choice of parameters

No. of devices	Throughput at $\phi = (5, 5)$	Max. Throughput	ϕ at maximum	% increase in throughput at optimal ϕ
2	0.18510	0.19019	(3,6)	2.7517
3	0.27835	0.28680	(1,7)	3.0380
4	0.36437	0.37603	(2,8)	3.1990
5	0.44396	0.46286	(1,6)	4.2570
6	0.51642	0.54509	(1,6)	5.5523
7	0.58497	0.61965	(2,2)	5.9278
8	0.64141	0.68636	(1,2)	7.0079
9	0.68957	0.74226	(2,2)	7.6406
10	0.73221	0.78753	(2,1)	7.5558

Figure 3. We see a similar pattern as to that in the homogeneous case.

We display the optimal values for different number of devices as Table 3. Comparing to Table 1, we see that the increased percentage throughput in choosing the backoff parameters is similar in this heterogeneous case.

For heterogeneous devices, our minimum throughput criterion is no longer particularly useful, as the minimum throughput will almost always correspond to the device which has the least traffic to transmit. We can therefore replace our minimum throughput criterion ψ_m by

$$\psi_{\text{het}}(n, \theta, \phi) = \min_i \left[\sum_{y \in \mathbb{Y}, a(y)=1, y_i=0} \pi(n, \theta, \phi) \left(\frac{\alpha_i}{\alpha_i + \beta_i} \right)^{-1} \right],$$

a scaled minimum throughput criterion, where the throughput of each device is divided by $\frac{\alpha_i}{\alpha_i + \beta_i}$, the proportion of time it would seek to transmit

Table 4: Percentage increase in scaled minimum throughput for heterogeneous devices by choice of parameters

No. of devices	Min Throughput at $\phi = (5, 5)$	Max (min Throughput)	ϕ at max.	% increase in min throughput at optimal ϕ
2	0.88781	0.92903	(3,6)	4.643
3	0.88290	0.93306	(1,7)	5.6816
4	0.87975	0.91699	(1,7)	4.232
5	0.84873	0.89517	(1,4)	5.4711
6	0.80158	0.86751	(2,3)	8.2251
7	0.74843	0.83049	(1,5)	10.964
8	0.69454	0.78101	(2,6)	12.449
9	0.65174	0.72900	(1,4)	11.853
10	0.59564	0.67385	(2,9)	13.132

were it to be able to act independently in the network without any other devices being present. The result is that the absolute value of the scaled minimum throughput becomes difficult to interpret, but the criterion is fair in that high values of the criterion occur when each device is able to transmit proportionately to its desired traffic flow. The scaled minimum throughput criterion for this example is shown in Figure 4.

The increase in this scaled minimum throughput for the heterogeneous devices is shown in Table 4. Again, we see that choice of backoff parameters has a larger effect in the heterogeneous environment for this criterion than in the homogeneous case, an effect ignored in previous research.

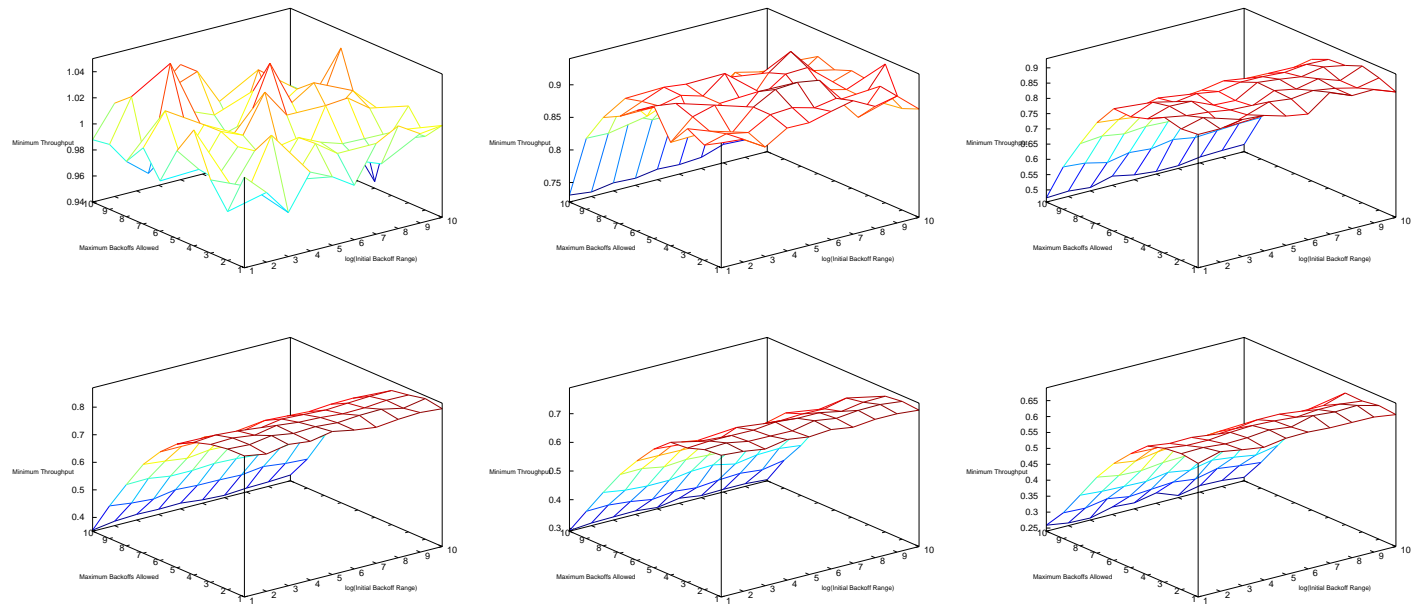


Figure 4: Scaled minimum throughput for heterogeneous devices (z-axis) against varying CW_{\min} (x-axis), and CW_{\max} (y-axis). From top left, top middle, etc., to bottom right the number of devices is: 1,2,4,6,8,10

5 Conclusions

In this paper we have shown how to build a Markov chain model to represent the evolution of a system using the CSMA-CA protocol, using the IEEE 802.11 standard as an example. We have extended models in previous research to model the evolution of the whole network as a Markov chain. This has advantages as the chain now models interactions between devices; as the standard specifies a slot time during which a transmitting device will not be interrupted, and the system can be thought of as evolving in slotted time, the model is a fairly faithful representation of the standard.

We have shown that a Monte Carlo approach allows us to find the stationary distribution of the model, and hence estimate the throughput for any given model of a network. The variance of this estimate can be made arbitrarily small given a large enough number of iterations, and we have shown that the convergence is good even for a relatively small number of iterations.

We have extended previous research to include non-saturated scenarios, and heterogeneous traffic mixes, all within the same Markov chain model.

This modelling has allowed us to assess the parameters CW_{\min} and CW_{\max} which determine backoffs within CSMA-CA. For the networks we have studied here, we show that the values recommended by the IEEE 802.11 standard are not optimal for throughput, or min-throughput, two commonly used optimality criteria, and we find optimal values. Having found optimal parameter settings the throughput is improved, especially in any non-

homogeneous case. This is particularly important as almost all previous analyses have focussed on homogeneous cases whereas reality is always non-homogeneous.

References

- [1] A. Alshanyour and A. Agarwal. Three-dimensional Markov chain model for performance analysis of the IEEE 802.11 distributed coordination function. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–7, 2010.
- [2] A. Banchs and L. Vulliamy. Throughput analysis and optimal configuration of 802.11 e EDCA. *Computer Networks*, 50(11):1749–1768, 2006.
- [3] G. Bianchi. IEEE 802.11-saturation throughput analysis. *IEEE Communications Letters*, 2(12):318–320, 1998.
- [4] G. Bianchi. Performance analysis of the IEEE 802. 11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.
- [5] C. Bordenave, D. McDonald, and A. Proutire. Random multi-access algorithms - a mean field analysis. 2005.

- [6] S. P. Brooks and G. O. Roberts. Assessing convergence of Markov chain Monte Carlo algorithms. *STATISTICS AND COMPUTING*, 8:319–335, 1997.
- [7] C. E. Campbell, K. Loo, H. A. Kurdi, and S. Khan. Comparison of iee 802.11 and iee 802.15. 4 for future green multichannel multi-radio wireless sensor networks. *International Journal of Communication Networks and Information Security (IJCNIS)*, 3(1), 2011.
- [8] L. Chen, S. H. Low, and J. C. Doyle. Random access game and medium access control design. *IEEE/ACM Transactions on Networking*, 18(4):1303–1316, 2010.
- [9] D.-J. Deng, H.-C. Chen, H.-C. Chao, and Y.-M. Huang. A collision alleviation scheme for iee 802.11 p vanets. *Wireless Personal Communications*, 56(3):371–383, 2011.
- [10] K. Duffy. Mean field markov models of wireless local area networks. *Markov Processes and Related Fields*, 2009.
- [11] J. Geweke and F. R. B. o. Minneapolis. *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Federal Reserve Bank of Minneapolis, Research Dept., 1991.
- [12] M. Ghazvini, N. Movahedinia, and K. Jamshidi. A game theory based contention window adjustment for iee 802.11 under heavy load. *Inter-*

national Journal of Communication Networks & Information Security, 5(2), 2013.

- [13] N. Hajlaoui, I. Jabri, and M. B. Jemaa. Experimental performance evaluation and frame aggregation enhancement in IEEE 802.11 n WLANs. *International Journal of Communication Networks and Information Security (IJCNIS)*, 5(1), 2013.
- [14] M. Heusse, F. Rousseau, R. Guillier, and A. Duda. Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless LANs. *ACM SIGCOMM Computer Communication Review*, 35(4):121132, 2005.
- [15] B.-J. Hwang, I.-S. Hwang, and H.-H. Chang. Saturation performance assessment of differentiation services supported in IEEE 802.11 e. *Journal of Internet Technology*, 12(6):855–864, 2011.
- [16] IEEE Computer Society. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Standard for Information Technology*, 802, 2007.
- [17] Z. Kong, D. Tsang, B. Bensaou, and D. Gao. Performance analysis of IEEE 802.11e contention-based channel access. *IEEE Journal on Selected Areas in Communications*, 22(10):2095, 2004.

- [18] S. Kunniyur and R. Srikant. End-to-end congestion control schemes: Utility functions, random losses and ecn marks. In *In Proceedings of IEEE Infocom*, pages 1323–1332, 2000.
- [19] D. Levin, Y. Peres, and E. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2009.
- [20] A. Lopez Toledo, T. Vercauteren, and X. Wang. Adaptive optimization of IEEE 802.11 DCF based on Bayesian estimation of the number of competing terminals. *IEEE Transactions on Mobile Computing*, pages 1283–1296, 2006.
- [21] D. Malone, I. Dangerfield, and D. Leith. Verification of common 802.11 MAC model assumptions. *Passive and Active Network Measurement*, page 6372, 2007.
- [22] D. Malone, K. Duffy, and D. Leith. Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions. *IEEE/ACM Transactions on Networking*, 15(1):159–172, 2007.
- [23] S. Mare, D. Kotz, and A. Kumar. Experimental validation of analytical performance models for IEEE 802.11 networks. In *Communication Systems and Networks (COMSNETS)*,, pages 1–8, 2010.
- [24] J. M. Pitts and O. M. Shepherd. Analysing the transition between unsaturated and saturated operating conditions in 802.11 network scenarios.

In *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pages 1–7, San Diego, CA, USA, 2008.

- [25] G. Sharma, A. Ganesh, and P. Key. Performance analysis of contention based medium access control protocols. *Information Theory, IEEE Transactions on*, 55(4):1665–1682, 2009.
- [26] H. Wu, S. Cheng, Y. Peng, K. Long, and J. Ma. IEEE 802.11 distributed coordination function (DCF): analysis and enhancement. In *IEEE International Conference on Communications*, volume 1, pages 605–609, 2002.
- [27] E. Ziouva and T. Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3):313–321, 2002.

Acknowledgements

This work was supported by EPSRC Grant EP/G012628/1: “Optimal design of performance measurement experiments for complex, large-scale networks” at Queen Mary University of London.