Analysis of a fountain codes based transport in an 802.11 WLAN cell

Dinesh Kumar, Tijani Chahed, Eitan Altman

Analysis of a Fountain Codes Based Transport in an 802.11 WLAN Cell

Dinesh Kumar
INRIA, Sophia Antipolis, France
Email: dkumar@sophia.inria.fr

Tijani Chahed
TELECOM & Management SudParis, Evry, France
Email: Tijani.Chahed@it-sudparis.eu

Eitan Altman
INRIA, Sophia Antipolis, France
Email: altman@sophia.inria.fr

***Abstract*—A Fountain Codes based Transport (FCT) protocol relies on an alternate paradigm to that of the ubiquitous TCP. It abolishes the need for a reverse feedback mechanism usually essential to provide reliability in packet data transmission. Absence of a reverse feedback mechanism can substantially improve the performance of networks with half-duplex wireless channels (such as 802.11 WLANs), where collisions between forward and reverse MAC frame transmissions contribute significantly towards performance degradation. We propose a Markovian stochastic framework to model the performance of a simple FCT protocol in a single cell IEEE 802.11 WLAN. Our model allows the WLAN Access Point to employ a generic rate control algorithm for MAC frame transmissions on the downlink. Using renewal theory we provide an explicit expression for the average downlink throughput. *ns2* simulations are used to validate our model and the analytically obtained throughput metric. A detailed performance analysis study is then carried out to provide insights into the choice of various system parameters that can lead to optimal throughput performance. Finally we present a brief comparison between the performance of FCT and TCP through simulations.**

I. Introduction

Fountain Codes [1], [2] are rateless erasure codes and offer a very promising future for ameliorating existing data packet transmission techniques. Specific type of fountain codes such as Luby Transform (LT) Codes, Raptor codes, etc. [3], [4], can be used by a sender to generate encoded packets from source data packets *on the fly* and the number of such encoded packets can be potentially limitless. Unlike traditional end-to-end transmission mechanisms like the popular TCP protocol, Fountain Codes based Transport (FCT) protocols do not require re-transmission of lost packets. If the original file at sender side comprises N_p packets, then by decoding *any* set of $N_p(1+\epsilon)$ (slightly more than N_p) fountain coded packets received, the receiver can recover the whole file with probability $1-\delta$. The probability of failure to decode the file, δ, is bounded above by $\delta \leq 2^{-\epsilon N_p}$ and depends on the *degree distribution* used to code the packets at sender side [1]. The quantity $1+\epsilon$ denotes the *decoding inefficiency* and its value may vary from 1.05 to 1.4 depending on the coding scheme used [1], [5]. End-to-end transport protocols based on Fountain Coded Packets ($FCPs$) can thus offer reliable transmission without the need for re-transmitting lost information.

It is well known that the standard TCP for wired networks performs very poorly for asymmetric wireless channels [6]–[8]. Asymmetry in wireless channels arises mainly due to difference in bandwidth and latency on the forward and reverse paths. Asymmetric bandwidth can easily break the TCP ack clocking mechanism, i.e., acks get spaced farther apart due to queuing at the bottleneck link with lower bandwidth (see [8] and references there in). On the other hand, asymmetric latency may cause high transmission delays due to half-duplex radios, overhead per packet due to MAC protocol, etc. [8]. Moreover, networks with half-duplex channels such as 802.11 WLANs may suffer from severe performance degradation due to large number of MAC frame collisions when TCP ack traffic is present on the reverse link [9]. One solution to all these problems could be to adopt an FCT protocol instead of TCP. A few FCT protocols have been proposed by some authors in [5], [10]. These protocols aim to replace TCP and hence increase a sender's transmission efficiency by abandoning altogether the need for TCP ack packets. In this paper, we propose a simple stochastic model for the performance analysis of an FCT protocol in a single cell 802.11 WLAN. Our model allows for the Access Point (AP) to employ a *generic* adaptive rate control algorithm such as Adaptive Auto Rate Fallback (AARF) or Adaptive Multi Rate Retry Algorithm (AMRR) [11]. Such algorithms are used to adaptively select PHY (physical layer) rates for packet transmission, depending on the varying wireless medium characteristics.

Motivation and Related Work: FCT protocols are expected to exhibit improved efficiency for various different kinds of networks and applications (see [12] and references there in). For instance in multicast networks they can assist in avoiding feedback implosion and provide simple means of handling heterogeneous users and disparate start times [12]. Other than that they can also be effective in overlay networks [12]. However, network architectures in the future are more likely to be *heterogeneous* in nature with the communication path from one end to another consisting of both (multi-hop) wired and (last-hop) wireless links, such as in the case of 802.11 WLANs [13]. As compared to wired links, wireless links in a heterogeneous network possess certain undesired characteristics such as high link error rates, limited capacity, etc. Thus, performance of an FCT protocol is more critical over the last-hop wireless link. Moreover, since 802.11 WLANs are becoming ubiquitous, we are motivated in this paper to carry out a performance analysis study of an FCT protocol for the specific case of last-hop wireless link in a single cell WLAN.

Due to patent protection of fountain codes (LT codes to be specific) [14], research efforts in areas related to fountain codes have been limited. Consequently, very little research literature on FCT protocols is available. Nonetheless, some authors in [5], [10], [15], [16] have proposed and studied a few FCT protocols. Lopez et al. in [10] present a game theoretic analysis of protocols based on fountain codes and their results suggest that in a generic setting, hosts using TCP have an incentive to switch to an FCT protocol. However, as per our knowledge none of the previous works have mathematically modeled and analyzed the performance of an FCT protocol in an 802.11 WLAN setting.

Main Contributions: The contributions of this work are *threefold*. Firstly, this work is the first we know of that provides a simple stochastic model for an FCT protocol in an 802.11 WLAN. In addition, our model allows for the WLAN AP to employ a *generic* rate control algorithm. Secondly, we provide a performance analysis study of the FCT protocol in a WLAN setting in which we observe certain trade-offs and performance variations w.r.t. various system parameters. These observations lead to hints for choosing system parameter values that can achieve optimal network throughput performance. Thirdly, we provide some performance comparisons between FCT and TCP through a simulation study.

The paper is organized as follows. Section II describes a basic FCT protocol that we study in this paper. Section III presents the modeling framework which is a basis for the throughput metric derived in Section IV. We then validate the model using *ns2* simulations in Section VI and present a performance analysis study in Section VI. A brief simulation based comparison between the FCT and TCP is presented in Section VII and Section VIII concludes the paper.

II. A Fountain Codes based Transport (FCT)

For our analysis and simulation purposes, we consider a very basic FCT protocol similar to the one proposed by authors in [10]. The protocol is described as follows. Given a data file of size N_p packets, the application or transport layer at the sender side encodes these packets *on the fly* using a suitable fountain code, say e.g., the LT codes. These fountain coded packets (FCPs) are then passed on to the lower layers which in turn transmit these FCPs to the receiver. The number of FCPs that can be generated from the original N_p packets is potentially limitless due to the design of LT codes. Thus, the sender keeps on transmitting a new set of FCPs pertaining to the same original file of N_p packets, until it receives some kind of acknowledgement from the receiver indicating the receipt of a sufficient number of FCPs that may now be decoded to recover the original file. Every FCP is useful and can be exploited to decode the complete file once the receiver has received at least $N_p(1+\epsilon)$ FCPs (slightly more than N_p). After collecting at least $N_p(1+\epsilon)$ FCPs, the receiver sends an *un-coded* 'STOP' message packet to the sender indicating the receipt of sufficient number of FCPs for being able to decode the complete file of N_p packets. The sender then immediately ceases to transmit any more FCPs. Note that this FCT protocol is different from the usual UDP protocol mainly because of the 'STOP' message acknowledgement.

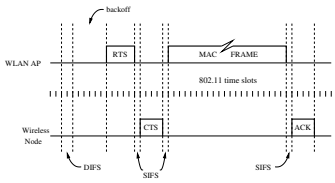

Fig. 1. MAC frame transmission *sequence*

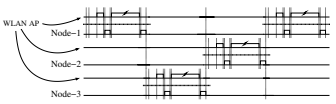

Fig. 2. Balanced resource allocation by AP

As for the 802.11 WLAN, we assume that the transport layer operates in a *split mode* similar to that in the case of Wireless TCP (see [8] and references there in). In split mode the WLAN AP acts as a terminal node for both the wired network and the last-hop wireless link. Both sender and receiver communicate independently with the WLAN AP router without any knowledge of each other. In our case, the split mode is assumed to operate such that TCP is employed for the wired network and FCT protocol is employed for the last-hop wireless link.

Remark: In our protocol description the receiver sends 'STOP' message after it has received at least $N_p(1+\epsilon)$ FCPs. However, our protocol could be modified such that the receiver sends 'STOP' message *not* after receiving $N_p(1+\epsilon)$ FCPs but only after it has successfully decoded and recovered the original file (since FCPs in excess of $N_p(1+\epsilon)$ will reduce the bound $2^{-\epsilon N_p}$ on the failure probability). It is to be noted that this modification or any such similar modification does not affect our modeling and analysis that follow because our model is valid for *any* number of FCPs received by the receiver.

III. Model

Assume there are N 802.11 nodes present in a single cell WLAN including the AP. Therefore, there are $N-1$ wireless nodes connected to an AP, engaged in either streaming (e.g., video streaming) or interactive (e.g., HTTP like traffic) data transmissions. We limit our focus to *downlink* transmissions in the WLAN cell since the bulk of data transfers are usually carried out from an AP to the wireless nodes. Assume a simple homogeneous scenario where each of the $N-1$ nodes downloads a file of size N_p un-coded packets from the AP. Each node must therefore receive at least $D \triangleq N_p(1+\epsilon)$ FCPs to decode the original file with probability $1-\delta$. Time is divided into a discrete set of 802.11 *slots*. For instance, in IEEE 802.11b standard each slot is $20\mu s$ long. The AP is assumed to implement the optional RTS/CTS mechanism in order to reserve the channel medium for transmission of each MAC frame. Figure-1 illustrates a MAC frame transmission *sequence* between the AP and a single wireless node, comprising RTS, CTS, MAC data and ACK frames. The MAC frame here consists of the FCP passed on to the MAC layer by upper layers. The commencement of every frame transmission coincides with the beginning of an 802.11 time slot. Various time intervals of Distributed Inter-Frame Spacing (DIFS), Short Inter-Frame Spacing (SIFS) and a random back-off,

separate the commencement of frame exchanges as shown in Figure-1. As per the IEEE 802.11 standard, the length of back-off interval (in terms of number of slots) is uniformly randomly chosen from a set $\{0, 1, \ldots, CW - 1\}$, where CW denotes the current size of the contention window corresponding to the CSMA/CA based Distributed Coordination Function (DCF). The mean length of back-off interval, $E[T_B]$, has been derived in previous works (see [17]–[19]).

A. Balanced Resource Allocation by AP

Considering only downlink traffic from the AP to the $N-1$ wireless nodes, it has been shown in a previous work [22] that for HTTP like traffic that employs TCP, the downlink throughput of AP to all wireless nodes is equally and fairly distributed among the $N-1$ nodes. In other words, the average number of transmission opportunities (*TxOPs*) acquired by the AP for MAC frame transmission to each wireless node are nearly equal over a long period of transmission. We can safely assume a similar balanced resource allocation in terms of *TxOPs* for our simple FCT due to the way it has been defined in Section-II.

B. Wireless Channel Model

For the purpose of modeling the wireless channel medium we consider a standard two state Markov channel error model. Such a model is depicted in Figure-3. Across *two consecutive 802.11 slots*, the channel may transit between a *good* and a *bad* state as shown in Figure-3, where p_g is the probability that the channel remains in good state, $1 - p_g$ is the probability of transiting from good to bad state, p_b is the probability of remaining in the bad state and $1 - p_b$ is the probability of transiting from bad to good state. The two probabilities p_g and p_b may or may not be related depending on the channel conditions. Note that, if the channel ever goes into bad state for *even a single slot* during the transmission of *any* of the RTS, CTS, MAC or ACK frames, the corresponding frame is received erroneously and considered lost. Such a situation will lead to an *unsuccessful* MAC frame transmission sequence. If the channel stays in good state throughout, it will lead to a *successful* MAC frame transmission sequence.

C. Generic Rate Control and Mobility

Most of the IEEE 802.11 standards (e.g., 802.11 a/b/g) support multiple PHY data transmission rates. In our framework, we allow the AP to dynamically select a PHY rate adapted to the continuously varying wireless medium characteristics such as fading, attenuation, etc. We further assume that the corresponding rate control algorithm ensures a balanced resource allocation (in terms of *TxOPs*) among the $N - 1$ wireless nodes, as discussed in Section-III-A. Many rate control algorithms have been proposed in the literature. Some of these include the Auto Rate Fallback (ARF) algorithm, Receiver Based Auto Rate (RBAR) algorithm and the Adaptive Multi Rate Retry (AMRR) algorithm (see [11] and references there in). Whichever algorithm is employed, we assume that the AP may dynamically select up to K distinct transmission

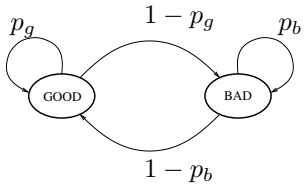

Fig. 3. Two state Markov channel error model

rates for the downlink *as a function of the channel conditions*. Let $\{r_1, \ldots, r_K\}$ denote these K different rates in units of $bits/slot$ and let $\mathcal{K} = \{1, \ldots, K\}$ denote their index set.

In brief, the aforementioned rate control algorithms function in the following manner. Either the receiver's measure of the Signal to Noise Ratio (SNR) of received signal or the sender's estimation of the channel loss probability are exploited to adaptively select a higher or lower PHY rate among the K available rates. The PHY rate is increased or decreased either on a MAC frame by frame basis (see [11] and references there in), or after a suitably calculated number of MAC frame transmission attempts elapse or at large ($\gg$ one 802.11 slot) periodic intervals with varying periodicity (see [11] and references there in). In order to *generalize* these different set of rate selection criteria, for our model we introduce a generic PHY rate transition matrix which is a function of the channel conditions represented by the pair $p = (p_g, p_b)$. This matrix is denoted as $H(p) = [h_{k,j}(p)]_{K \times K}$, $k, j \in \mathcal{K}$, which is common to all $N - 1$ AP to wireless node connections. The elements of this matrix denote the *mean* probability, $h_{k,j}(p)$, with which the chosen rate control algorithm will alter the PHY rate to r_j for the next MAC frame transmission attempt, given that the PHY rate for the ongoing frame transmission is r_k. Such a matrix can be appropriately defined *as a function of the channel conditions*, $p = (p_g, p_b)$, for any given rate control algorithm. For instance, for good channel conditions, e.g., $p = (0.99, 0.05)$, a rate control algorithm could define the values $h_{k,j}(p)$ such that a high PHY rate is selected in steady state and no rate transitions occur unless channel conditions deteriorate. As a crucial basis for the discussion in Section-III-D, we assume that the PHY rate transition process is stationary for each node and given channel conditions. Equivalently, each $H(p)$ is assumed to be time homogeneous.

Wireless nodes in the AP cell are allowed to be mobile in our framework. However, we do not explicitly model the mobility of such nodes. Instead, we assume that the PHY rate transition matrix H can be tuned in real time in order to incorporate an additional rate control strategy that may be required to take care of the effects on channel conditions due to mobility of nodes.

D. Formulation as Markov Renewal Reward Process

Due to its design, the DCF protocol defined in 802.11 standard allows only one successful MAC frame exchange to be ongoing at any time. Then if we focus only on downlink, the AP will be engaged in a MAC frame transmission sequence with only one of the $N - 1$ nodes at any instant. Moreover, by the balanced resource allocation assumption, such frame transmissions will be nearly equally distributed over time for

all $N-1$ wireless nodes. Figure-2 shows a snapshot of such a scenario for $N=4$. Further, collision between RTS frames from AP and the $N-1$ nodes can be ignored since 'STOP' messages on the reverse link (uplink) can be negligible for large file sizes. Indeed, fountain codes are most effective when N_p is large [1], [2], [5]. Hence, modeling of 'STOP' messages can be ignored for relatively large values of N_p. In fact, our simulation results in Section V-B confirm this assumption. Therefore, from now onwards we will consider FCP transmissions from AP to only one of the $N-1$ nodes and we argue that the same analysis holds for any of the other nodes in the AP cell.

Remark: By ignoring the 'STOP' messages for modeling purpose, the proposed FCT essentially reduces to UDP. However, our main goal here is to study performance variation of FCT in terms of the choice of *fountain code parameter* (ϵ) and *FCP size* as a function of the wireless channel conditions and number of nodes N.

Now, for given channel conditions note that the PHY rate transition from one MAC frame sequence to another will be independent of time since $H(p)$ is time homogeneous. Also, due to negligible 'STOP' message traffic on the uplink (negligible collision between RTS frames), every successful or unsuccessful MAC frame transmission sequence occurs independently of all the preceding MAC frame transmission sequences. We can thus assume the FCP transmission process, between the AP and any given node, being represented by a Markov renewal reward process whose renewal instants J_{t_i} are embedded at the time instants t_i of completion of an i^{th} successful or unsuccessful MAC frame transmission sequence. The reward W_i for each i^{th} renewal cycle is denoted by whether the MAC frame was successfully transmitted or not. In order to formalize this notion, we construct the corresponding discrete time Markov chain as follows. First, we assume that all FCPs are of identical size. Though the 802.11 standard permits packet fragmentation into MAC frames, we assume that *packet size is equal to a single MAC frame size* and that the whole FCP is transmitted as a single MAC frame. Recall that a node must receive at least D FCPs in order to recover the original file of N_p packets. Now, at the beginning of each successful or unsuccessful MAC frame transmission sequence between the AP and a given node, let r_k, $k \in \mathcal{K}$, be the PHY rate that will be employed during this transmission sequence and let n denote the total number of FCPs or MAC frames that have already been received by the node. Thus $n=0$ at the beginning of file transfer. Then, let $S_k^{(n)}$, $k \in \mathcal{K}$, denote the *state* of the transmission process between the AP and the wireless node at the beginning of any successful or unsuccessful transmission sequence. With some abuse of notation let $S^{(n)}$ denote the collection $\bigcup_{k\in\mathcal{K}}\{S_k^{(n)}\}$ of all states corresponding to a given n. Now, from one transmission sequence to the next, the state of the process evolves from $S_k^{(n)}$ to either $S_j^{(n)}$ (in case of unsuccessful transmission) or $S_l^{(n+1)}$ (in case of successful transmission) for some $j,l \in \mathcal{K}$. Figure-4 shows a single step in such a Markov chain evolution

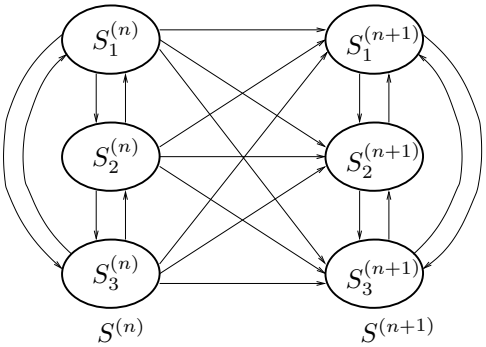

Fig. 4. Single step of Markov chain for $K=3$

for $K=3$. For any general K the state transition matrix T, for the complete Markov chain until all D FCPs have been successfully received, can be formulated as,

$$T = \begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ D-1 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & & \cdots & & D \end{array} \\ \begin{bmatrix} T_f & T_s & & \mathbf{0} & & \\ & T_f & T_s & & & \\ & & \ddots & \ddots & & \\ & \mathbf{0} & & T_f & T_s & \end{bmatrix}_{D\times(D+1)} \end{array}$$

where,

$$T_f = \begin{bmatrix} t_{1,1}^{(f)} & \cdots & t_{1,K}^{(f)} \\ \vdots & \ddots & \vdots \\ t_{K,1}^{(f)} & \cdots & t_{K,K}^{(f)} \end{bmatrix}_{K\times K}$$

and

$$T_s = \begin{bmatrix} t_{1,1}^{(s)} & \cdots & t_{1,K}^{(s)} \\ \vdots & \ddots & \vdots \\ t_{K,1}^{(s)} & \cdots & t_{K,K}^{(s)} \end{bmatrix}_{K\times K}$$

where,

$$t_{k,j}^{(f)} = p_f(k)h_{k,j}(p) \quad \& \quad t_{k,l}^{(s)} = p_s(k)h_{k,l}(p).$$

In the preceding two formulae, $t_{k,j}^{(f)}$ denotes the probability that the state of the process evolves from $S_k^{(n)}$ to $S_j^{(n)}$ due to unsuccessful transmission, $t_{k,l}^{(s)}$ denotes the probability that the state evolves from $S_k^{(n)}$ to $S_l^{(n+1)}$ due to successful transmission, $p_s(k)$ denotes the probability that the channel conditions during a transmission sequence remain suitable for a successful transmission and $p_f(k)$ denotes the probability that the channel conditions during a transmission sequence lead to a failure of the MAC frame transmission. We will soon observe at the end of this sub-section how the two probabilities $p_s(k)$ and $p_f(k)$ are a function of some $k \in \mathcal{K}$.

Now, note that the collection $\bigcup_{k\in\mathcal{K}}\{S_k^{(D)}\}$ corresponding to $n=D$ forms a set of absorption states. When all D FCPs have been successfully transmitted the FCP transmission process terminates in one of the states $\{S_l^{(D)},\ l \in \mathcal{K}\}$. Thus for the purpose of computing the stationary probabilities, $\{\pi_l, l \in \mathcal{K}\}$, we may consider only the absorption states and ignore all other states which would lead to a computational complexity of only $O(K^2)$ instead of $O(K^2D)$.

Now, let T_R, T_C and T_A denote the times (in slots) required to transmit an RTS, CTS and ACK frame, respectively. Also, let d be the size (in bits) of a single FCP or MAC frame, which are assumed to be of identical size as mentioned earlier. Then, with the channel model described in the foregoing discussion, we can now explicitly define the probabilities $p_s(k)$ and $p_f(k)$ for some $k \in \mathcal{K}$ as,

$$p_s(k) = p_g^{T_R+T_C+\left\lceil \frac{d}{r_k} \right\rceil + T_A}, p_f(k) = p_f^{(R)} + p_f^{(C)} + p_f^{(D)}(k) + p_f^{(A)}(k), \tag{1}$$

$$\text{where,} \quad p_f^{(R)} = 1 - p_g^{T_R}, \quad p_f^{(C)} = p_g^{T_R}\left(1 - p_g^{T_C}\right),$$

$$p_f^{(D)}(k) = p_g^{T_R+T_C}\left(1 - p_g^{\left\lceil \frac{d}{r_k} \right\rceil}\right), p_f^{(A)}(k) = p_g^{T_R+T_C+\left\lceil \frac{d}{r_k} \right\rceil}\left(1 - p_g^{T_A}\right). \tag{2}$$

IV. Mean Downlink Throughput

We now derive an explicit expression for the mean downlink throughput achieved by a single node. The throughput will be calculated as a function of the given channel conditions, $p = (p_g, p_b)$. However, since we are concerned with only successful MAC frame transmissions, i.e., the *goodput*, it turns out that the throughput (or goodput) will only be a function of p_g. The value of p_g itself however, may or may not be related to p_b depending on the channel conditions. Recall the renewal reward process formulation of the MAC frame transmission sequences from the AP to a given node (Section-III-D). Let $L_i \triangleq t_i - t_{i-1}$ denote the length (in slots) of i^{th} renewal cycle. Applying the renewal reward theorem (Theorem D.15 in [20]) to our formulation, we can state that the mean downlink throughput, Θ, from AP to any wireless node is given by,

$$\Theta = \frac{\lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} W_i}{\lim_{m\to\infty} \frac{1}{m}\sum_{i=1}^{m} L_i} = \frac{E[W]}{E[L]}, \tag{3}$$

where, $E[W]$ is the expected reward over a renewal cycle in terms of number of bits of data transmitted and $E[L]$ is the expected length of a renewal cycle.

In order to compute the throughput we first compute $E[L]$ for a given value of p_g as follows. Using Equations 1 and 2, define the following entities,

$$p_s = E_{\mathcal{K}}[p_s(k)] = p_g^{T_R+T_C+T_A} E_{\mathcal{K}}[p_g^{\left\lceil \frac{d}{r_k} \right\rceil}], \tag{4}$$

$$p_f^{(D)} = E_{\mathcal{K}}[p_f^{(D)}(k)] = p_g^{T_R+T_C}\left(1 - E_{\mathcal{K}}[p_g^{\left\lceil \frac{d}{r_k} \right\rceil}]\right), \tag{5}$$

$$p_f^{(A)} = E_{\mathcal{K}}[p_f^{(A)}(k)] = p_g^{T_R+T_C} E_{\mathcal{K}}[p_g^{\left\lceil \frac{d}{r_k} \right\rceil}]\left(1 - p_g^{T_A}\right), \tag{6}$$

where, $E_{\mathcal{K}}[\cdot]$ is the expectation operator over the index set $\mathcal{K}$. This expectation may be computed using the stationary probabilities $\{\pi_l,\ l \in \mathcal{K}\}$ mentioned earlier in Section III-D. Now, from Figure-1 we can see that $E[L]$ can be derived as,

$$E[L] = \mathbb{E}(\text{length of renewal cycle}) = p_s\left(E_{\mathcal{K}}\left[\left\lceil \frac{d}{r_k} \right\rceil\right] + T_{ov}\right) + p_f^{(R)} T_{ov}^{(R)} + p_f^{(C)} T_{ov}^{(C)} + p_f^{(D)}\left(E_{\mathcal{K}}\left[\left\lceil \frac{d}{r_k} \right\rceil\right] + T_{ov}^{(D)}\right) + p_f^{(A)}\left(E_{\mathcal{K}}\left[\left\lceil \frac{d}{r_k} \right\rceil\right] + T_{ov}^{(A)}\right), \tag{7}$$

where,

$$T_{ov}^{(R)} = T_{DIFS} + E[T_B] + T_R + T_{SIFS} = \hat{T}_{ov}^{(R)} + E[T_B], \tag{8}$$

$$T_{ov}^{(C)} = T_{DIFS} + E[T_B] + T_R + T_C + T_{SIFS} = \hat{T}_{ov}^{(C)} + E[T_B], \tag{9}$$

$$T_{ov}^{(D)} = T_{DIFS} + E[T_B] + T_R + T_C + 3T_{SIFS} = \hat{T}_{ov}^{(D)} + E[T_B], \tag{10}$$

$$T_{ov}^{(A)} = T_{DIFS} + E[T_B] + T_R + T_C + T_A + 3T_{SIFS} = \hat{T}_{ov}^{(A)} + E[T_B], \tag{11}$$

$$T_{ov} = T_{ov}^{(A)}. \tag{12}$$

In the preceding formulae, T_{DIFS} and T_{SIFS} denote the DIFS and SIFS times (in slots), respectively. $E[T_B]$ denotes the expected length of back-off duration as discussed earlier in the beginning of Section-III. $T_{ov}^{(\cdot)}$ denotes the contribution from MAC layer entities to the overhead caused by an *unsuccessful* transmission of the corresponding RTS, CTS, MAC data or ACK frame. T_{ov} denotes the same for a *successful* transmission of the MAC data frame. An extra term of T_{SIFS} has been included in Equations 8 and 10 since in order to detect the preceding frame's loss, the AP must wait (for SIFS duration) till the time instant when the wireless node starts to send the next frame in the sequence.

Next, for computing $E[W]$ for a given value of p_g, observe that an FCP of size d bits (or an original un-coded packet of size $\frac{d}{1+\epsilon}$ bits) is received by a wireless node only when the corresponding MAC frame is successfully transmitted over a renewal cycle. From this observation the expected reward over a renewal cycle is given by,

$$E[W] = \frac{p_s\, d}{1+\epsilon}. \tag{13}$$

Plugging Equations 7 and 13 in Equation 3 one can now compute the mean downlink throughput.

V. Validation of the Model

The goal of this section is to validate our model and expression for mean downlink throughput using numerical analysis and *ns2* simulations.

A. Implementation Details

MATLAB software was used for all numerical computations and the values used for various system parameters are as follows: 802.11 slot $= 20\mu s$, $T_R = 4.5$ slots, $T_C = 2.8$ slots, $T_A = 2.8$ slots, $T_{DIFS} = 2.5$ slots, $T_{SIFS} = 0.5$ slots, $CW = 64$ slots, $E[T_B] = 31.5$ slots, $d = 1024$ bytes, $K = 3$, $r_1 = 6$ Mbits/s, $r_2 = 12$ Mbits/s, $r_3 = 24$ Mbits/s. The WLAN topology considered for *ns2* simulations was identical to the one proposed for modeling analysis in Section III. Mobility of wireless nodes was not introduced in the simulation scenarios. The configuration of AP node was modified to implement the Adaptive Auto Rate Fallback (AARF) rate control algorithm

and its parameters used in all our simulation scenarios are identical to those specified in Table-2 in [11]. It can be shown that these parameters for $K = 3$ give rise to the following PHY rate transition matrix that we used in our simulations,

$$H = \begin{bmatrix} 0.63\,(1-p_g) & 0.37\,p_g & 0 \\ 0.63\,(1-p_g) & 0.0925\,p_g/2 & 0.2775\,p_g/2 \\ 0 & 0.63\,(1-p_g) & 0.37\,p_g \end{bmatrix}.$$

The FCT protocol defined previously in Section-II was implemented by modifying the `Agent/UDP` class module in `apps/udp.h` and `apps/udp.cc` files in *ns2*. To be specific, it was modified to support a new `advanceby()` public function (similar to the one present in `Agent/TCP` class module) and a new `sendSTOP()` private function. The `advanceby()` function can be called in a TCL script to let a `MAC/802_11` node (AP) to send dummy fountain coded packets at a pre-specified constant bit rate. The `sendSTOP()` function is called in the `recv()` function to automatically send a 'STOP' message to the sender (AP) as soon as a pre-specified number of FCPs have been successfully received at receiver side. In turn, on receipt of this 'STOP' message the sender terminates its current file session.

The wireless medium in all simulation scenarios was modeled by a two-ray ground radio propagation model and all nodes were configured to use a single antenna. Similar to the analytical model, a two state Markov channel error model was used on a *per slot basis* for the simulation scenarios. Different values for the probability p_g of good channel state for different scenarios shall be explicitly specified later. Each result was obtained by executing the simulation scenario for about 50 runs in order to gather sufficient statistics for calculation of mean values.

B. Comparison

We compare here the results of numerical analysis with those of simulations. Figure-5 shows mean downlink throughputs from AP to a single node for a file of size 769200 bytes ($N_p \approx 769.2$ packets), a fountain code with design parameter $\epsilon = 0.3$, $K = 3$, $E[T_B] = 33$ slots, $d = 1500$ bytes and a wireless channel for which $p_g = 0.99$. With a fountain code of design parameter $\epsilon = 0.3$, a file of size ≈ 769200 bytes can be encoded such that the wireless node must receive at least approximately $D = 1000$ FCPs in order to decode the original file. Also note that $p_g = 0.99$ *does not necessarily represent highly favorable channel conditions* since state transitions in the Markov channel error model occur on a *per slot basis*. Therefore, the channel must continue to remain in good state for a couple of tens of slots successively, for the MAC transmission to be successful. For instance, with $p_g = 0.99$ and a complete MAC transmission sequence taking 60 slots (not including back-off) for a given rate control algorithm, the successful MAC transmission probability would be only $p_s = 0.99^{60} \approx 0.55$.

In Figure-5 we can observe a very good match between the simulation and analysis results. Figure-6 shows mean downlink throughputs for another scenario of $D = 50$ FCPs and the remaining parameters identical to that of Figure-5.

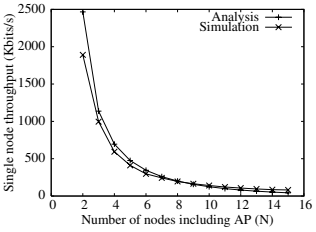

Fig. 5. $D = 1000$

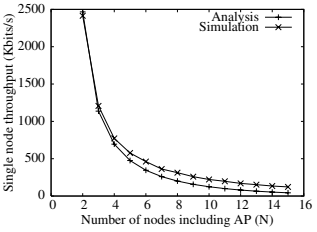

Fig. 6. $D = 50$

We again observe a fairly good match of the results from simulations with those from analysis. For higher values of N the throughput results tend to deviate marginally from each other. This may be attributed to the fact that we ignored the modeling of 'STOP' messages on the uplink and assumed that the collision between RTS frames on uplink and downlink is negligible for large file sizes (see Section III-D). Though file size in Figure-5 was indeed large ($N_p \approx 769.2$ packets), in Figure-6 it is relatively small ($N_p \approx 38.5$ packets), leading to the slight deviation in results. Moreover, with increasing number of wireless nodes (N) the collision probability will only become non-negligible.

Before concluding this section we would like to mention that apart from the scenarios discussed above, we have carried out numerical analysis and simulations for additional scenarios, i.e., for different values of D, ϵ, K, d and p_g. In all these additional scenarios, a fairly good match between analysis and simulation results was observed as for those discussed above. It was also observed in all simulation results that the AP achieved nearly equal downlink throughputs for each node in the WLAN cell. Some of these additional scenarios are presented in the following section. As a conclusion to this section, we may state that the simulation results presented here demonstrate that our modeling analysis captures fairly accurately the performance of the FCT protocol in a WLAN last-hop link.

VI. Performance Analysis

Having validated the model, we now proceed towards analyzing performance of the obtained throughput metric w.r.t. the various system parameters. Unless otherwise stated, numerical values for the parameters used in this section are identical to those specified in Section V-A.

Figure 10 shows a plot of throughput (obtained by plugging Equations 7 & 13 in 3) as a function of the fountain code parameter, ϵ, for different values of N. For a given fountain code, lower is the value of ϵ, higher is its efficiency. But higher efficiency comes at the cost of a more complex coding-decoding algorithm and hence increased delay. In Figure 10 it is seen that employing a fountain code with higher efficiency may result in better throughput performance *only for low values of* N. As value of N increases, throughput performance becomes insensitive to the choice of ϵ. Observation of this trade-off can be of considerable practical importance as it can be exploited to *dynamically* select an appropriate fountain code, based on an estimate of N by the AP. Instead of choosing the same value of ϵ for transmission of each file, an algorithm

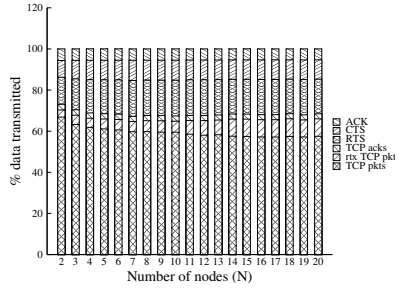

Fig. 7. % data with TCP, $p_g = 0.99$

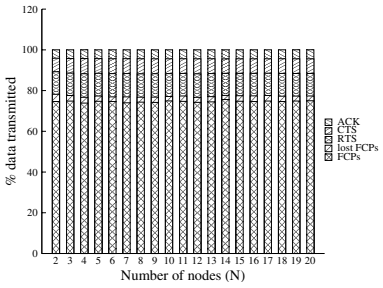

Fig. 8. % data with FCT, $\epsilon = 0.3$ & $p_g = 0.99$

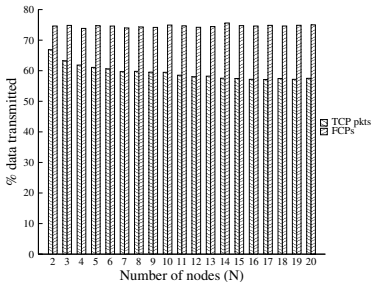

Fig. 9. Comparison of TCP packets and FCPs

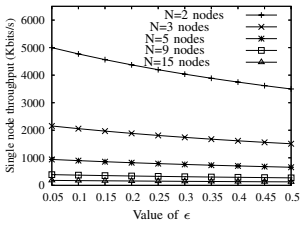

Fig. 10. $D = 20$, $K = 3$, $d = 1024$ bytes and $p_g = 0.99$

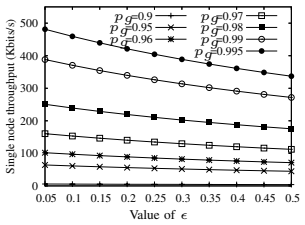

Fig. 11. $N = 9$, $\beta = 0.075$, $D = 20$, $K = 3$ and $d = 1024$ bytes

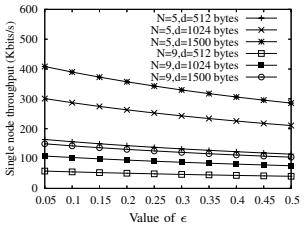

Fig. 12. $D = 20$, $K = 3$, $\beta(5) = 0.125$, $\beta(9) = 0.075$ and $p_g = 0.99$

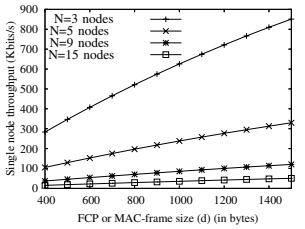

Fig. 13. $D = 20$, $K = 3$, $\epsilon = 0.3$ and $p_g = 0.99$

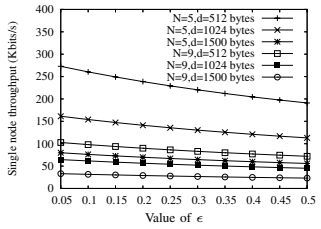

Fig. 14. $D = 20$, $K = 3$, $\beta(5) = 0.125$, $\beta(9) = 0.075$and $p_g = 0.95$

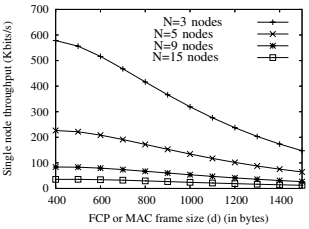

Fig. 15. $D = 20$, $K = 3$, $\epsilon = 0.3$ and $p_g = 0.95$

can be devised so that the AP uses a lower value of ϵ when N is small (for improved throughput performance) and a higher value when N is relatively large (to avoid non-beneficial increased delay due to a more complex coding-decoding).

Figure 11 shows similar characteristics as those observed in Figure 10, but for dependance on channel conditions (i.e., p_g). Selecting a fountain code with higher efficiency can be beneficial in terms of throughput *only when good channel conditions prevail*, i.e., for high values of p_g or $p_s = 0.99^{60} \approx 0.55$. If channel conditions are bad (low values of p_g or $p_s = 0.95^{60} \approx 0.05$), throughput performance is insensitive to the choice of the fountain code (i.e., ϵ). Again, this observation can be of considerable practical importance as it can be exploited to dynamically select an appropriate fountain code based on an estimate of the channel conditions by AP.

At last, let us look at the dependance of throughput on the FCP size (or MAC-frame size; see Section III-D). As an example for $N = 5$ and 9, Figure 12 illustrates that for any given fountain code and number of nodes N, increasing the FCP size, d, results in an increase in the throughput. Figure 13 confirms this illustration for a particular value of $\epsilon = 0.3$ and different values of $N = 3, 5, 9, 15$. However, this observation is valid only for fairly good channel conditions with $p_g = 0.99$, i.e., $p_s \approx 0.55$. In Figures 14 and 15 we observe that for bad channel conditions with $p_g = 0.95$, i.e., $p_s \approx 0.05$, the trend reverses as compared to Figures 12 and 13. For any given fountain code and number of nodes N, increasing the FCP size, d, now results in a decrease in the throughput. Again this result is of considerable practical importance from which we may conclude that during good channel conditions FCPs of largest possible size must be transmitted and during bad channel conditions it is preferable to transmit FCPs of smallest possible size.

VII. Comparison of FCT with TCP

As per our discussion in Section I, the FCT protocol may be a better alternative to TCP in an 802.11 WLAN cell since it abandons the need for TCP ack like traffic on reverse link. In the forthcoming discussion we explicitly quantify this performance improvement in terms of percentage of useful data transmitted. Using settings similar to the ones described in Section V-A, we performed *ns2* simulations to measure the percentage of un-coded N_p *useful data* packets, successfully transmitted in a WLAN cell, among all kinds of packets and MAC frames. This percentage measure has been carried out for downlink transfers using TCP and FCT (with $\epsilon = 0.3$), separately. Figures 7 and 8 show simulation results for a long file transfer that lasts up to 3 hours over a fairly good wireless channel with $p_g = 0.99$ or $p_s \approx 0.55$. In the figures, % of all kinds of packets and MAC frames is plotted against the number of nodes N (including AP). Figure 9 shows a comparison between the percentages of useful data packets for TCP and FCT. It can be clearly observed that FCT *outperforms* TCP. Moreover, the performance improvement increases, though marginally, with increasing number of nodes. We may thus infer that for a suitable design parameter (ϵ) of a fountain code and fairly good channel characteristics ($p_g = 0.99$ or $p_s \approx 0.55$), there is greater incentive to switch to FCT when number of nodes in a WLAN cell are

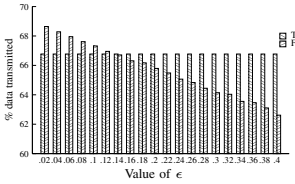

Fig. 16. % of data, $N = 2$, $p_g = 0.95$

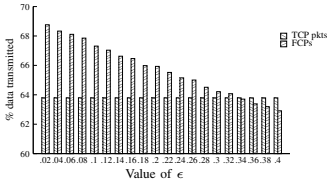

Fig. 17. % of data, $N = 3$, $p_g = 0.95$

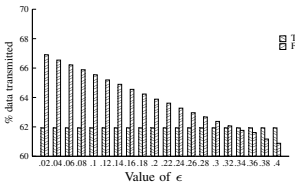

Fig. 18. % of data, $N = 4$, $p_g = 0.95$

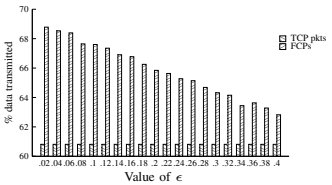

Fig. 19. % of data, $N = 5$, $p_g = 0.95$

relatively high. However, this may not be the case when channel conditions are bad and we discuss this issue next.

Figures 16, 17, 18 and 19 show for $N = 2$, 3, 4 and 5, respectively, the percentage of file data successfully transmitted using TCP and different FCTs represented by varying values of ϵ. The wireless channel characteristics for these results pertain to poor channel conditions, i.e., $p_g = 0.95$ or $p_s \approx 0.05$. Consider e.g., an FCT with $\epsilon = 0.2$. We observe in the figures that for $N = 2$ TCP performs better than the FCT, whereas for values of N greater than 2 TCP performs worse than the FCT. Compare this FCT with another FCT having $\epsilon = 0.4$. We now observe that TCP performs better than this new FCT ($\epsilon = 0.4$) for $N =$ 2,3 and 4 and vice-versa holds true for $N = 5$. For yet another FCT with $\epsilon = 0.1$, we observe that it performs better than TCP for all values of $N = 2$, 3, 4 and 5.

The foregoing discussion helps us to deduce that an FCT may perform better or worse than TCP depending on its design parameter ϵ, the number of nodes N and the wireless channel characteristics. This trade-off between the performance of an FCT (compared to TCP) and the extra ϵN_p redundant packets, can be intuitively understood to be related to the underlying wireless channel characteristics and number of nodes N. This observation can be practically exploited in designing a dynamic *protocol selection* algorithm that can optimally and/or adaptively select one of the TCP or FCT protocols for downlink transmission in a WLAN.

VIII. Conclusion

We have proposed a Markovian stochastic model to analyze the performance of a simple Fountain Codes based Transport (FCT) in an 802.11 WLAN setting where the AP is allowed to employ a generic rate control algorithm. The results of this study clearly demonstrate that choosing a fountain code with higher efficiency (lower value of ϵ) may improve the throughput performance only when number of wireless nodes (N) are low or when good channel conditions prevail. Next, for a given fountain code and estimate of N, during good channel conditions the size of FCP must be chosen as large as possible and during bad channel conditions it must be chosen as small as possible. Finally, through a brief simulation study we have illustrated that FCT may perform better or worse than TCP depending on the fountain code parameter, ϵ, the number of nodes in a WLAN cell and the wireless channel conditions.

References

[1] D. J. C. MacKay. Fountain Codes. *IEEE Proceedings Communications*, 152(6):1062-1068, December 2005.

[2] Byers et al. A Digital Fountain Approach to the Reliable Distribution of Bulk Data. *IEEE Journal on Selected Areas in Communications*, 20(8):1528-1540, 2002.

[3] M. Luby. LT codes. *43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[4] A. Shokrollahi. Raptor Codes. *Unpublished manuscript*, 2003.

[5] Kumar et al. Fountain Broadcast for Wireless Networks. *Second International Workshop on Networked Sensing Systems (INSS)*, 2005.

[6] Lakshman et al. The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss. *IEEE/ACM Transactions on Networking*, 5(3):336-350, June 1997.

[7] Lefevre et al. Understanding TCP's Behavior over Wireless Links. *Symposium on Communications and Vehicular Technology*, October 2000.

[8] Tian et al. TCP in Wireless Environments: Problems and Solutions. *IEEE (Radio) Communications Magazine*, 43(3):S27-S32, March 2005.

[9] Xylomenos et al. TCP and UDP performance over a wireless LAN. *IEEE Infocom*, March 1999.

[10] Lopez et al. A Game Theoretic Analysis of Protocols Based on Fountain Codes. *10th IEEE Symposium on Computers and Comm.*, 2005.

[11] Lacage et al. IEEE 802.11 Rate Adaptation: A Practical Approach. *ACM International Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, Venice, October, 2004.

[12] M. Mitzenmacher. Digital Fountains: A Survey and Look Forward, *IEEE Information Theory Workshop*, 24-29 October 2004.

[13] Lehr et al. Wireless Internet Access: 3G vs. WiFi. *White Paper, MIT*, August 2002.

[14] M. Luby. Information additive code generator and decoder for comm. systems. *U.S. Patent 6,307,487; 6,373,406; 6,614,366*, 2001-03.

[15] S. Yang. Experimental Comparison between TCP and Digital Fountain over High Performance Network on UK Grid. *Project Report, University of Manchester*, March 2005.

[16] Usman et al. A Testbed for Assessment of Fountain Codes for Wireless Channels. *European Wireless*, Nicosia, Cyprus, April 2005.

[17] Cali et al. Dynamic tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit. *IEEE Trans. on Networking*, 8(6), Dec. 2000.

[18] Kherani et al. Throughput Analysis of TCP in Multi-Hop Wireless Networks with IEEE 802.11 MAC. *IEEE WCNC*, USA, March 2004.

[19] Kherani et al. Performance Improvement of TCP with Delayed ACKs in IEEE 802.11 Wireless LANs. *IEEE WCNC*, USA, March 2004.

[20] Kumar et al. Communication Networking: An Analytical Approach. *Morgan Kaufman Series in Networking*, May 2004.

[21] Hu et al. An adaptive p-persistent 802.11 MAC scheme to achieve maximum channel throughput and QoS provisioning. *IEEE WCNC*, 2006.

[22] Miorandi et al. A Queueing Model for HTTP Traffic over IEEE 802.11 WLANs. *Computer Networks*, 50(1):63-79, Jan. 2006.

[23] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE JSAC*, 18(3):535-547, Mar. 2000.

[24] Kumar et al. New insights from a fixed point analysis of single cell IEEE 802.11 WLANs. *Proceedings of IEEE Infocom*, USA, Mar. 2005.

[25] M. F. Neuts. Matrix-Geometric Solutions in Stochastic Models - An Algorithmic Approach. *Johns Hopkins University Press*, Baltimore, 1981.

[26] Sikdar et al. An Integrated Model for the Latency and Steady State Throughput of TCP Connections. *Performance Evaluation Journal*, 46(2001):139-154, 2001.