Modeling the Effect of Transmission Errors on TCP Controlled Transfers over Infrastructure 802.11 Wireless LANs

Subhashini Krishnasamy
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore
subhashini.kb@ece.iisc.ernet.in

Anurag Kumar
Dept. of Electrical Communication Engg.
Indian Institute of Science, Bangalore
anurag@ece.iisc.ernet.in

ABSTRACT

There have been several studies on the performance of TCP controlled transfers over an infrastructure IEEE 802.11 WLAN, assuming perfect channel conditions. In this paper, we develop an analytical model for the throughput of TCP controlled file transfers over the IEEE 802.11 DCF with different packet error probabilities for the stations, accounting for the effect of packet drops on the TCP window. Our analysis proceeds by combining two models: one is an extension of the usual TCP-over-DCF model for an infrastructure WLAN, where the throughput of a station depends on the probability that the head-of-the-line packet at the Access Point belongs to that station; the second is a model for the TCP window process for connections with different drop probabilities. Iterative calculations between these models yields the head-of-the-line probabilities, and then, performance measures such as the throughputs and packet failure probabilities can be derived. We find that, due to MAC layer retransmissions, packet losses are rare even with high channel error probabilities and the stations obtain fair throughputs even when some of them have packet error probabilities as high as 0.1 or 0.2. For some restricted settings we are also able to model tail-drop loss at the AP. Although involving many approximations, the model captures the system behavior quite accurately, as compared with simulations.

Categories and Subject Descriptors

C.2.5 [**Computer Communication Networks**]: Local and Wide Area Networks - *Access schemes*

General Terms

Performance

Keywords

WLANs, performance analysis, modeling TCP over 802.11 CSMA/CA, channel errors

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.

1. INTRODUCTION

IEEE 802.11 supports multiple data transmission rates with different modulation schemes at the PHY layer. The aim of rate adaptation algorithms is to exploit this multi-rate capability to obtain the best throughput in different channel conditions. Some recent rate adaptation algorithms ([2, 17]) estimate the packet error rate periodically and adjust the transmission rates by comparing the error rate with some thresholds. Others ([6, 1]) use a combination of factors such as estimates of the throughput and, the probability of success, to adapt the rates. A model that predicts the throughputs obtained at different data transmission rates and different packet error rates would be very useful in providing insights in setting the threshold parameters in [2, 17] or in deciding the transmission rates and the retry counts in the retry chain in [6, 1]. We address the problem of estimating the throughputs obtained by the stations when each of them download TCP files at a constant PHY rate and experience some packet error rate on the wireless channel.

We consider an IEEE 802.11 infrastructure WLAN with a single Access Point (AP). The stations (STAs) can experience different channel conditions and bit-error-rates depending on their locations, the RF propagation environment, the external interference, and the PHY with which they are associated. We consider the problem of modeling TCP controlled file downloads from a server on the wired LAN. We focus on downloads, as studies have shown that a large fraction of wireless access traffic tends to be downlink in nature.

Related Literature.

There is a vast literature on performance modeling of the MAC protocol in IEEE 802.11 WLANs under various traffic scenarios. There have been many models ([25, 15]) which use Bianchi's fixed point analysis [5] of saturated nodes in a single-cell network. Bruno *et al.* [8] studied TCP file downloads by modeling the interaction of TCP with a p-persistent IEEE 802.11 MAC protocol by using Markov chain analysis for the number of stations that are contending for the wireless channel. In [16], TCP file downloads over the standard IEEE 802.11 DCF was modeled by using the saturation analysis ([5, 15]) assuming that the AP buffer is infinite. However, in all these papers, the authors assume ideal channel conditions and that frames can be received in error only due to collision. Some researchers have recently extended the 802.11 saturation analysis to include channel errors [9, 12, 20]. Chatzimisios *et al.* [9] perform saturation analysis assuming that all stations are associated at the same rate and experience the same BER. Multi-rate and different BER for each station with saturated queues is considered in [12, 20].

On the other hand, there have been many models which analyze the performance of TCP over wireless links by considering random packet losses [18, 14]. However, these papers consider point-to-point wireless links. Mare *et al.* [21] analyze the effect of channel errors in an IEEE 802.11 WLAN for the simple case of single TCP long file transfer. To the best of our knowledge, the interaction between packet losses, CSMA/CA random access and TCP has not been analyzed in a multi-station infrastructure WLAN.

Contributions of this paper.

We study the impact of channel errors on TCP performance in a more general framework where there are multiple STAs with different channel conditions and bit-error rates, downloading long TCP files from the wireline LAN. It is known that, without channel errors, all TCP controlled file transfers obtain equal throughputs [8, 16]. With different channel error rates, however, the TCP window processes of the connections are affected differently, and it is expected that there would be throughput unfairness and also a reduction in the total network throughput. We provide an analytical model that yields insights into these performance issues.

Our analysis proceeds by iterating between two sub-models. One is an extension of the Markov regenerative model in [8, 16] that provides the STAs' throughputs given the probability that the head-of-the-line (HOL) packet at the AP belongs to a particular STA. These HOL probabilities are then obtained from the analysis of the TCP window process by focusing on the sharing of the AP buffer by the various TCP connections. In modeling the congestion window processes, we initially assume that the AP buffer is infinite and, therefore, packet drops are only due to channel errors and not due to buffer overflow. We then extend this model to the finite buffer case. Even though many approximations are made during the course of our analysis, we find that we are able to capture the essential factors that govern the performance, and the results from our analysis compare very favorably with the simulations. It is observed that MAC layer retransmissions help in avoiding critical throughput unfairness up to fairly large error probabilities (almost 10%) but with a slight decrease in the total network throughput. But at higher error rates, STAs with higher packet loss probabilities get lower throughput due to two reasons: a) Inability of TCP to recognize losses due to channel errors, thus, causing a reduction in the windows of the STAs with higher loss probabilities, and b) FIFO queuing at the AP buffer.

2. THE SETTING AND MODELING ASSUMPTIONS

We consider an IEEE 802.11 WLAN in which the stations (STAs) are associated with a single Access Point (AP). Each STA is downloading a large file from a "server" which is on the high-speed LAN to which the AP is connected. The STAs may experience different channel conditions and therefore, are subject to different bit error rates (BER), yielding different packet error rates (PER).

We briefly describe how DCF of the 802.11 MAC protocol works: For every packet, a node begins with a uniformly distributed random backoff of mean b_0 slots (b_0 = CWmin/2) and defers the backoff when it senses a busy medium. At the end of the backoff, if the channel is idle for DIFS time, it transmits the packet. If the transmission was unsuccessful, it retries the frame with a random backoff of increased mean. We denote the mean backoff duration (in slots) after the k^{th} attempt for a frame by b_k ($b_k = \min(2^k \times b_0$, CWmax/2)). The node tries to retransmit the frame until it is successful or the maximum retry limit is reached. The maximum possible number of attempts for an RTS or for a short frame whose size is less than or equal to the RTS threshold, is called the dot11ShortRetryLimit, denoted by K_s. For a long frame of length greater than the RTS threshold, the maximum possible number of attempts is called the dot11LongRetryLimit, denoted by K_l. After the number of attempts for a frame reaches its maximum (K_s or K_l according to the size of the frame), the packet is discarded [3]. Table 1 shows the various parameters used in the analysis and simulation.

Parameter	Symbol	Value
PHY data rate	C_d	11 Mbps
Control rate	C_c	2 Mbps
PLCP preamble time	T_P	144 μs
PHY Header time	T_{PHY}	48 μs
MAC Header size	L_{MAC}	34 bytes
RTS Frame size	L_{RTS}	20 bytes
CTS Frame size	L_{CTS}	14 bytes
MAC ACK Header size	L_{ACK}	14 bytes
IP Header	L_{IPH}	20 bytes
UDP Header	L_{UDPH}	8 bytes
TCP Header	L_{TCPH}	20 bytes
TCP ACK Packet size	$L_{TCP-ACK}$	40 bytes
Payload Data size	L	1460 bytes
Slot time	δ	20 μs
DIFS time	$DIFS$	50 μs
SIFS time	$SIFS$	10 μs
EIFS time	$EIFS$	308 μs
Min. Contention Window	CW_{min}	31
Max. Contention Window	CW_{max}	1023
Short retry limit	K_s	7
Long retry limit	K_l	4

Table 1: Various parameters used in simulation and analysis

2.1 System Model

Let M be the total number of STAs indexed by $1 \le j \le M$. Each STA can be associated with the AP at a different packet error rate. For simplicity, we assume that the error rates belong to a finite set (this is not restrictive as this set can have as many error rates as the number of STAs). Let n_i be the number of STAs that experience packet error rate, ε_i, $1 \le i \le C$, where C is the number of error rates. We say that the set of these n_i STAs belong to "Class" i, denoted by $\mathscr{C}_i$. Since we are concerned with TCP controlled downloads, the AP transmits TCP data packets. A TCP data packet transmitted by the AP to an STA of Class i fails due to channel errors (given that the packet did not suffer a collision) with probability ε_i independent of anything else.

2.2 Approximations

We make the following approximations in our model:

A1 Since, for TCP file downloads, the STAs have to send only TCP ACK packets, which are small (40 bytes), the probability that they are received in error at the AP is also small. Therefore, we assume that *errors when the STAs transmit can be ignored*.

A2 Since TCP (for the undelayed ACK case) requires equal number of packets to be transmitted by the AP (data packets) and the STAs (ACK packets), and DCF is packet fair, it should be true that a very small number of STAs must be contending for the channel most of the time [8, 16]. Therefore, the probability that the HOL packet at the AP belongs to one of these contending STAs is also small and, we assume that every successful transmission from the AP generates a packet at a previously empty STA. Furthermore, when an STA receives data from the AP, it is assumed that it immediately generates an ACK to be queued for transmission. From the above arguments we make the approximation that *an STA can have at most one TCP ACK in its buffer* [8, 16, 4].

A3 Since the server is connected to the AP on a high speed Ethernet, it is assumed (as in [16, 8, 4]) that the AP is the bottleneck for the TCP transfers and it always has packets in its buffer to contend for the channel. Moreover, we can assume that *all the packets in the TCP window of any connection are in the AP buffer*. This can be explained as follows: Since the server is on the same LAN as the AP, the number of packets "in flight" between the server and the AP can be ignored. Also, since the STAs can have at most one TCP ACK in their buffers (as seen in A2), the number of packets in the STA buffers is very small relative to the TCP windows.

A4 *The RTS and CTS packets that have not collided are not received in error.* This assumption is justified since the RTS and CTS packets are small and are sent at a low physical bit-rate.

2.3 Assumptions

Apart from the above approximations, we also make the following assumptions:

M1 The "delayed ACK" mechanism is disabled.

M2 When RTS-CTS is enabled, RTS/CTS mechanism is used to transmit data packets whereas ACK packets are transmitted by the Basic Access mode.

3. SATURATION ANALYSIS

In this section, we deviate from the setting described in Section 2 to characterize the channel contention process when all the nodes in the WLAN are saturated (infinitely backlogged queues). Results from this analysis are used in Section 4 to obtain the stationary distribution of the number of contending STAs and the error rate of the HOL packet in the AP buffer, when the STAs are doing TCP file downloads.

We extend the single-cell model with saturated queues ([5, 15]) to include channel errors. In the single-cell model, all nodes are in the carrier-sensing range of each other. We consider the scenario in which there are n STAs and an AP, all of which are infinitely backlogged and use the same backoff parameters. Due to Approximation A1, we assume that only one node (the AP) experiences poor channel condition when it transmits, while the STAs all transmit in perfect channel conditions. Each packet sent by the AP is received in error with probability ε. DCF works such that the channel is idle when all the nodes are in the backoff state and such intervals alternate with activity on the channel (a transmission or a collision). During backoff periods, time is slotted; countdown of the backoff timers and attempts by the nodes are made at the slot boundaries. Let the backoff slots be indexed by t ($t \geq 1$). We denote the long run average transmit attempt rate of a node by β, i.e., if $A(t)$ is the number of attempts by a node until slot t, then β for that node is defined as.

$$\beta := \lim_{t\to\infty} \frac{A(t)}{t} \tag{1}$$

As in [5, 15], we assume that the node attempts at every slot with probability β independent of anything else. Let β_a and β_s be the average attempt rate of the AP and the STAs respectively. We note that, in this saturated setting, the attempt rates of all the STAs are equal due to symmetry (all of them transmit with zero error probability). Similarly, define $D(t)$ for a node, as the number of attempts by the node until slot t that have resulted in collision, and $Q(t)$ as the number of attempts until slot t in which there was no collision but the transmitted frame was received in error. Then, we define γ as the long run fraction of attempts by the node that have resulted in collision, and α as the long run fraction of attempts that have failed either due to collision or erroneous reception, i.e., for a given node,

$$\gamma := \lim_{t\to\infty} \frac{D(t)}{A(t)} \tag{2}$$

and

$$\alpha := \lim_{t\to\infty} \frac{D(t)+Q(t)}{A(t)} \tag{3}$$

We denote the collision and failure probabilities for the AP and the STAs by γ_a, γ_s, α_a and α_s. Since an attempt by an STA fails only due to collisions, the failure probability for an STA, α_s, is the same as the collision probability, γ_s.

We extend the fixed point analysis in [15] to include channel errors. In [20], the authors calculate the saturation throughput obtained when different STAs experience different channel errors. In calculating the attempt and failure probabilities, they do not differentiate between the MAC retry limits for long and short frames. We show that, unlike the case where there are no channel errors, the analysis proceeds differently for the following two cases: (1) RTS-CTS is disabled and all the packets are transmitted by the Basic Access mode (which is the same as in [20]), (2) The AP uses RTS/CTS mechanism to transmit its packets while the STAs transmit in the Basic Access mode. It can be shown that, in both the cases the fixed point equations have a unique solution for β_a and β_s for any ε and n [24].

3.1 RTS-CTS Disabled

In this case, the data frames are retransmitted until the number of attempts reaches dot11ShortRetryLimit, K_s. With α_a and α_s as defined above, the attempt probabilities of the AP (β_a) and the STAs (β_s) can be written as in [15].

$$\beta_a = G(\alpha_a) \tag{4}$$

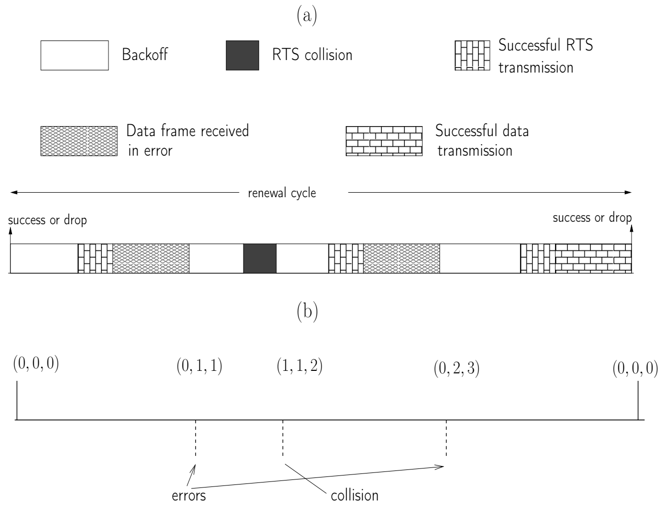

Figure 1: (a) Backoff periods and transmission attempts at the AP. Failures could be due to RTS collisions as well as channel errors. (b) Changes in the state vector at the end of activity periods. Each attempted packet starts a new backoff cycle with state $(0, 0, 0)$

$$\beta_s = G(\gamma_s) \tag{5}$$

where

$$G(\gamma) = \frac{1 + \gamma + \gamma^2 + \cdots + \gamma^{K_s-1}}{b_0 + b_1\gamma + b_2\gamma^2 + \cdots + b_{K_s-1}\gamma^{K_s-1}} \tag{6}$$

The collision probabilities for the AP and the STAs are given by

$$\gamma_a = 1 - (1 - \beta_s)^n \tag{7}$$

$$\gamma_s = 1 - (1 - \beta_a)(1 - \beta_s)^{n-1} \tag{8}$$

The failure probability for the AP can be written as

$$\alpha_a = \gamma_a + (1 - \gamma_a)\varepsilon = 1 - (1 - \beta_s)^n(1 - \varepsilon) \tag{9}$$

Equations 4, 5, 8 and 9 can be solved to obtain β_a and β_s for a given ε and n.

3.2 RTS-CTS Enabled

In this case, the data frames are retransmitted until the number of attempts reaches dot11LongRetryLimit, K_l and the RTS frames can be attempted a maximum of K_s times. Evolution of transmission attempts at the AP is shown in Fig. 1(a). As in the RTS-CTS disabled case, the instants at which the packet is transmitted successfully or discarded are taken as renewal instants (see [15]). Applying the Renewal Reward theorem to this renewal process, the attempt probability, β_a is given by

$$\beta_a = \mathbb{E}[A_a]/\mathbb{E}[B_a]$$

where $E[A_a]$ is the expected number of attempts by the AP in a renewal cycle and $E[B_a]$ is the expected total backoff duration (in slots) at the AP before the frame is either transmitted successfully or discarded. To obtain these quantities, it suffices to know the backoff periods and the outcome of the transmission (collision, erroneous transmission or successful transmission) and thus, the time periods when the channel

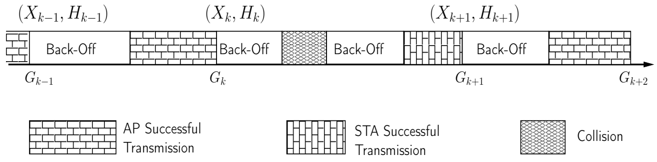

Figure 2: Evolution of channel activity: successful transmissions end at instants G_k

is active can be discarded. To calculate $\mathbb{E}[A_a]$ and $\mathbb{E}[B_a]$, we maintain in every cycle, a three-dimensional state vector (i, j, k) which changes at the end of every channel activity in the cycle (see Fig. 1(b)). Here, i represents the number of RTS collisions after the last successful RTS transmission, j is the number of data frame transmission failures since the beginning of the cycle and k is the total number of failures (collisions or errors) since the beginning of the cycle. Every cycle begins with the state $(0, 0, 0)$. Let $\mathbb{E}[A^{(i,j)}]$ be the expected number of attempts in the cycle after the current RTS frame has collided i times and the data frame transmission has failed j times, i.e., $\mathbb{E}[A^{(i,j)}]$ is the expected number of attempts in the cycle after the busy period that ends with the state (i, j, k) for any k. Then, $\mathbb{E}[A_a] = \mathbb{E}[A^{(0,0)}]$ and

$$\mathbb{E}[A^{(i,j)}] = 1 + \gamma_a\mathbb{E}[A^{(i+1,j)}] + \varepsilon(1 - \gamma_a)\mathbb{E}[A^{(0,j+1)}]$$
$$\text{for all } 0 \le i \le K_s - 1, \quad 0 \le j \le K_l - 1$$

$$\mathbb{E}[A^{(K_s,j)}] = \mathbb{E}[A^{(i,K_l)}] = 0 \qquad \text{for all } i, j$$

Similarly, let $\mathbb{E}[B^{(i,j,k)}]$ be the expected backoff duration in the cycle after the busy period that ends with state (i, j, k). Then, $\mathbb{E}[B_a] = \mathbb{E}[B^{(0,0,0)}]$ and

$$\mathbb{E}[B^{(i,j,k)}] = b_k + \gamma_a\mathbb{E}[B^{(i+1,j,k+1)}] + \varepsilon(1 - \gamma_a)\mathbb{E}[B^{(0,j+1,k+1)}]$$
$$\text{for all } 0 \le i \le K_s - 1, \quad 0 \le j \le K_l - 1, \quad k$$

$$\mathbb{E}[B^{(K_s,j,k)}] = \mathbb{E}[B^{(i,K_l,k)}] = 0 \qquad \text{for all } i, j, k$$

Thus, $\beta_a = \mathbb{E}[A_a]/\mathbb{E}[B_a] = \tilde{G}(\gamma_a, \varepsilon)$ is a function of γ_a and ε, and is obtained from the equations above. Since the STAs transmit the TCP ACKs in the Basic Access mode, $\beta_s = G(\gamma_s)$. The expressions for γ_a, γ_s, and α_a are the same as in the RTS/CTS disabled case. Equations 7, 8, $\beta_a = \tilde{G}(\gamma_a, \varepsilon)$ and $\beta_s = G(\gamma_s)$ can be solved to obtain β_a and β_s.

Comparisons between the estimates from the fixed point analysis and simulation in QualNet validate the model accuracy [24] for both the cases, i.e., when RTS/CTS is disabled and enabled.

4. TCP CONTROLLED FILE DOWNLOADS

4.1 Model for DCF Based Medium Sharing

We now consider the case when there are M STAs that are downloading large files from the local server. Taking the assumptions in Section 2 into consideration, we can develop a stochastic model akin to the ones in [16, 4]. Consider the aggregate process of channel contention and activity (see Fig. 2). Since we have long file transfers and no external arrivals, the states of the AP and the STA queues can change only at the instants when there are successful transmissions or packet discards. Due to MAC retransmissions, we note

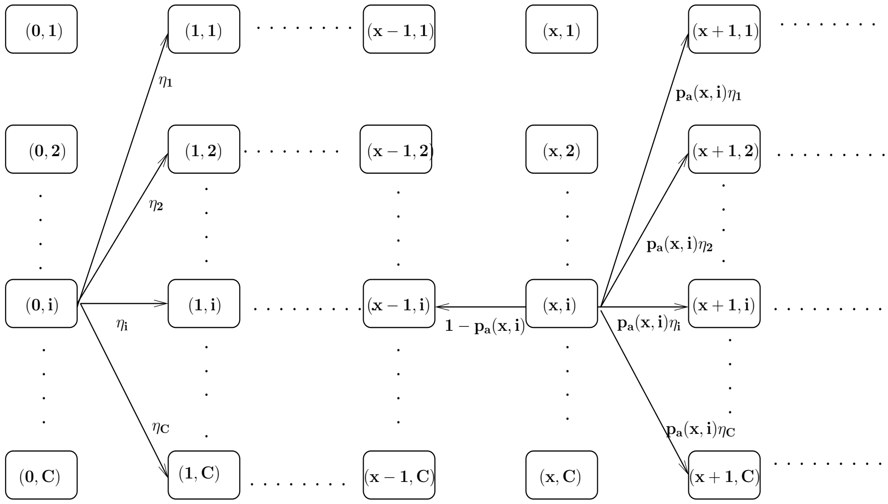

Figure 3: Transition probability diagram for the process $\{(X_k, H_k), k \geq 0\}$ embedded at epochs $G_k, k \geq 0$

that the fraction of attempts that result in a packet discard is negligibly small as compared to the fraction of attempts that result in a successful transmission. This can be verified analytically and numerically. Therefore, we make the following approximation:

A5 The states of the AP queue and the STAs' queues change only at the end of successful transmissions.

Let G_k, $k \geq 0$, denote the instants when a successful transmission ends. The successful transmission could be a data packet from the AP or a TCP ACK from one of the STAs. Let X_k denote the number of contending or *active* STAs, and H_k denote the class to which the HOL (Head Of Line) packet in the AP buffer belongs during the k^{th} *cycle* $[G_k, G_{k+1})$. By A5, these quantities change only at the end of the cycle, i.e., at G_{k+1}. We say that a packet belongs to Class i if the destination STA of the packet belongs to Class i.

A6 We assume that in a cycle if there are x active STAs and the HOL packet in the AP buffer is from Class i, then the attempt and failure probabilities are the same as those in the saturated case with $n = x$ saturated STAs and the AP frame error probability, $\varepsilon = \varepsilon_i$ ([16, 4]).

Thus, the attempt probability and failure probability for all the nodes in the cycle $[G_k, G_{k+1})$ depend only on (X_k, H_k). Therefore, the time interval between the successes, $G_{k+1} - G_k$ depends only on (X_k, H_k). At any success instant, by Approximation A2, the number of active STAs decreases by one if the successful transmission was from an STA and increases by one if the successful transmission was from the AP. Define η_i as the fraction of packets served by the AP that belong to Class i. In other words, η_i is the fraction of the aggregate TCP throughput that is obtained by STAs of Class i.

$$\eta_i := \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} I_{\{C(m)=i\}} \qquad (10)$$

where $C(m)$ is the class to which the m^{th} packet served by the AP belongs. Based on 10 (see also [4]), we assume that

A7 When a successful transmission is due to the AP, the next HOL packet is of Class i with probability η_i independent of anything else.

Therefore, it can be seen that the process, $\{(X_k, H_k), k \geq 0\}$ is a DTMC with state space $\mathcal{S} = \{(x, i) : x \geq 0, 1 \leq i \leq C\}$ where, C is the number of error classes. Note that we allow the number of active STAs to exceed M. This does not affect the accuracy of the results as, for M sufficiently large, the stationary probability of the number of active STAs exceeding M is very small for this DTMC (see Approximation A2). Also, since $G_{k+1} - G_k$ depends only on (X_k, H_k), $\{((X_k, H_k), G_k), k \geq 0\}$ forms a Markov renewal sequence [13].

The DTMC, $\{(X_k, H_k), k \geq 0\}$ has the following transition probabilities,

$$\Pr\{(X_{k+1}, H_{k+1}) = (x+1, j) | (X_k, H_k) = (x, i)\} = p_a(x, i)\eta_j \qquad (11)$$

$$\Pr\{(X_{k+1}, H_{k+1}) = (x-1, i) | (X_k, H_k) = (x, i)\} = 1 - p_a(x, i) \qquad (12)$$

where $p_a(x, i)$ is the probability that when the state is (x, i), the next successful transmission is from the AP. This is the same as the probability that the AP transmits successfully given that there is a successful transmission in a slot (see Equation 13). The transition probability diagram is shown in Fig. 3 and the matrix is of the form shown in Equation 14

$$\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{A_0} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{S_1} & \mathbf{0} & \mathbf{A_1} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{S_2} & \mathbf{0} & \mathbf{A_2} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \qquad (14)$$

where $\mathbf{A}_x$ and $\mathbf{S}_x$, $x > 0$ are non-negative matrices of order C. It can be observed that the DTMC is a level dependent Quasi-Birth-and-Death (LDQBD) process [22], where the number of active STAs represents the *level* and the class of the HOL packet represents the *phase*. Since $p_a(x, i) \in (0, 1)$, the right-hand side of 11 and 12 are positive if $\eta_j > 0$, for all j and therefore, P is irreducible if $\eta_j > 0$, for all j. It can be proved using the Foster-Lyapunov stability criterion [11] that the process is positive recurrent.

Theorem 1. *The LDQBD $\{(X_k, H_k), k \geq 0\}$ is positive recurrent.*

Proof. *Let $V(x, i) = x$ and $\mathcal{B} = \{(x, i) : 0 \leq x \leq 1, 1 \leq$*

$$p_a(x,i) = \frac{\beta_a(x,i)(1-\beta_s(x,i))^x(1-\varepsilon_i)}{\beta_a(x,i)(1-\beta_s(x,i))^x(1-\varepsilon_i) + x\beta_s(x,i)(1-\beta_s(x,i))^{x-1}(1-\beta_a(x,i))} \quad (13)$$

$i \le C\}$. Then, $\mathbb{E}[V(X_{k+1},H_{k+1}) - V(X_k,H_k)|(X_k,H_k) = (x,i)] = 2p_a(x,i) - 1 < 0$, for all $x \ge 2$ and for all i. Also, since $p_a(x,i)$ is a decreasing function of x and $2p_a(x,i)-1 < \infty$ for $x = 0,1$, we can find constants ϵ and b such that, $\mathbb{E}[V(X_{k+1},H_{k+1}) - V(X_k,H_k)|(X_k,H_k) = (x,i)] \le -\epsilon + bI_{\{i\in\mathcal{B}\}}$ for all $i \in \mathcal{S}$. Therefore, by the Foster-Lyapunov stability criterion, the process $\{(X_k,H_k), k \ge 0\}$ is positive recurrent and a stationary distribution exists. □

Thus, there exists a $\boldsymbol{\pi}$ that solves the system of equations $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ and $\boldsymbol{\pi}\mathbf{e} = 1$, where $\mathbf{e}$ is a column vector of ones. The vector $\boldsymbol{\pi}$ is partitioned by the *levels* (number of active STAs) into sub-vectors of length C such that $\boldsymbol{\pi} = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots)$ and the elements of $\boldsymbol{\pi}_x$ are the stationary probabilities for the states in level x, i.e., $\boldsymbol{\pi}_x(i)$ which is the i^{th} element of $\boldsymbol{\pi}_x$ is the stationary probability that the process, $\{(X_k,H_k), k \ge 0\}$ takes the value (x,i). Theorem 12.1.1 in [19] proves that when the process is irreducible and positive recurrent, the stationary distribution has the matrix-product form, $\boldsymbol{\pi}_{x+1} = \boldsymbol{\pi}_x\mathbf{R}_x, x \ge 0$ where $\mathbf{R}_x, (x \ge 0)$ are square matrices of order C. $\mathbf{R}_x$ can be computed recursively using the equation

$$\mathbf{R}_x = \mathbf{A}_x + \mathbf{R}_x\mathbf{R}_{x+1}\mathbf{S}_{x+2}, \qquad x \ge 0 \quad (15)$$

The stationary probabilities can, therefore, be obtained by solving the following system of equations for $\boldsymbol{\pi}_0$: $\boldsymbol{\pi}_0 = \boldsymbol{\pi}_0\mathbf{R}_0\mathbf{S}_1$ and $\boldsymbol{\pi}_0(\sum_{x=0}^{\infty}\prod_{n=0}^{x-1}\mathbf{R}_i)\mathbf{e} = 1$. To compute the matrices, $\mathbf{R}_x$, we use the algorithm by Bright and Taylor [7] in which the transition probability matrix is truncated at some large level, N. Having obtained the stationary distribution, we can find, 1) the TCP throughput, $\Theta_i^{(TCP)}$ obtained by a STA belonging to Class i (Section 4.2), and 2) the probability, p_i that a packet belonging to Class i is discarded/dropped by the AP after its Maximum Retry Limit is reached (Section 4.3), as functions of the packet error rates, $\boldsymbol{\varepsilon} := (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_C)$ and $\boldsymbol{\eta} := (\eta_1, \eta_2, \dots \eta_C)$.

4.2 TCP Throughput

Every successfully transmitted packet from the AP contributes to the aggregate throughput from the server to the STAs. Let $Z_k \in \{0,1\}$ denote the number of AP successes in the k^{th} cycle and $U_k = G_{k+1} - G_k$ denote the length of the k^{th} cycle. Let $Z(t)$ denote the number of packets served by the AP in the interval $(0,t)$. Since $\{((X_k,H_k),G_k), k \ge 0\}$ forms a Markov renewal sequence, by Markov regenerative analysis [13], the aggregate network TCP throughput, $\Theta^{(TCP)}$ is given by,

$$\Theta^{(TCP)} := L\lim_{t\to\infty}\frac{Z(t)}{t} \overset{a.s.}{=} L\frac{\sum_{x,i}\boldsymbol{\pi}_x(i)p_a(x,i)}{\sum_x\boldsymbol{\pi}_x(i)\mathbb{E}_{(x,i)}U} \quad (16)$$

where $\mathbb{E}_{(x,i)}U$ is the expected cycle time when the state is (x,i) (see [24]) and L is the payload size (assumed to be the same for all the TCP connections). Let the TCP throughput of each STA belonging to Class i be Θ_i^{TCP}. Since η_i is the ratio of the total throughput obtained by STAs of Class i to the total throughput obtained by all STAs,

$$\Theta_i^{(TCP)} = \frac{\eta_i\Theta^{(TCP)}}{n_i} \quad (17)$$

Thus, given $\boldsymbol{\eta}$ and the error rates, $\boldsymbol{\varepsilon}$, the stationary distribution, $\boldsymbol{\pi}$, can be obtained as described in Section 4.1, and the throughput of any STA can be obtained from Equation 17. The method to obtain $\boldsymbol{\eta}$ by fixed point iteration is shown in Section 4.5.

4.3 Obtaining the Drop Probability

4.3.1 RTS-CTS Disabled

In the Basic Access mode, a packet is dropped if it could not be transmitted successfully even after attempting to transmit K_s times. Define α_i as the fraction of AP attempts that result in failure when the HOL packet is of Class i. Let $\Phi_i(t)$ and $A_i(t)$ denote the number of failures and the number of attempts by the AP with HOL packet of Class i until slot t. Again, using Markov regenerative analysis, we can write

$$\alpha_i := \lim_{t\to\infty}\frac{\Phi_i(t)}{A_i(t)} \overset{a.s.}{=} \frac{\sum_x\boldsymbol{\pi}_x(i)\mathbb{E}_{(x,i)}\Phi_i}{\sum_x\boldsymbol{\pi}_x(i)\mathbb{E}_{(x,i)}A_i} \quad (18)$$

where $\mathbb{E}_{(x,i)}\Phi_i$ denotes the expected number of failed attempts (due to collisions and errors) by the AP in a cycle given the HOL packet at AP is of Class i and the number of active STAs is x, and $\mathbb{E}_{(x,i)}A_i$ is the expected number of attempts by the AP in a cycle given the HOL packet at AP is of Class i and the number of active STAs is x. Expressions for these can be obtained from renewal arguments [24]. With the following approximation, it can be seen that the drop probability, p_i, is given by $\alpha_i^{K_s}$.

A8 Every attempt to transmit a packet of Class i fails with probability α_i independent of anything else.

4.3.2 RTS-CTS Enabled

When RTS-CTS is enabled, a packet is dropped either if the number collisions of an RTS frame reaches K_s or if the packet is received in error K_l times. Define γ_i as the fraction of AP attempts that result in collision when the HOL packet belongs to Class i. Let $\Psi_i(t)$ and $A_i(t)$ denote the number of RTS collisions and the number of attempts at the AP with HOL packet of Class i in the time interval $(0,t)$. Then,

$$\gamma_i := \lim_{t\to\infty}\frac{\Psi_i(t)}{A_i(t)} \overset{a.s.}{=} \frac{\sum_x\boldsymbol{\pi}_x(i)\mathbb{E}_{(x,i)}\Psi_i}{\sum_x\boldsymbol{\pi}_x(i)\mathbb{E}_{(x,i)}A_i} \quad (19)$$

where $\mathbb{E}_{(x,i)}\Psi_i$ denote the expected number of attempts by the AP that result in RTS collisions in a cycle given the HOL packet at AP is of Class i and the number of active STAs is x. We assume that,

A9 Every attempt to transmit an RTS frame corresponding to a packet of Class i results in collision with probability γ_i independent of anything else.

As described, a packet can be dropped if is received in error K_l times, or if the RTS transmitted each time before the data frame is sent collides K_s times. The probability that the packet is dropped due to collisions in the k^{th} transmission attempt of the data frame is $\left((1-\gamma_i^{K_s})\varepsilon_i\right)^{k-1}\gamma_i^{K_s}$ $(1 \le k \le K_l)$, while the probability that the packet is dropped due to the packet being received in error K_l times is equal to $\left((1-\gamma_i^{K_s})\varepsilon_i\right)^{K_l}$. Thus, the total probability that a packet of Class i is dropped is given by,

$$p_i = \gamma_i^{K_s} \sum_{k=1}^{K_l} \left((1-\gamma_i^{K_s})\varepsilon_i\right)^{k-1} + \left((1-\gamma_i^{K_s})\varepsilon_i\right)^{K_l} \quad (20)$$

4.4 Determining η by TCP Window Analysis

We have seen in Section 4.3 that, given $\boldsymbol{\varepsilon}$, the drop probabilities, $\mathbf{p} = (p_1, p_2, \dots, p_C)$ is a function, $\mathbf{F}(.)$ of $\boldsymbol{\eta}$, both when RTS/CTS is disabled and when it is enabled. In this section, we show how to obtain the AP HOL probabilities, $\boldsymbol{\eta}$, given $\mathbf{p}$, thus providing the ingredients of fixed point iteration which will be described in Section 4.5.

Recall that η_i (as defined in Equation 10) is the fraction of packets served by the AP that belong to Class i. To calculate $\boldsymbol{\eta} = (\eta_1, \eta_2, \dots \eta_C)$, we consider the congestion window evolution of the TCP connections. As mentioned in Approximation A3 in Section 2, we assume that all the packets of the window are in the AP buffer for every TCP connection. We also initially assume that the AP buffer size is large enough to avoid buffer overflows. Later, in Section 5, we extend the analysis to finite buffer size. To analyze the window evolution, the concept of *rounds* similar to that in [23] is used. A round ends when all the packets present in the AP buffer at the beginning of the round have been either transmitted successfully or discarded. The end of a round is the beginning of the next round. Consider the process $\{\boldsymbol{W}_k, k \geq 0\} := \{(W_k^{(1)}, W_k^{(2)}, \dots, W_k^{(M)}), k \geq 0\}$, where $W_k^{(l)}$ denotes the number of packets of connection l in the AP buffer at the beginning of round k.

A10 It is assumed that an HOL packet of Class i is dropped by the AP with probability p_i independent of all other events.

A11 The loss recovery due to packet loss in any round is completed by the end of that round and, therefore, at the beginning of a round, the number of packets of any connection in the AP buffer is equal to the congestion window of that connection in the server.

Since the TCP window evolves according to packet losses, by A10 and A11, the process $\{\boldsymbol{W}_k, k \geq 0\}$ is a DTMC with a finite state space, $\mathcal{W} = \{0, 1, 2, \dots, W_{max}\}^M$ (W_{max} is the maximum receive window). If the DTMC is irreducible, let $\boldsymbol{\nu}$ be its stationary distribution (for the window evolution models that we consider in this paper, the DTMC turns out to be irreducible).

To calculate η_i, note that $\sum_{j \in \mathscr{C}_i} W_k^{(j)}$ packets of Class i and a total of $\sum_{l=1}^{M} W_k^{(l)}$ packets are served in the round k. Define Y_k as the total number of packets served in all the rounds before the beginning of round k. Then, $Y_{k+1} - Y_k = \sum_{l=1}^{M} W_k^{(l)}$ and, $\{(\boldsymbol{W}_k, Y_k), k \geq 0\}$ forms a Markov renewal sequence. Therefore, by Markov regenerative analysis [13],

$$\eta_i = \lim_{n\to\infty} \frac{1}{n} \sum_{m=1}^{n} I_{\{C(m)=i\}} \quad (21)$$

$$\overset{a.s.}{=} \frac{\sum_{j \in \mathscr{C}_i} \sum_{\boldsymbol{w} \in \mathcal{W}} \boldsymbol{\nu}(\boldsymbol{w}) w_j}{\sum_{j=1}^{M} \sum_{\boldsymbol{w} \in \mathcal{W}} \boldsymbol{\nu}(\boldsymbol{w}) w_j} \quad (22)$$

$$= \frac{n_i \mathbb{E}_{\boldsymbol{\nu}} W_i}{\sum_{l=1}^{C} n_l \mathbb{E}_{\boldsymbol{\nu}} W_l} \quad (23)$$

where $\boldsymbol{w} = (w_1, w_2, \dots, w_M)$ in Equation 22, and $\mathbb{E}_{\boldsymbol{\nu}} W_i$ is the stationary expected value of the window of a STA of Class i. Since $\mathbb{E}_{\boldsymbol{\nu}} W_i$ is a function of p_i, $\boldsymbol{\eta}$ can be calculated as a function, $\mathbf{T}(.)$ of $\mathbf{p}$.

When the AP buffer is infinite, losses are only due to channel errors and thus, by A10, the window evolution of any connection is independent of that of other connections, i.e., $\{W_k^{(j)}, k \geq 0\}$ is independent of $\{W_k^{(l)}, k \geq 0\}$ for all $j, l \in \{1, 2, \dots, M\}$. Denote the stationary distribution of the DTMC, $\{W_k^{(j)}, k \geq 0\}$, $j \in \mathscr{C}_i$ by $\boldsymbol{\nu}_i$. We calculate $\mathbb{E}_{\boldsymbol{\nu}} W_i$ by considering a stochastic model for the evolution of the window of a TCP Reno connection when the probability that a packet is dropped is p_i. We make the following assumptions:

A12 The probability of more than one loss in a round is negligible.

A13 It is assumed that if there is a loss, recovery always happens by fast retransmit and fast recovery so that the connection is always in congestion avoidance phase.

In the congestion avoidance phase, the congestion window increases by one if all the packets in the window are acknowledged. Therefore, at the end of a round, the window of a connection increases by one if all the packets belonging to that connection have been transmitted successfully by the AP, else the window reduces to half of its previous value. For drop probability p, the window evolution process, $\{V_k, k \geq 0\}$ has the state space $\{1, 2, \dots, W_{max}\}$ and transition probabilities given by

$$\mathsf{Pr}\{V_{k+1} = w+1 | V_k = w\} = (1-p)^w$$

$$\mathsf{Pr}\{V_{k+1} = \lceil (w/2) \rceil | V_k = w\} = 1 - (1-p)^w$$

The stationary distribution for this DTMC can be computed numerically which can be used to calculate $\boldsymbol{\nu}_i$ and $\mathbb{E}_{\boldsymbol{\nu}} W_i$ in Equation 23 for a given drop probability, p_i.

4.5 The Fixed Point Iteration

As seen in Section 4.3, $\mathbf{p} = \mathbf{F}(\boldsymbol{\eta})$. Further, in Section 4.4, we obtained $\boldsymbol{\eta} = \mathbf{T}(\mathbf{p})$. Hence, we can obtain $\boldsymbol{\eta}$ by solving the fixed point equation, $\boldsymbol{\eta} = \mathbf{T}(\mathbf{F}(\boldsymbol{\eta}))$. We obtained $\boldsymbol{\eta}$ from the fixed point iteration, $\boldsymbol{\eta}^{(k+1)} = \mathbf{T}(\mathbf{F}(\boldsymbol{\eta}^{(k)}))$. It has been observed that $\boldsymbol{\eta}^{(k)}$ converges to the same solution irrespective of the initial value but we have not been able to prove convergence to a unique solution. The value of $\boldsymbol{\eta}$ thus obtained can be used to calculate the stationary probabilities of the LDQBD and throughputs of the TCP connections using Equations 16, 17 .

5. EXTENSIONS TO FINITE AP BUFFER

Until now, we assumed that the size of the buffer at the AP is so large that for the given W_{max}, there are no packet drops due to buffer overflow (tail drop losses). But with TCP window scale option, this assumption is not valid as W_{max} is very large. We now consider the effects of finite buffer size at the AP.

Let the size of the buffer at the AP be B packets. It can be seen that the Markov chain model (Section 4) for channel contention is still valid when the AP buffer is finite. But the fraction of packets of each connection served by the AP, $\boldsymbol{\eta}$, varies with B. Thus, we need to develop a model for

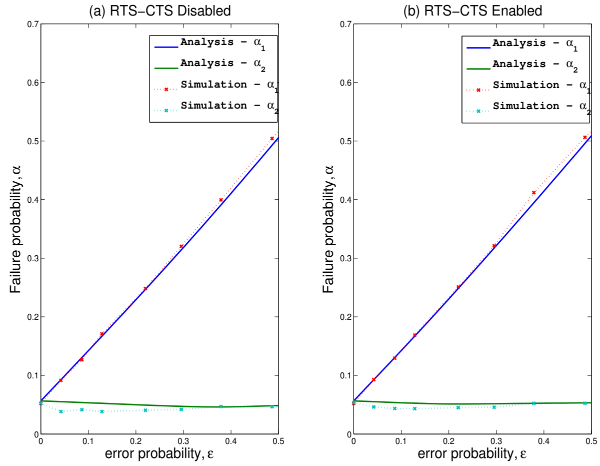

Figure 4: Probability that an attempt by the AP fails for each class when there are 2 classes and the AP buffer is very large. Class 1 has error probability ε and Class 2 has error probability 0. TCP Reno is used with $W_{max} = 45$ packets (a) RTS-CTS Disabled (b) RTS-CTS Enabled

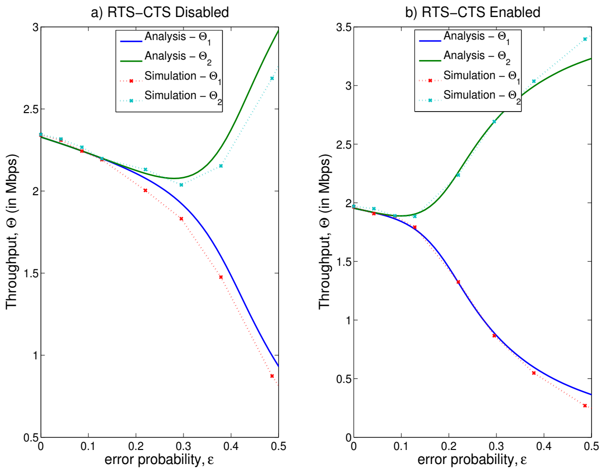

Figure 5: TCP Throughput obtained by each class when there are 2 classes and AP buffer is very large. Class 1 has error probability ε and Class 2 has error probability 0. TCP Reno is used with $W_{max} =$ 45 packets (a) RTS-CTS Disabled (b) RTS-CTS Enabled

TCP congestion window which takes into account not only the packet losses due to channel errors but also losses due to buffer overflow to determine the value of $\boldsymbol{\eta}$. The evolution of the congestion windows of different connections are not any more independent of each other. To model the dependence, we extend the TCP Reno model. In addition to A12 and A13, we make the following approximation:

A14 As in [4], we assume *immediate feedback*, i.e., the time between the data packet being served at the AP and the corresponding TCP ACK being received at the server is ignored. This is justified since the average number of active STAs is small and, thus, the STAs can transmit the ACKs within a short period of time. A consequence of this assumption is that there is a tail drop loss only when the congestion window increases at the server. When there is no window growth, every packet served by the AP is immediately replaced in the buffer by a new packet due to immediate acknowledgment. When the buffer is full and the window grows by one, two packets arrive at the tail of the buffer but the packet served by the AP leaves space only for a single packet, resulting in tail drop loss. This leads us to the approximation that *only those connections which have had no losses due to channel errors in the round can have window growth and therefore, can lose a packet due to buffer overflow.*

The process $\{\boldsymbol{W}_k, k \geq 0\}$ evolves as follows: Let $S_k \subset \{1, 2, \ldots, M\}$ be the set of connections that lose packets due to channel errors in round k.

$$\mathsf{Pr}\{S_k = S | \boldsymbol{W}_k = (w_1, w_2, \ldots, w_M)\} = \prod_{i \in S} (1 - (1 - p_i)^{w_i}) \times \prod_{i \in S^c} ((1 - p_i)^{w_i}) \quad (24)$$

The congestion windows of the connections in S_k are halved and thus, these connections do not have tail drop losses in round k. If $l \in S_k$, then $W_{k+1}^{(l)} = W_k^{(l)}/2$. All the other connections grow their window by one and the number of packets lost due to buffer overflow is given by $\max\{\sum_{i \in S_k} W_k^{(i)}/2 + \sum_{j \in S_k^c} (W_k + 1) - B, 0\}$. Since all the connections in S_k^c need not have tail drop loss, we pick the connections that have tail drop loss randomly from S_k^c. It can be observed that the process $\{W_k, k \geq 0\}$ is a DTMC with finite state space, $\{1, 2, \ldots, \min\{B, W_{max}\}\}^M$. Thus, we can obtain the stationary expected value of $\{\boldsymbol{W}_k, k \geq 0\}$ to determine $\boldsymbol{\eta}$ from Equation 23.

6. MODEL VALIDATION AND DISCUSSION

All the simulation results are obtained using QualNet 4.5.1 with 802.11b. The parameters used in simulation and analysis are summarized in Table 1. We set the TCP Maximum Segment Size (MSS) to 1460 bytes and Maximum Receive Window, W_{max} to 65,535 bytes (45 packets) when there is no window scaling. Auto-rate fallback is disabled. Simulation results are obtained for two values of RTS Threshold: (a) When the RTS threshold is set to 1600 bytes, RTS-CTS mechanism is disabled as all the frames have sizes less than 1600 bytes (see Table 1), and (b) When the RTS threshold is set to 600 bytes, the RTS-CTS mechanism is enabled as the size of the data frames exceeds this threshold.

Consider the case where there are two STAs ($C = 2$); STA in Class 1 receives frames at an error rate of ε and STA in

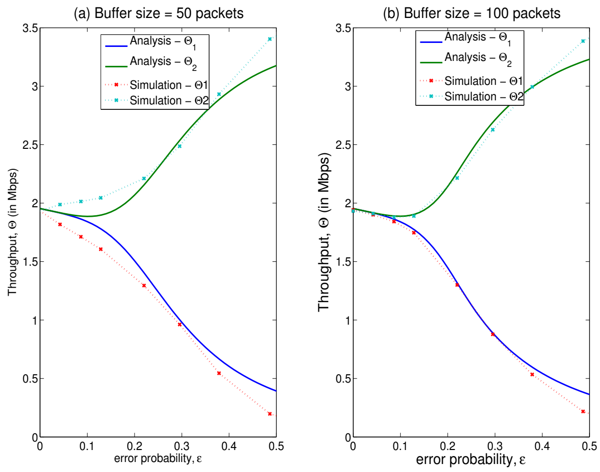

Figure 6: TCP Throughput obtained by each class when there are 2 classes. Class 1 has error probability ε and Class 2 has error probability 0. TCP Reno is used with W_{max} = 45 packets and RTS-CTS is enabled (a)B = 50 packets (b) B = 100 packets

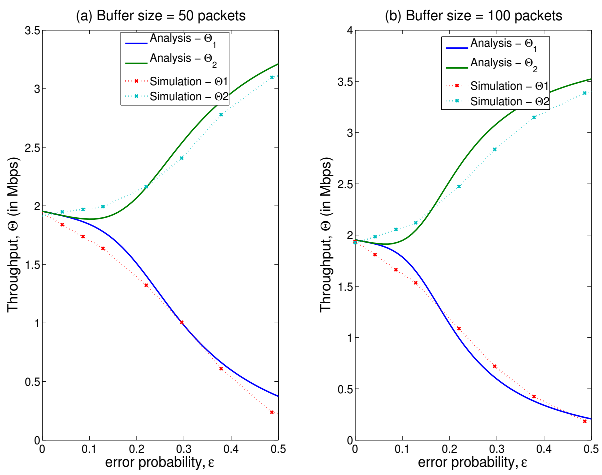

Figure 7: TCP Throughput obtained by each class when there are 2 classes. Class 1 has error probability ε and Class 2 has error probability 0. TCP Reno is used. RTS-CTS and window scaling are enabled (a)B = 50 packets (b) B = 100 packets

Class 2 correctly receives, all frames that do not collide, i.e., $\varepsilon_1 = \varepsilon$ and $\varepsilon_2 = 0$. We first consider the case when there is no window scaling and the AP buffer is large enough to avoid buffer overflows.

Fig. 4 shows plots of the probability that an attempt by the AP to transmit packets of Class 1 (α_1) and Class 2 (α_2) fails. It is seen that α_1 increases linearly with ε while α_2 remains almost constant as failure for packets of Class 2 is only due to collisions. The TCP throughput obtained by the STAs in each class is shown in Fig. 5 for the two cases.

1. It is seen that, while the aggregate network throughput decreases due to wasted time in transmitting erroneous frames, the throughputs of the two classes is equal at low error probabilities. This is because at low values of ε, the drop probability is low and both the classes will have their TCP window very close to W_{max}. Thus, the MAC level retransmission mechanism protects the TCP connections from channel losses to a large extent, and connections with sufficiently low frame error rate get the same throughput.

2. As ε increases, the STA that has a poor channel obtains a much smaller throughput as compared to the STA with a good channel. Unlike in the saturated case where unfairness occurs due to the backoff mechanism in IEEE 802.11 DCF which fails to distinguish between a collision and transmission failure due to link errors [20], unfairness in the case of TCP file downloads is due to: a) Inability of TCP to distinguish between packet losses due to congestion and packet losses due to channel errors. This leads to STAs with worse channels having a smaller TCP window and thus, fewer packets in the AP buffer. b) FIFO queuing at the AP buffer, which results in each STA getting a throughput proportional to the fraction of its packets in the AP buffer.

3. We can observe that the divergence in the throughput curves occurs at a lower value of ε when RTS-CTS is enabled as compared to the RTS-CTS disabled case. This can be justified by noting that the packet transmission attempt limit, K_l when RTS-CTS is enabled is less than the attempt limit, K_s when RTS-CTS is disabled. Thus, for the same value of ε, more number of packets of Class 1 are dropped when RTS-CTS is enabled, leading to lower throughput for Class 1.

We now consider the effect of finite buffer size at the AP for two cases: (i) window scaling option disabled, (ii) window scaling option enabled. When there is no window scaling, we set W_{max} to 45 packets and when window scaling is enabled, the scale factor is 14, which means that the window can grow up to 45×2^{14}. Fig. 6 shows the throughput plots when there is no window scaling (W_{max} = 45 packets) for buffer sizes of 50 and 100 packets. Note that, in this case, having a buffer size of 100 packets is same as having an infinite buffer. Fig. 7 shows the throughput plots when window scaling is enabled and B = 50, 100 packets. The difference in the values obtained from analysis and simulations is due to Approximation A14, where we neglect the tail drop losses of STAs that have lost packets due to channel errors.

7. CONCLUDING REMARKS

In this paper, we have addressed the problem of modeling TCP controlled file downloads by STAs with different channel conditions in an IEEE 802.11 WLAN. By combining a model for contention by DCF with a model for the TCP congestion window, we could calculate the packet failure probabilities and the throughputs obtained by the STAs. The simple TCP model that was valid for an infinite AP buffer was then extended to the case where the buffer size is finite. It was shown that STAs with poorer channel conditions get lower throughput due to two reasons: a) Inability of TCP to recognize losses due to channel errors b) FIFO queuing at the AP buffer. The latter problem can be solved by having a separate queue for each STA at the AP and serving each queue in a fair manner. Such an approach would be possible with IEEE 802.11n APs and our analysis has been found to be useful in this situation as well [24]. The results in this paper can also provide insights to the trade offs involved in designing rate adaptation algorithms. For eg., if two STAs are associated at the same PHY rate, and one of them experiences channel losses at that rate, the rate adaptation algorithm must compare the reduction in the throughput due to channel losses at that rate (small for packet error rates up to 10-20%) with the reduction in the throughput if data is transmitted to that STA at a lower rate.

8. REFERENCES

[1] Minstrel rate control algorithm. `http://madwifi-project.org/changeset/4030/madwifi/trunk/ath_rate/minstrel`.

[2] Onoe rate adaptation algorithm. `http://madwifi-project.org/svn/madwifi/branches/madwifi-0.9.4/ath_rate/onoe/`.

[3] Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specification, 2007.

[4] O. Bhardwaj, G. V. V. Sharma, M. Panda, and A. Kumar. Modeling finite buffer effects on tcp traffic over an ieee 802.11 infrastructure wlan. *Comput. Netw.*, 53:2855–2869, November 2009.

[5] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *IEEE Journal on Selected Area in Comm.*, 15(3):535–547, March 2000.

[6] J. C. Bicket. Bit-rate selection in wireless networks. Technical report, Master's thesis, MIT, 2005.

[7] L. Bright and P. G. Taylor. Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Stochastic Models*, 11(3):497–525, 1995.

[8] R. Bruno, M. Conti, and E. Gregori. Performance modelling and measurements of tcp transfer throughput in 802.11-based wlan. In *MSWiM*, pages 4–11, 2006.

[9] P. Chatzimisios, A. Boucouvalas, and V. Vitsas. Influence of channel BER on IEEE 802.11 DCF. *IEE Electronic Letters*, 39(23), Nov 2003.

[10] S. Ha, I. Rhee, and L. Xu. Cubic: a new tcp-friendly high-speed tcp variant. *SIGOPS Oper. Syst. Rev.*, 42:64–74, July 2008.

[11] B. Hajek. An exploration of random processes for engineers, January 2009.

[12] B. H. Jung, S. J. Kim, H. Jin, H. Y. Hwang, J. W. Chong, and D. K. Sung. Performance improvement of error-prone multi-rate wlans through adjustment of access/frame parameters. In *ICC*, pages 1–6, 2009.

[13] V. G. Kulkarni. *Modeling and analysis of stochastic systems*. Chapman & Hall, Ltd., London, UK, UK, 1995.

[14] A. Kumar. Comparative performance analysis of versions of tcp in a local network with a lossy link. *IEEE/ACM Trans. Netw.*, 6(4):485–498, 1998.

[15] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed-point analysis of single cell ieee 802.11 wlans. *IEEE/ACM Trans. Netw.*, 15(3):588–601, 2007.

[16] G. Kuriakose, S. Harsha, A. Kumar, and V. Sharma. Analytical models for capacity estimation of ieee 802.11 wlans using dcf for internet applications. *Wireless Networks*, 15(2):259–277, 2009.

[17] M. Lacage, M. H. Manshaei, and T. Turletti. Ieee 802.11 rate adaptation: a practical approach. In *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 126–134, 2004.

[18] T. V. Lakshman and U. Madhow. The performance of tcp/ip for networks with high bandwidth-delay products and random loss. *IEEE/ACM Trans. Netw.*, 5:336–350, June 1997.

[19] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling.* ASA-SIAM, Philadelphia, 1999.

[20] E. Lopez-Aguilera, J. Casademont, and E. G. Villegas. A study on the influence of transmission errors on wlan ieee 802.11 mac performance. *Wireless Communications and Mobile Computing*, March 2010.

[21] S. Mare, D. Kotz, and A. Kumar. Experimental validation of analytical performance models for IEEE 802.11 networks. In *Proceedings of the Workshop on WIreless Systems: Advanced Research and Development (WISARD 2010)*, pages 1–8. IEEE Computer Society Press, January 2010.

[22] M. Neuts. *Matrix Geometric Solutions in Stochastic Models. An Algorithmic Approach.* John Hopkins University Press, Baltimore, 1981.

[23] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose. Modeling tcp reno performance: a simple model and its empirical validation. *IEEE/ACM Trans. Netw.*, 8(2):133–145, 2000.

[24] K. Subhashini. Towards centralized control of channel allocation and scheduling in IEEE 802.11 WLANs. Master's thesis, Indian Institute of Science, 2011.

[25] Y. C. Tay and K. C. Chua. A capacity analysis for the ieee 802.11 mac protocol. *Wireless Networks*, 7:159–171, 2001.