Modeling Multimedia and Heterogeneous traffic in DCF-Based Networks

Tamir Husain, Marco Levorato, Christopher A. Wood
Department of Computer Science
University of California Irvine
Email: thusain@uci.edu, levorato@uci.edu, woodc1@uci.edu

***Abstract*—We present a highly extensible Markov model and implementation for modeling multimedia and heterogeneous traffic in networks based on the 802.11 DCF access scheme. Our parameterized model can be tuned according to the packet interarrival time, packet buffer arrival probability, packet length, and interarrival wait probability associated with any type of application traffic. We show how to simulate truly heterogeneous traffic by instantiating multiple versions of this model in the context of a single network simulator. This enables one to study the performance of the DCF access scheme with respect to heterogeneous traffic in a true multi-node system.**

I. Introduction

Wireless networks have become pervasive in modern computer and communication systems. The widespread adoption of mobile and personal computing devices, such as smartphones and tablets, continues to drive commercial investment in high performance wireless networking technologies. Concurrently, the type of traffic traversing these networks has also evolved – divergently – an increasing rate. Every kind of information from textual data to video stream segments now traverses wireless networks. Society's dependence on WLAN technologies such as the 802.11 protocol suites have cultivated a massive amount of analytical and experimental research studying their performance.

To date, a major portion of this research has focused on the 802.11 random access control protocol – the distributed control function (DCF). This work was pioneered by Bianchi's original Markov model for the DCF function [4], which stimulated several advancements to his simplistic model [3] and numerous applications in the field [7], [5]. Many of these models and empirical studies represent a network of nodes by a single collision parameter. In particular, there is a lack of work combining these models together beyond a single parameter. Furthermore, many of these works study the performance of the DCF function (with respect to individual and average system throughput) for single types of traffic. True multi-node models that study the interaction of different traffic models have not been studied.

The goal of this work is to present a framework in which such interaction can be studied. To this end, we present a series of extensions to Bianchi's original DCF Markov model to emulate arbitrary types of traffic. In particular, we provide extensions to support arbitrary packet length and variable packet inter-arrival time. We then instantiate individual instances of these Markov models that emulate the behavior of the DCF access scheme by tailoring each of these parameters, e.g., packet length and interarrival time, using realistic values drawn distributions representative of modern application traffic. We then show how to combine these model instances to study the performance of the DCF function with respect to heterogeneous traffic in a true multi-node system.

The rest of this paper is organized as follows. In Section II, we discuss several unique types of application traffic that often co-exist in modern networks. Section III then presents the extensions included in our more flexible and expressive model. We conclude with a discussion of how to implement multi-node simulators using our models, and a summary of related work.

II. Heterogeneous Traffic Models

Today's computer and communication networks are being used to transfer increasingly heterogeneous traffic between parties. In particular, file downloads, standard web browsing, video streaming, and client-server video game traffic are four very common types of traffic that dominate the Internet traffic today. In this section, we describe the characteristics of each of these traffic types. We use a variety of parameters to describe these types of traffic. Specifically, we focus on packet size, interarrival time, traffic saturation, and burstiness.

File Download Traffic: File downloads are elastic, meaning that they must be reliable but are also tolerant to random delays. Applications that *generate* file download traffic usually do so with long bursts of packet arrivals and long interarrival times [8]. File downloads are sufficiently random that we do not consider a fixed distribution for any parameters. Rather, we choose these at random by our own free will.

Web Browsing Traffic: Web browsing traffic is quite diverse in packet size, interarrival time, and burstiness. Mah [9] studied various web browsing traces to develop statistics for these characteristics. In particular, it was found that these characteristics, such as content request and reply lengths (packet size) follow a Zipf distribution. For simplicity, we assume that web browsing traffic has random packet sizes sample from a Zipf distribution, deterministic interarrival time, and probabilistic packet arrival in the buffer. This enables us to use a slightly modified version of the nonsaturated DCF model (described in Section III-B) to study this traffic.

Video Streaming Traffic: In this work we consider MPEG video encoding, though our modeling techniques are can be generalized to any similar multimedia streaming protocol. MPEG video traffic streams consist of I, P, and B packets [14], [13]. The video stream is divided into runs of GOPs (Group of Pictures). The composition of a GOP is determined by two constant parameters, the full frame distance (distance between two I-Frames) and the anchor frame distance (distance between two P-Frames). A GOP is always headed by an I (intraframe) Frame encoded without any other temporal references). Following the I-Frame will come a series of groups of B frames tailed with a P-Frame. The anchor frame distance determines the composition of these groups (a value of 3 would result in BBP groups). P (predicted) Frames are encoded based off of previously decoded frames. B (bi-predicted) Frames , based off both previously decoded frames and frames in the future. B-Frames achieve the highest amount of compression.

III. Extensible Markov Models

The 802.11 DCF [6] is at the core of this work. It is a simplistic random access scheme based on the carrier sense multiple access with collision avoidance (CSMA/CA) protocol. Failed packets are retried according to a binary exponential backoff rule. At each packet transmission, the backoff is uniformly in the range $(0, w-1)$, where w is the length of the "contention window" and is directly dependent on the number of failed attempts for the packet thus far. The window w is set to W_{min} to begin, and upon every failure, the backoff counter is doubled. At stage i, the backoff timer is $2^i W_{min}$. The maximum backoff is bounded by $W_{max} = 2^m W_{min}$. Selections of W_{min} and W_{max} are dependent upon the physical layer specifications in the 802.11 standard [6], [2]. For example, frequency hopping spread spectrum calls for $W_{min} = 16$ and $W_{max} = 1024$ $(m = 6)$.

In this section we describe the necessary evolutions of Bianchi's original Markov model to support parameterized packet length, interarrival time, and buffer saturation. Ultimately, the extensions enable us to parameterize the following characteristics: interarrival time, packet size, probability of a new packet arriving (i.e., buffer saturation), and the probability of a node waiting for a non-zero interarrival time. The usage of these parameters to emulate web browsing, file downloads, and multimedia traffic is elaborated upon in Section **??**.

A. Original DCF Model

A simple Markov model for the 802.11 DCF function was proposed in the seminal work by Bianchi [2]. The system state is described by two discrete-time stochastic processes $b(t)$ and $s(t)$: $b(t)$ represents the backoff *time counter* and $s(t)$ represents the backoff *stage* for a given station or node, each of which is advanced at each time step (i.e., from t to $t+1$). Each node is equipped with a *saturated* buffer of packets to transmit; there is never a case where a node has to wait for a packet to arrive in the buffer before attempting a transmission. Also, instead of modeling multiple nodes simultaneously to accurately assess the probability of collision, this model assumes a constant probability of collision p – the conditional collision probability – at each time slot. In the real world, this probability is a function of the network and environmental interferences, e.g., shadowing, fading, etc. However, by making this a constant and therefore independent from other nodes, the model simplifies to capturing the dynamics of the bidimensional process $\{b(t), s(t)\}$ as a discrete-time Markov chain, shown in Figure 2.

The only non-null transition probabilities in this Markov chain are given below:

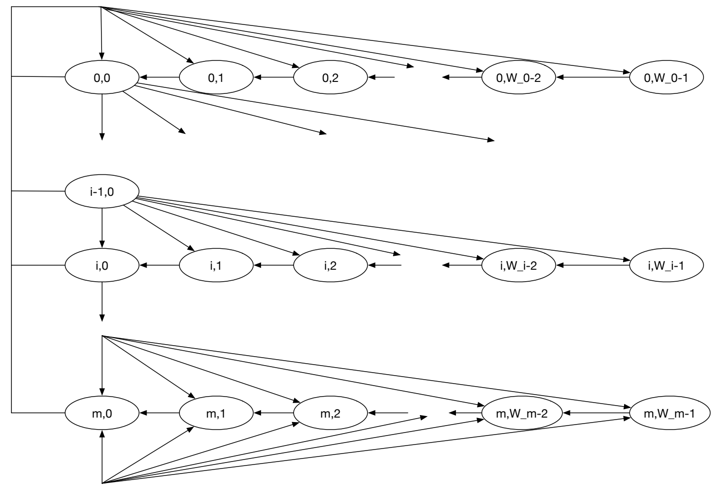

Fig. 2. The original saturated DCF Markov model [2].

$$
\begin{aligned}
&\Pr[(i,k)|(i,k+1)] = 1, k \in [0, W_i - 2], i \in [0, m] \\
&\Pr[(0,k)|(i,0)] = (1-p)/W_0, k \in [0, W_0 - 1], i \in [0, m] \\
&\Pr[(i,k)|(i-1,k)] = p/W_i, k \in [0, W_i - 1], i \in [1, m] \\
&\Pr[(m,k)|(m,0)] = p/W_m, k \in [0, W_m - 1]
\end{aligned}
$$

B. Limited and Diverse Traffic

While accurate, the original DCF model is constrained in that it assumes homogeneous traffic and a saturated stream of packets for transmission. Malone et al. [10] presented a modified version of Bianchi's model that captures non-saturated traffic loads. The essential idea behind their variant is that there exists a constant packet arrival probability q at each time slot, much like there exists a constant collision probability p. If a node successfully transmits a packet and the buffer is not-empty, the state of the system proceeds as normal. Conversely, if a packet is not ready for transmission, then the chain enters a stage known as "postbackoff," denoted $(0, k)_e$ for $k \in [0, W_0 - 1]$. A node may remain in this set of states indefinitely until a packet arrives and the channel is idle.

To capture the mechanics of the DCF function in this condition, the postbackoff set of stages are nearly identical to backoff stage $i = 0$. If a packet arrives at any time when the system is in a postbackoff state, it immediately transitions into the backoff stage $i = 0$, with a decremented backoff timer. However, if the postbackoff timer reaches 0, where the postbackoff is said to be complete, the node will stay in this state until a packet arrives with probability q. Once a packet arrives in state $(0, 0)_e$ with probability q, there are three possible outcomes: (1) the packet is transmitted successfully, a collision occurs with probability p, or the medium (channel) is sensed as busy with probability $1 - P_{idle} = p$.

With the addition of the postbackoff states S', the Markov chain itself is now a multidimensional process $\{b(t), s(t), S'\}$. The new state transition probabilities needed to capture the transitions between these multiple dimensions are below.

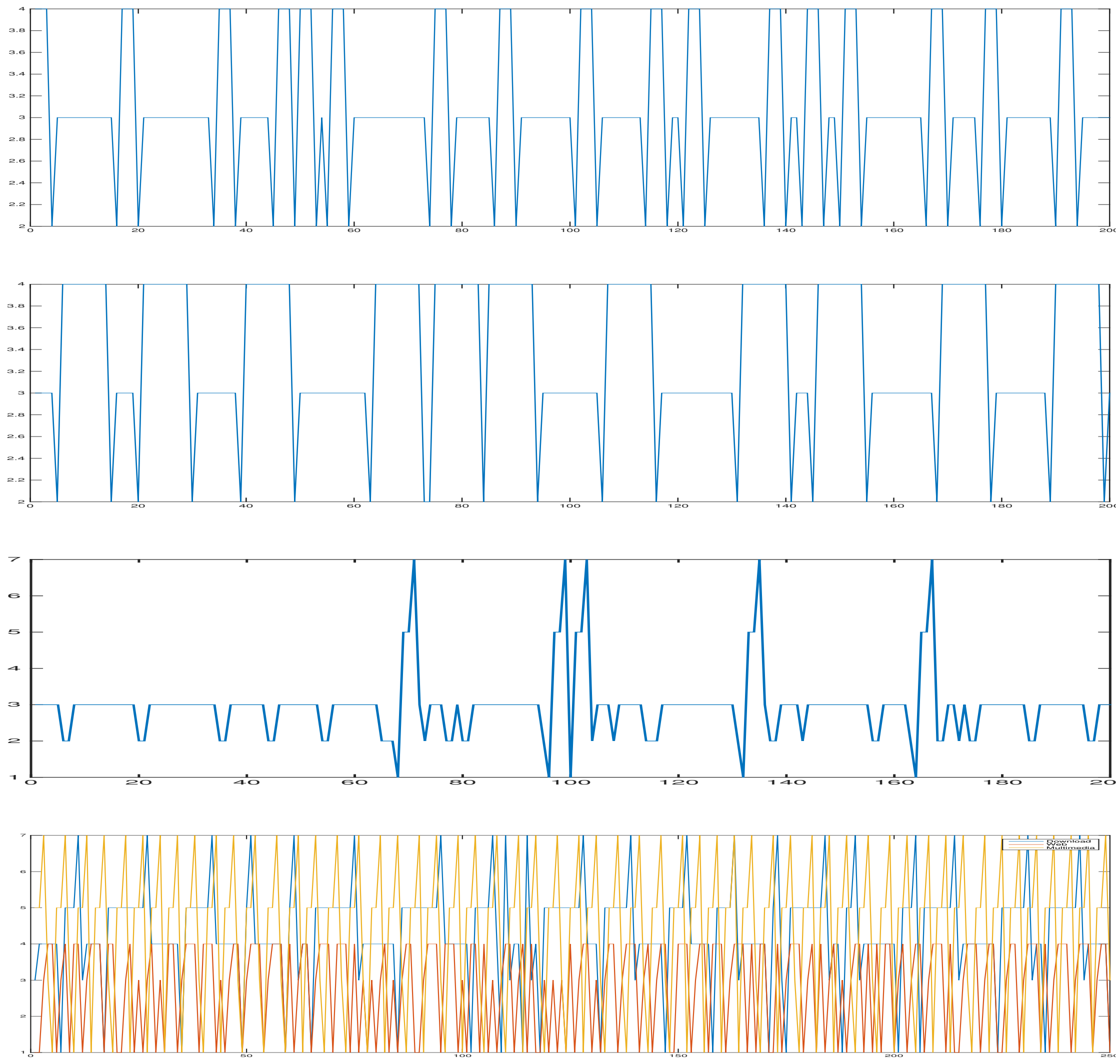

Fig. 1. Individual system transition traces for (1) web browsing, (2) file download, and (3) video streaming traffic. The bottom trace shows the superposition of all the above traces. The x axis is the discrete time in the system, and the y axis is the state. For completeness, state 5 is interarrival, 4 is the packet size calculation, 3 is a waiting state (e.g., backoff, postbackoff), 2 is a failure transmission, and 1 is a successful transmission.

$$
\begin{aligned}
&\Pr[(i,k-1)|(i,k)] = 1, k \in [1, W_i - 1], i \in [1, m] \\
&\Pr[(0,k-1)_e|(0,k)_e] = 1 - q, k \in [1, W_i - 1], i \in [1, m] \\
&\Pr[(0,k-1)|(0,k)_e] = q, k \in [1, W_i - 1], i \in [1, m] \\
&\Pr[(0,k)_e|(i,0)] = \tfrac{(1-p)(1-q)}{W_0}, k \geq 0, i \in [0, m] \\
&\Pr[(0,k)|(i,0)] = \tfrac{(1-p)q}{W_0}, k \geq 0, i \in [0, m] \\
&\Pr[(\max\{(i+1,m\},k)|(i,0)] = \tfrac{p}{W_{\max\{i+1,m\}}}, k \geq 0, i \in [0, m] \\
&\Pr[(0,0)_e|(0,0)_e] = 1 - q + \tfrac{qP_{idle}(1-p)}{W_0} \\
&\Pr[(0,k)_e|(0,0)_e] = \tfrac{qP_{idle}(1-p)}{W_0}, k > 0 \\
&\Pr[(1,k)|(0,0)_e] = \tfrac{qP_{idle}p}{W_1}, k \geq 0 \\
&\Pr[(0,k)|(0,0)_e] = \tfrac{q(1-P_{idle})}{W_0}, k \geq 0
\end{aligned}
$$

Malone et al. [10] exploited the inclusion of this postback-off state and the fixed arrival probability constant to study the performance of the DCF while transferring packets from unsaturated heterogeneous traffic, e.g., file downloads, web traffic, etc. However, their analysis was limited in scope, since traffic types were parameterized only by p and q.

C. *Adding Variable Length Frame Payloads*

Despite the flexibility in the previous model, it is still limited with respect to packet size. In particular, the model assumes that each packet has an equally sized payload. This is not true, especially for video traffic [1], which typically consists of two types of packets of distinctly different sizes: d packets are small video frame/segment packets that carry information needed to render a piece of video data, and i packets are large "metadata" packets that contain information necessary to decode d packets. Multiple d packets are often tied to a single i packet in such a way that if the i packet is lost, none of its children d packets can be decoded correctly. Consequently, we need to be able to model packets of different sizes.

To do this, we consider a range of possible packet sizes, e.g., $[1, l]$, where l is the maximum packet payload size. In this context, the packet size corresponds to *how many* discrete time slots are needed to transfer the content over the channel. In the

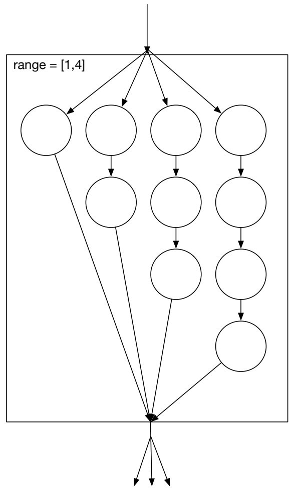

Fig. 3. A black box Markov chain that captures bounded variabilities in a particular parameter, such as packet length or packet interarrival time.

previously two discussed models, the packet size was assumed to be 1 since they were assumed to be transferred in a single time step. Since we may want to support both deterministic and arbitrary packet sizes, drawn from a specific distribution, we use what we call size chain black boxes to capture this type of variability. Figure 3 shows the internals of such a size chain black box.

To illustrate how this chain would be used, assume that the size chain black box represents the size of a particular packet. Further, assume that the packet size is a discrete random variable ranging from 1 to 4 with uniform distribution. The probability of transferring to any chain within the black box from the entry state is $(1/4)$. If the transition was to the last chain of length $l = 4$, then the state would appear to "loop" in place for 4 time steps before exiting the black box. Conversely, if the transition was to the first chain of length $l = 1$, then the state would only loop once before exiting. Although these states are costly from the perspective of space, this type of construction enables us to model such discrete random variables with any distribution within our Markov chain.

The first application of these size chain black boxes are to extend the previously discussed nonsaturated model to support variable packet sizes. Specifically, the number of time steps needed to transfer a packet and detect collision is now a bounded discrete random variable. This means that once a packet has begin transmitting it enters a size chain black box before either (a) detecting collision or (b) completing successfully. It is important to note that for packets of length $l > 1$, the probability of a collision is no longer q. Rather, a collision occurs if there is a collision in *any* time slot during which the packet was being transmitted. This means that the probability of collision for a packet of length l is $1-(1-p)^l$.

The new states required to model the size chain black boxes are tuples of the form (i, l, l'), where i is the backoff stage and l is the packet length, and l' is the *state* within the size chain of length l. For example, if the state transitions into a size chain of length $l = 4$ from state $(i, 0)$, then the following series of

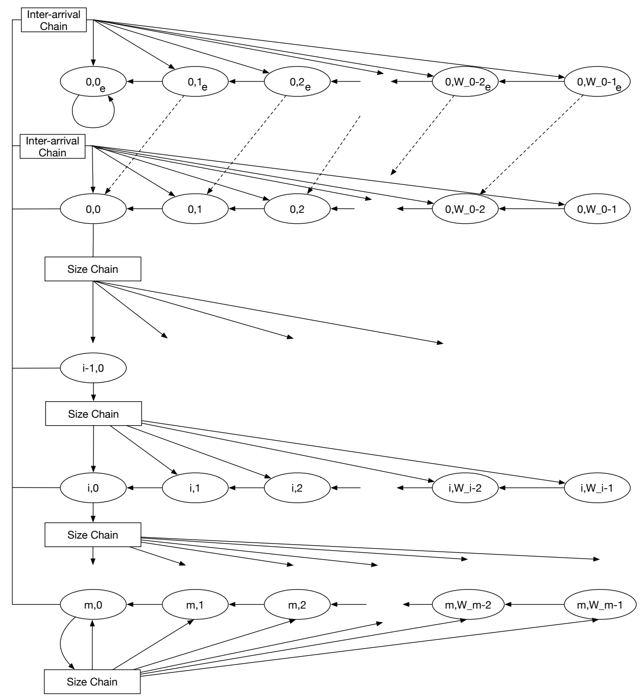

Fig. 4. The modified unsaturated DCF Markov model that captures variable-length packet payloads.

transitions would occur:

$$(i, 4, 4) \rightarrow (i, 4, 3) \rightarrow (i, 4, 2) \rightarrow (i, 4, 1)$$

To model this behavior, the following new state transition probabilities are included into the model:

$$\begin{array}{lll}
\Pr[(i, l, l'-1)|(i, l, l')] = 1 & l' \leq l & l' \geq 0 \\
\Pr[(i+1, k)|(i, l, 0)] = \frac{1-(1-p)^l}{W_{i+1}} & & k \in (0, W_{i+1}-1) \\
\Pr[(0, k)|(i, l, 0)] = \frac{(1-q)(1-p)^l}{W_0} & & k \in (0, W_0-1) \\
\Pr[(0, k)_e|(i, l, 0)] = \frac{q(1-p)^l}{W_0} & & k \in (0, W_0-1)
\end{array}$$

D. Adding Arbitrary Inter-Arrival Times

The final extension of our Markov model addresses the interarrival packets. The nonstaurted model proposed by Malone et al. [10] partly solves this problem by introducing a fixed probability q such that, at every time step, a new packet will arrive in the buffer to be transmitted. While useful, this does not allow us to capture more sophisticated interarrival times. For example, the interarrival time may be a random variable with a Zipf distribution. This mode would accurately capture a user quickly browsing though websites like Pinterest or Reddit.

To capture these dynamics, we re-use the size chain black boxes introduced in the previous section. In particular, after a successful transmission, the state of the system can enters an interarrival time black box before either (a) receiving the next available packet or (b) entering the postbackoff state because a packet has not yet arrived. This extension is shown in Figure 4. If need be, we can enforce that $q = 1.0$ so that the postbackoff states are replaced by the interarrival time black boxes.

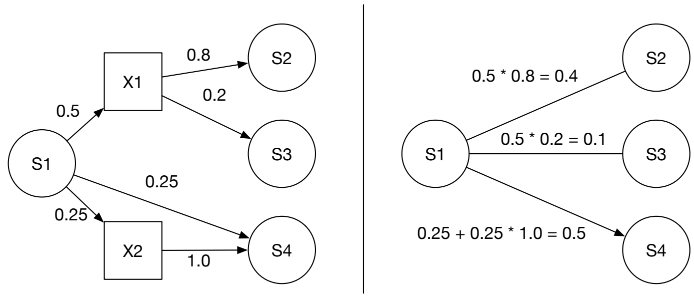

Fig. 5. A sample usage of "compressible" states that bridge the transition gap between separate Markov chains.

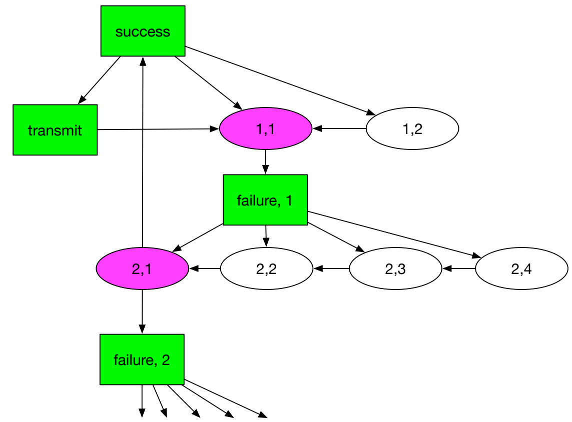

Fig. 6. Bianchi's DCF model represented using compressible states. Green states are collapsible.

IV. Simulator Design and Implementation

A major part of this research was developing a simulator that could manage the growing complexity of the multi-dimensional Markov state. To this end, in this section we describe the relevant design and implementation details that were used to realize the analytical models just described. These details will be of importance to those seeking to extend our tunable analytical models.

A. Managing Markov Model Dimensional Complexity

By adding more tunable parameters to Bianchi's 2-dimensional DCF, we needed a way to manage the growing complexity of the model. We accomplished in two ways: by (1) creating small Markov models for each "state" of the system, i.e., the packet size calculation, treating them as black boxes, and (2) using "compressible" states as logical and instantaneous bridges between Markov model black boxes. An example of a compressible state is shown in Figure 5. Observe that the $x1$ and $x2$ states are the "bridges" between the states $S1$ and $S2, S3, S4$. When "compressed," the transition probabilities between $S1$ and $S2$, for example, is the product of the probabilities across the compressible state.

To illustrate the efficacy of these states, consider the compressible variant of the DCF model shown in Figure 6. It is easy to see its equivalence to the original DCF model after the compressible (green) states are compressed. Our implementation uses these compressible states to bridge between separate Markov chains. In this case, we treat each backoff timer stage as a separate chain. Moving between these chains, either through transmission success or failure, happens through compressible states.

With this representation, adding support for the postback-off, interarrival, and packetsize Markov chains is simple. Consider the problem of modelling interarrival time. After a packet successfully transmits, it enters the transmit state, indicating that it is ready to transfer another packet. If there is a non-zero probability of an interarrival time greater than one (1), the transmit state will enter the inter-arrival state. Then, depending on the distribution of the interarrival times, the state will transition into the interarrival chain at the appropriate index. Notice that the index into this chain determines the interarrival time.

As an example, let the interarrival time be a discrete random variable with uniform distribution sampled from the range [1,3]. If the interarrival time is determined to be 3, which will happen with probability exactly 1/3, then the chain will enter the index at state 3. Since the probabilities between the interarrival chain states are deterministic with probability 1.0, this means that there *will always* be 3 epochs before the state transitions out of the interarrival chain and into the "ready-to-transmit" state.

To support different packet lengths, we follow the same approach and use packet size chains, which are similar to interarrival time chains, bef ore attempting to transmit a packet. We emphasize here that the implementation does not strictly adhere to the model previously described. The astute reader will have observed that the model should compute the size of a packet *once*, and then use that same packet size for every transmission attempt. The extension we presented in the previous section recomputes the packet size at every transmission attempt. We presented the model this way for clarity only.

In the actual implementation, a packet size range of length n is modeled by *duplicating* the extended DCF model n times, where each copy has $i = 1, \ldots, n$ has a packet size chain of length i at the "beginning" of the model. This is illustrated in Figure 7, where, after a packet arrives, the state of the system transitions into the copy with the appropriate packet size chain. For example, let the packet size be a discrete random variable sampled from [1,3] with a uniform distribution. With probability 1/3 the state of the chain will transition into the the DCF copy (black box) with the packet size of length 1, where it will remain until the packet is transmitted successfully.

B. Multi-Node Simulator Overview and Metrics Computations

Having described the architecture of the Markov chain, we now describe how we implement the multi-node simulator using individual instances of these chains. Let N be the number of nodes in a system, each with a unique set of traffic characteristics and, therefore, a unique Markov model, state space, and transition probability matrix. The simulator maintains a collection of individual simulator nodes, each one with its own Markov chain state space and probability transition matrix. At each time step, the simulator advances the

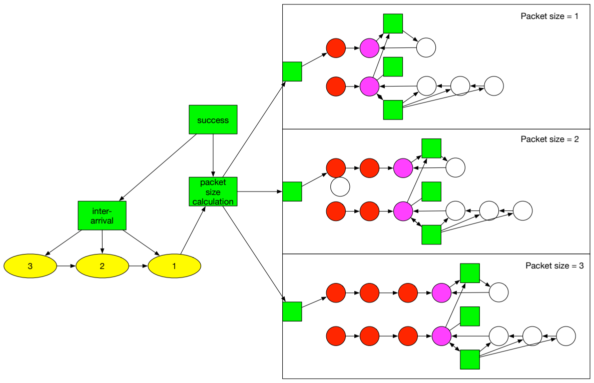

Fig. 7. The extended compressible DCF model with support for parameterized packet lengths.

state of each node according to the current system state and its individual state. For example, if at time t the simulator sees that nodes n_i and n_j are trying to transmit, then the simulator will advance each from its current state by setting $p = 1.0$. Ultimately, the state of the collection of nodes determines parameters such as p, the conditional collision probability.

The master simulator individual simulator nodes are also responsible for bookkeeping relevant events. Specifically, traces of state transitions are recorded so that the entire system dynamics can be replayed and relevant metrics can be computed about both individual nodes and the group as a whole. In our work, we mainly consider the following metrics: throughput, packet transmission success probability, and packet failure probability. Let n be the total number of steps in the simulation, s be the number of successful transmissions, and f be the number of failed transmissions. Throughput S is defined as the ratio of successful transmissions out of *all* possible time steps in the system, i.e., $S = p/n$. The success probability P_s is defined as the number of success transitions compared to the number of transmission attempt transitions, i.e., $P_s = p/(p+f)$. Finally, the packet failure probability P_f is defined as the number of failed transitions compared to the number of transmission attempts, i.e., $P_f = f/(p+f)$.

V. Related Work

Analytical models of the 802.11 DCF function began with Bianchi's seminal work in [4]. His saturated model was extended to handle non-saturated traffic by Malone et al. in [10] and Zhao et al. in [15] (the latter is a significantly simpler model than the former). In addition to a constant conditional collision probability p, the work of Malone et al. includes a per-epoch probability of a packet arrival in the buffer q. The authors used this probability to study the performance of non-saturated traffic parameterized by q. Nguyen et al. [11] combine nodes with different saturated and unsaturated characteristics, QoS requirements, and even different backoff timer windows (e.g., W_{max} and W_{min} values).

Qiu et al. [12] extended this even further to study the interference caused by an arbitrary number of nodes, and provide additional support for receiver models of packet-loss rates and unicast sender and receiver models. The latter additions are made possible by including model support for senders and receivers of different packets, e.g., data, ACKs, and NACKs. Our work is distinct from this in that we model traffic as a *distribution* of packet types, sizes, lengths, and interarrival times.

VI. Conclusion

In this work we presented a novel parameterized Markov model that can used to emulate arbitrary types of application traffic. This is particularly useful for studying the coexistence of heterogeneous traffic in DCF-based networks. Future work will include using this model to study the performance of DCF under random and deterministic backoff selection strategies with many sources of different application traffic.

References

[1] L. Badia, N. Baldo, M. Levorato, and M. Zorzi, "A markov framework for error control techniques based on selective retransmission in video transmission over wireless channels," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 3, pp. 488–500, 2010.

[2] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, March 2000.

[3] ——, "Ieee 802.11-saturation throughput analysis," *Communications Letters, IEEE*, vol. 2, no. 12, pp. 318–320, 1998.

[4] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2. IEEE, 1996, pp. 392–396.

[5] H. S. Chhaya and S. Gupta, "Performance modeling of asynchronous data transfer methods of ieee 802.11 mac protocol," *Wireless networks*, vol. 3, no. 3, pp. 217–234, 1997.

[6] I. C. S. L. M. S. Committee *et al.*, "Wireless lan medium access control (mac) and physical layer (phy) specifications," 1997.

[7] B. P. Crow, "Performance evaluation of the ieee 802.11 wireless local area network protocol," Ph.D. dissertation, UNIVERSITY OF ARIZONA, 1996.

[8] A. Kumar, D. Manjunath, and J. Kuri, *Communication networking: an analytical approach*. Elsevier, 2004.

[9] B. A. Mah, "An empirical model of http network traffic," in *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, vol. 2. IEEE, 1997, pp. 592–600.

[10] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 159–172, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/TNET.2006.890136

[11] S. H. Nguyen, H. L. Vu, and L. L. Andrew, "Performance analysis of ieee 802.11 wlans with saturated and unsaturated sources," *Vehicular Technology, IEEE Transactions on*, vol. 61, no. 1, pp. 333–345, 2012.

[12] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. ACM, 2007, pp. 171–182.

[13] M. Reisslein and G. Prasanth, Van der Auwera, "Traffic and quality characterization of single-layer video streams encoded with h.264mpeg-4 advanced video coding standard and scalable video coding extension," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 698–718, 2008.

[14] M. Reisslein, P. Seeling, and B. Kulapala, "Network performance evaluation with frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, no. 3, pp. 58–78, Third Quarter 2004.

[15] Q. Zhao, D. H. Tsang, and T. Sakurai, "A simple and approximate model for nonsaturated ieee 802.11 dcf," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 11, pp. 1539–1553, 2009.