

Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows

Raffaele Bruno, Marco Conti, and Enrico Gregori

Abstract—There is a vast literature on the throughput analysis of the IEEE 802.11 media access control (MAC) protocol. However, very little has been done on investigating the interplay between the collision avoidance mechanisms of the 802.11 MAC protocol and the dynamics of upper layer transport protocols. In this paper, we tackle this issue from an analytical, simulative, and experimental perspective. Specifically, we develop Markov chain models to compute the distribution of the number of active stations in an 802.11 wireless local area network (WLAN) when long-lived Transmission Control Protocol (TCP) connections compete with finite-load User Datagram Protocol (UDP) flows. By embedding these distributions in the MAC protocol modeling, we derive approximate but accurate expressions of the TCP and UDP throughput. We validate the model accuracy through performance tests carried out in a real WLAN for a wide range of configurations. Our analytical model and the supporting experimental outcomes show that 1) the total TCP throughput is basically independent of the number of open TCP connections and the aggregate TCP traffic can be equivalently modeled as two saturated flows and 2) in the saturated regime, n UDP flows obtain about n times the aggregate throughput achieved by the TCP flows, which is independent of the overall number of persistent TCP connections.

Index Terms—802.11 MAC protocol, TCP, UDP, performance modeling, Markov chain.

1 INTRODUCTION

THE IEEE 802.11 technology [1] is currently the most popular wireless local area network (WLAN) standard, and its commercial success continues to grow as new high-speed versions are produced (for example, the 802.11e [2] and 802.11n [3] standards), and new market opportunities are explored (for example, *metroscale* 802.11 networks [4]). However, the tremendous increase in the number of wireless users, the demand for high-speed high-bandwidth wireless communications, and the evolution toward more sophisticated and quality-sensitive applications require the design of mechanisms to improve the media access control (MAC)-layer performance. For instance, the maximization of the network capacity requires the development of accurate modeling tools in order to identify the optimal network operating points. In addition, the system behavior is affected at various extents by the dynamics of the transport layer protocols adopted to deliver users' traffic (for example, closed-loop protocols versus open-loop protocols and responsive flows versus unresponsive flows). For these reasons, it is crucial to develop rigorous analytical models so as to characterize the interactions between the 802.11 MAC protocol and the behavior of the upper layer transport protocols.

Due to the popularity of the IEEE 802.11 standard, there is a vast literature on the throughput analysis of its MAC

protocol. The seminal theoretical foundations on this research area have been obtained, focusing on 1) flat single-hop ad hoc networks and 2) saturated and unidirectional traffic [5], [6], [7], [8], [9]. Consequently, these models are not suitable for analyzing the performances of feedback-based flow-controlled transport protocols in a WLAN. In fact, the accurate modeling of the Transmission Control Protocol (TCP) dynamics requires the characterization of the correlation between transmissions in both directions, that is, how data and acknowledgment (ACK) packets interfere with each other, going from the access point (AP) to the wireless stations and vice versa. A few models (for example, [10], [11], and [12]) have relaxed the saturation assumption to develop the MAC throughput analysis. However, these studies adopt synthetic traffic patterns such as Poisson processes to model the arrival rate of packets at the MAC buffer, which cannot be applied to the TCP modeling. On the other hand, there is a vast literature specifically focusing on the TCP modeling, for example, [13], [14], [15], [16], [17], [18], and references therein. These analytical studies mainly concentrate on characterizing the TCP evolution either at the packet-based level or at the macroscopic level, evaluating the TCP throughput in the presence of congestion and loss events caused by bottleneck links and noisy channels. These models can help in understanding the impact of the network and TCP parameters on the TCP performance, but they are not useful in describing the interplay between TCP and the underlying network, in particular, the MAC protocol. In conclusion, *the interactions between the collision avoidance mechanisms employed by the 802.11 MAC protocol and the closed-loop behavior of TCP have not been sufficiently explored.* However, understanding these interactions is fundamental

• The authors are with the Institute for Informatics and Telematics (IIT) of the Italian National Research Council (CNR), Via G. Moruzzi, 1 - 56124 PISA, Italy.

E-mail: {raffaele.bruno, marco.conti, Enrico.gregori}@iit.cnr.it.

Manuscript received 29 Sept. 2006; revised 5 Mar. 2007; accepted 30 May 2007; published online 14 June 2007.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0255-0906. Digital Object Identifier no. 10.1109/TMC.2007.70718.

to identify and explain performance issues. In addition, the previously cited studies are focused on either TCP or User Datagram Protocol (UDP) in isolation. To the best of our knowledge, *a unified analysis of the interactions between TCP and UDP flows in 802.11 WLANs is still missing*, and this is the focus of our paper.

In this paper, we develop a comprehensive and rigorous analysis of the throughput performance of long-lived upstream and downstream TCP connections competing with finite-load UDP upstream flows in an 802.11 WLAN. One major contribution of our work is the *formulation of Markov chain models to compute the steady-state distributions of the number of active TCP and UDP stations in the network*. Our model takes into account both the TCP flow-control algorithms and the 802.11 backoff procedure. To cope with the problem complexity, in our analysis, we assume that packet losses are rare events and that the paths between the AP and the TCP end points do not introduce significant delays, which is a reasonable scenario in local environments. By embedding these distributions in the MAC protocol modeling, we derive approximate but accurate expressions of the TCP and UDP throughput. A second relevant contribution of our mathematical study is the *exploitation of aggregation techniques to define a simplified but equivalent model of the TCP dynamics, which permits us to accurately predict the TCP performance by substituting the real TCP stations with a limited number of saturated and unidirectional traffic sources*. This aggregation approach is particularly useful to simplify the modeling of the interactions between TCP and UDP traffic and to explain the differences between the channel resources allocated by the 802.11 MAC protocol to TCP and UDP flows.

In contrast with many prior studies that employ only simulations to verify analytical results, we validate our model accuracy by using both simulations and performance tests carried out in a real WLAN for a wide range of configurations. Although simulations provide a more flexible and manageable method to evaluate system performance, real experiments are important to substantiate the accuracy of the theoretical formulations. In addition, experimenting with real hardware and protocol implementations is useful to verify that the underlying modeling assumptions capture all the fundamental characteristics of the real system.

Our analytical results and the supporting experimental outcomes show the following:

1. When the flow control performed by the TCP receivers is the main factor limiting the sending rate of the corresponding TCP senders, TCP downstream flows and TCP upstream flows equally share the channel bandwidth.
2. Due to the interplay between the collision-avoidance mechanisms employed by the 802.11 MAC protocol and the TCP flow control, TCP stations are sporadically active, whereas the AP accumulates most of the traffic generated by the TCP connections.
3. Without UDP traffic, the total TCP throughput is basically independent of the number of open TCP connections, and the aggregate TCP traffic can be equivalently modeled as two saturated flows.

4. In the saturated regime n UDP flows obtain about n times the aggregate throughput achieved by the TCP flows, which is independent of the total number of persistent TCP connections.

We believe that this paper facilitates the understanding of the interdependencies between the 802.11 MAC protocol and upper layer transport protocols, which have not received sufficient consideration in prior studies. For this reason, our analysis may be useful to derive more accurate bounds of the MAC protocol capacity for the case of flow-controlled traffic. In addition, our model can provide the analytical basis to explore enhancements of the 802.11 standard, as well as to guide the development of efficient schedulers [19], more accurate admission control policies, effective traffic differentiation techniques, and so forth.

The rest of the paper is organized as follows: Section 2 outlines related work and positions our analysis with respect to prior studies. In Section 3, we present the system model and we discuss the modeling assumptions. Section 4 develops the throughput analysis. In Section 5, we show experimental results to validate the accuracy of our model. Finally, Section 6 concludes the paper with final remarks.

2 RELATED WORK

The literature on the IEEE 802.11 MAC protocol modeling is considerable, and it is out of the scope of this paper to present a complete survey. Rather, we refer to those papers that are most related to our work.

The basic foundations for the performance modeling of the 802.11 MAC protocol can be identified in three papers [5], [6], [8]. In [5], Bianchi developed a discrete-time Markov chain model to describe the evolution of the 802.11 backoff process, assuming a constant and independent collision probability. Calí et al. [6] elaborated a p -persistent variant of the standard 802.11 distributed coordination function (DCF) access method, assuming a constant and independent per-slot transmission probability. The persistence factor p was derived from the contention windows used in the 802.11 DCF such that the p -persistent IEEE 802.11 protocol closely approximates the standard protocol. In [8], Tay and Chua adopted a different modeling approach based on an average value analysis to derive approximate expressions of the collision probability. These seminal models have many commonalities (for example, the use of renewal theory arguments) and, in particular, they require that the system operates in *saturated conditions*; that is, each station has always a packet to transmit. Several other papers have built on these popular models, leading to many extensions. For instance, [20] extends Bianchi's model, integrating the finite retry-limit operations. Research developed in [21], [22], [23], and [24] account for additional modeling details and more precise definitions of the backoff process. The model developed in [25] is used to investigate multiple traffic classes, whereas [26] analyzes the quality-of-service (QoS) extensions defined in the novel 802.11e standard [2]. In [7], the energy consumption efficiency of the 802.11 MAC protocol has been analyzed. Finally, a recent paper [9] simplifies and generalizes Bianchi's analysis, providing a novel fixed-point formalization.

The aforementioned models are very accurate if the wireless stations are saturated. However, the saturation assumption is unlikely to be valid in many configurations since most of the Internet traffic is bursty in nature and often operates in an on-off manner. For these reasons, renewed research efforts have been devoted to develop analytical tools that relax the restrictions on saturated conditions. A few papers have extended Bianchi's analysis, introducing new states, called *postbackoff* states, in the Markov chain describing the backoff process in order to represent a node that has an empty output queue after successfully transmitting a packet [11], [12]. To address the modeling difficulties, Duffy et al. [11] assume that each station can buffer only one packet, whereas, in [12], the MAC buffers are modeled using an $M/G/1$ queue. In [10], a model for low-rate sources is proposed because it is assumed that each station has at most a single packet when starting a new backoff. An alternative approach is proposed in [27], in which each node is modeled as a discrete-time $G/G/1$ (that is, the arrival process is arbitrary) queue and the average backoff window of the saturated system is used as an approximation of the one in the unsaturated case.

Our review of previous papers has pointed out that the performance analysis of 802.11 wireless networks under finite-load sources is generally based on the use of synthetic traffic patterns (for example, Poisson processes) to model the packet arrival rate at the MAC buffers. Although this methodology has been commonly used in the performance analysis of computer networks, it does not appear adequate to characterize the behavior of transport-layer protocols in 802.11-based wireless networks. For these reasons, papers concentrating on the analysis of TCP and MAC interactions are exploring different approaches. To the best of our knowledge, one of the first contributions in this research area is [28], where we formulated a Markov model to calculate the statistical distribution of the number of active TCP stations in the case of multiple downstream flows. However, the analysis developed in [28] restricts the maximum TCP receive window size to a single packet. Successively, we extended our mathematical study to consider both downstream and upstream flows [29], as well as arbitrary TCP receive window sizes [30]. Note that the idea of using a Markov chain to model the number of TCP packets stored in the stations' queues has also been adopted in two recent papers [31] and [32]. However, in [31], the state-transition probabilities are empirically computed from simulations instead of mathematical analysis, whereas Yu and Choi [32] propose a multidimensional Markov chain to compute the distribution of the number of stations with a particular number of TCP packets in their queue, which leads to the state space explosion for large maximum TCP receive window sizes. Before concluding this review of related work, it is useful to notice that alternative approaches have been proposed to model the interactions of TCP with the IEEE 802.11 MAC protocol. A simpler analysis is conducted in [33], where the statistics of the number of stations with nonempty buffers are obtained by assuming that they follow a Bernoulli distribution law. In that paper, a queuing model is also proposed to compute the mean session delay in the presence of short-lived TCP

flows. A mathematical study is also presented in [9] and [34] to show how the saturated network analysis can be used to estimate TCP transfer throughput. However, the scenarios where this analysis can be applied are very limited and do not include common situations such as multiple TCP flows downloading files via an AP. Related to our work, the analysis developed in [15] is also interesting, which investigates TCP unfairness conditions caused by the use of a small-capacity buffer at the AP.

Concerning the study of the interdependencies between the 802.11 MAC protocol and UDP flows, very little can be found. Some experimental and simulation investigations [15], [35], [36] have been conducted to characterize the behavior of the UDP transport protocol in isolation, mainly in terms of throughput and average delay. In a prior work [37], we attempted to derive a preliminary unified model of TCP and UDP dynamics. However, in that paper, we considered only saturated UDP sources, and again, we restricted the analysis to a maximum TCP receive window size equal to one segment.

3 SYSTEM MODEL

In this section, we present the network model and the underlying modeling assumptions of our analysis.

3.1 Network and Station Model

In this study, we consider a typical WLAN, in which an AP provides access to external networks and Internet-based services (for example, Web transfers, retrieval of multimedia contents, and access to shared data repositories) to N wireless stations (in the following, they are also referred to as nodes). In this work, we are not concerned with mobility issues, and we assume that nodes are static. Concerning the traffic patterns, we assume that N^U wireless stations generate a connectionless UDP stream toward a server located in the high-speed wired LAN that the AP is connected to. As the streams are not acknowledged, the only traffic generated at the MAC level is composed of UDP data packets. The remaining N^T wireless stations are communicating with the server behind the AP by using TCP-controlled connections. Specifically, we consider a configuration in which there are N_d^T TCP downstream flows (that is, N_d^T stations are the receivers of TCP downlink connections) and N_u^T TCP upstream flows (that is, N_u^T stations are the senders of TCP uplink connections). In this case, both TCP data segments and TCP ACK packets travel over the wireless channel. The AP forwards either TCP ACK packets to the N_u^T senders or TCP data segments to the N_d^T receivers. For the sake of brevity, hereafter, we denote with S^U a station generating UDP datagrams, with S_u^T a station behaving as the sender for TCP upstream flows, and with S_d^T a station behaving as the receiver for TCP downstream flows. Fig. 1 illustrates the network layout and the traffic configuration considered in this paper. In the figure, the arrows indicate the direction along which the data packets travel over the network. For the TCP flows, in the reverse direction, there is also the feedback traffic composed of TCP ACK packets.

In this paper, we do not consider UDP downstream flows from the AP to the wireless stations because it is

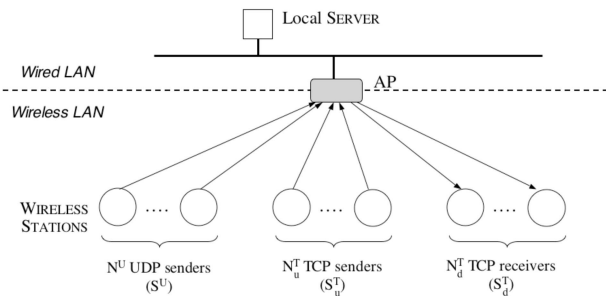


Fig. 1. The network model: network and traffic configurations.

widely recognized that drastic unfairness occurs when TCP flows share the same buffer with unresponsive UDP flows, competing for scarce bandwidth. In fact, the TCP flows reduce their sending rates in response to congestion, whereas the uncooperative UDP flows consume all the available bandwidth [38], [39]. Previous research has addressed this issue by either proposing per-flow scheduling mechanisms that separately regulate the bandwidth between responsive and unresponsive flows [40], [41], [42] or queue management techniques to identify and isolate unresponsive flows [43], which could be easily applied to an AP handling both UDP and TCP transfers. Thus, the focus of our work is the modeling of the interactions between TCP and UDP flows not when they share a common buffer but when they share a common wireless channel regulated by a random access protocol.

Before starting with the analysis, it is important to point out that the modeling problem is further complicated by the subtle interactions that may occur between the different queue management policies employed at various layers of the protocol stack. Specifically, the access control policy implemented in the MAC layer determines the order that the interface queues (that is, the transmission buffers) are served, as well as the related service time. Each frame will have a different service time depending on the number of nonempty queues in the network. The status of these transmission buffers is affected by several factors, including the packet arrival patterns, the dynamics of upper layer transport protocols, and the 802.11 backoff process. In addition to the transmission queue associated to the network interface, a pair of send and receive buffers is allocated to the transport layer to decouple the application program from the network data flow.¹

3.2 Modeling Assumptions

In our analysis, we adopt the following assumptions:

- A.1 We model the TCP dynamics in the absence of packet losses. To this end, we assume that the transmission queue of each node is large enough to avoid buffer overflows and that the TCP time-outs are sufficiently long to avoid time-out expirations due to round-trip time (RTT) fluctuations. In addition, since we consider a local sever, we assume negligible RTTs.

1. The reader is referred to [44] for a detailed description of a typical node architecture.

- A.2 We assume an *ideal wireless channel* that does not corrupt frame transmissions.

- A.3 We assume that the wireless channel is the bottleneck link of the system.

- A.4 We consider only long-lived TCP connections having an infinite amount of data to send. This means that our analysis is concerned with large file transfers.

- A.5 We assume that, for each TCP flow in the steady state, the TCP advertised window is smaller than the TCP congestion window.

- A.6 We assume that the application program at the receiver reads data from the socket receive buffer at the rate that it is received from the network. Thus, the TCP ACK packets always advertise the maximal TCP receive window size.

- A.7 We assume that the delayed ACK mechanism is disabled. This implies that each TCP data segment is separately acknowledged with an immediate ACK.

The fundamental assumptions that we have adopted to cope with the problem complexity are A.1 and A.2. In fact, based on those assumptions, we can assert that, after an initial transient phase, during which the TCP congestion window of each flow grows to its maximum value, each TCP connection reaches a stationary regime and operates according to the rules of the TCP flow-control algorithm. In these conditions, the sending rate of each TCP sender is limited only by the flow control performed by the corresponding TCP receiver. More precisely, owing to assumptions A.5 and A.6, it holds that the number of outstanding TCP packets (including TCP data packets and TCP ACKs) for a TCP flow is fixed and equal to the maximum TCP receive window size. This observation plays a crucial role in elaborating our analysis, as explained in the following. Finally, from assumption A.3, it follows that we may elaborate our analysis, as the local TCP server (shown in Fig. 1) is located on the AP. Note that, in our model, we do not consider TCP congestion control. However, this is not seen as a limitation because previous work [13] and the outcomes of our experiments show that packet losses may be negligible in high-speed WLANs with a limited channel interference. In addition, many of our modeling assumptions are commonly adopted in previous papers [30], [32], [33] on analyzing the TCP dynamics in 802.11 WLANs.

4 THROUGHPUT ANALYSIS OF UDP AND TCP FLOWS

Our analysis is based on a Markov renewal-reward approach [45]. Following the footprints of [6], *our key approximation is to assume a p -persistent variant of the 802.11 MAC protocol* in which the frame transmission probability in a randomly chosen slot time is independent of the number of retransmissions experienced by the frame. However, in contrast to [6], this *per-slot transmission probability p is not constant, but it varies, depending on the number of active stations*, that is, stations with nonempty transmissions queues. In the subsequent sections, we introduce the renewal cycles adopted during the analysis, and we develop the analytical tools needed to derive the stochastic

relation between the per-slot transmission probability p and the number of concurrent UDP and TCP flows. Another fundamental observation relates to the methodology that we used in the analysis. Whereas many previous papers have modeled the MAC buffers as independent queues (either $M/G/1$ or $G/G/1$ queues), in our analysis, we use *aggregate stochastic processes* whenever it is possible. As explained in the following, we elaborate our model to describe the evolution of the total number of TCP and UDP data segments and TCP ACK packets in the network. This information is sufficient to estimate the probability distribution of the number of active stations and to discern between active TCP and UDP transmitters and active TCP receivers. Note that we apply aggregation techniques in our theoretical analysis also to ensure *model scalability* and *limited numerical complexity*. In fact, when analyzing the TCP performance in WLANs with a large number of TCP connections having several TCP packets to transmit (that is, large TCP windows), the number of states needed to describe the TCP dynamics can easily raise unbridled. However, finding the steady-state solution of unrestricted large-scale Markov models can be a challenge. On the contrary, our modeling formulation carefully takes into account these scalability concerns so as to develop manageable analytical tools. To further cope with the model complexities and to ease the development of the theoretical analysis, we also employ *decomposition techniques*. More precisely, we decompose the initial modeling problem into two separate simpler problems. First, we obtain the stationary probability $b_{i,j}$ that i TCP data packets and j TCP ACKs are stored in the stations' output queues without considering UDP streams (a formal definition of the $b_{i,j}$ probability is given by (2) in Section 4.1). Then, we exploit this analysis to express the throughput performance of TCP-controlled data transfers and to define an *equivalent saturated model of the TCP traffic*. The second part of our mathematical study addresses the joint modeling of TCP and UDP by exploiting the equivalent abstraction of TCP dynamics derived in the first part.

4.1 Analysis of TCP-Controlled File Transfers

In this section, we briefly sketch the analysis in the case $N^U = 0$. In particular, we introduce the main results that we presented in [30], which are needed to develop the analysis in the case of concurrent TCP and UDP flows. However, due to space constraints, the formal proofs and the detailed mathematical derivations are not reported here, and the reader is referred to our extended technical report [44].

We start our analysis from the consideration that, under the assumptions listed in Section 3.2, it holds that the state of each TCP connection (that is, the number of outstanding TCP packets) can change only when there is a successful transmission at the MAC layer. In other words, the state of the nodes' output queues evolves only after a successful frame transmission. Owing to this observation, we may adopt the completion of successful transmissions as the embedding points of our model so as to ignore the backoff periods in the description of the TCP connection state. More precisely, let t and $t+1$ denote the epochs corresponding to the completion of two consecutive successful frame transmissions. Considering the i th TCP connection

($i = 1, \dots, N^T$), we indicate with $X_{tcp}^i(t)$ the stochastic process representing the number of TCP data segments stored in the TCP sender's output queue at time t and with $X_{ack}^i(t)$ the stochastic process representing the number of TCP ACK packets stored in the TCP receiver's output queue at time t . In the stationary regime, it holds that

$$\forall t, X_{tcp}^i(t) + X_{ack}^i(t) = W, \quad i = 1, \dots, N^T, \quad (1)$$

where W is the maximum TCP receive window size (in terms of fixed-size packets). Equation (1) formalizes the fact that the flow-control mechanism implemented in the TCP receivers is the dominant factor that regulates the TCP sending rate and limits the total number of outstanding TCP packets.

Complex interactions regulate the joint evolution of the $X_{tcp}^i(t)$ and $X_{ack}^i(t)$ processes. However, as shown in [44], instead of developing a per-connection model, we may describe the system evolution by using the aggregate stochastic processes $X_{tcp}(t) = \sum_{i=1}^{N^T} X_{tcp}^i(t)$ and $X_{ack}(t) = \sum_{i=1}^{N^T} X_{ack}^i(t)$. In addition, the bidimensional process $\{X_{tcp}(t), X_{ack}(t)\}$ can be modeled using the equivalent bidimensional process $\{Y_{tcp}(t), Y_{ack}(t)\}$, where $Y_{tcp}(t)$ expresses the total number of TCP data segments that are stored in the transmission queues of all the S_u^T stations at time t , and $Y_{ack}(t)$ expresses the total number of TCP ACKs stored in the transmission queues of all the S_d^T stations at time t . In other words, to compute the distributions of the number of TCP data segments and TCP ACK packets stored in each node's output queues (that is, wireless stations and AP), it is sufficient to compute these probability functions only for the output queues of the S_u^T and S_d^T stations.

Our aim is to compute the probability that, in the steady-state conditions, there are i TCP data segments in the S_u^T stations' transmission queues and j TCP ACK packets in the S_d^T stations' transmission queues. This probability can be formally defined as follows:

$$b_{i,j} = \lim_{t \rightarrow \infty} P\{Y_{tcp}(t) = i, Y_{ack}(t) = j\} \quad i \in (0, m_u), j \in (0, m_d), \quad (2)$$

where $m_u = N_u^T \cdot W$ and $m_d = N_d^T \cdot W$. For the sake of brevity, in the following, we adopt the short notation $Y_{i,j}$ to denote the system state such that $\{Y_{tcp}(t) = i, Y_{ack}(t) = j\}$. To cope with the model complexity, our *key approximation is to assume that*, for each transmission attempt and regardless of the number of retransmissions experienced by the frame, *the per-slot access probability is a fixed value p_k* , depending only on the number k of active TCP stations (that is, S_u^T and S_d^T stations with at least one packet to transmit). Later in this section, we describe how we can express the number k of active TCP stations when the network is in state $Y_{i,j}$ (for simplicity, we omit the subscripts (i, j) in the k parameter, unless ambiguity occurs). To compute the sequence of the p_k values, for $k \in (1, N^T + 1)$, we use the iterative algorithm first proposed in [6] to estimate the per-slot transmission probability p in a network with k saturated nodes under the assumption of infinite retry limits. Note that, in the real IEEE 802.11 MAC protocol, the transmission probability

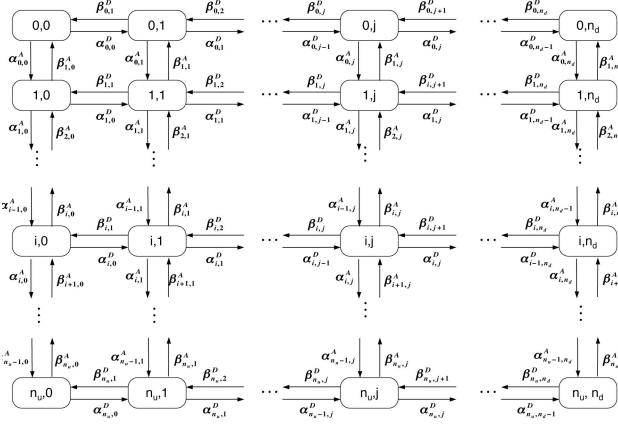


Fig. 2. General Markov model of the $\{Y_{tcp}(t), Y_{ack}(t)\}$ process.

depends on the history of channel accesses; however, we show through experimentations and simulations that our approximation still provides accurate estimates of the system behavior. Once the p -persistence is assumed for the MAC protocol operations, it follows that the successful transmissions on the channel are independent events. Therefore, it is possible to model the bidimensional process $\{Y_{tcp}(t), Y_{ack}(t)\}$ with the discrete time Markov chain depicted in Fig. 2. Note that, in this Markov chain, the state transitions occur only at the end of a successful transmission. Furthermore, owing to assumption A.7, the state transitions occur only toward adjacent states. In other words, the number of TCP data packets and TCP ACK packets stored in the stations' output queues can increase or decrease by only one packet after a successful frame transmission. More precisely, as shown in [30], the not-null one-step transition probabilities of the Markov chain depicted in Fig. 2 can be computed as follows:²

$$\begin{cases} Pr\{i+1, j|i, j\} = \alpha_{i,j}^A & i \in (0, m_u - 1), j \in (0, m_d) \\ Pr\{i-1, j|i, j\} = \beta_{i,j}^D & i \in (1, m_u), j \in (0, m_d) \\ Pr\{i, j+1|i, j\} = \alpha_{i,j}^D & i \in (0, m_u), j \in (0, m_d - 1) \\ Pr\{i, j-1|i, j\} = \beta_{i,j}^A & i \in (0, m_u), j \in (1, m_d), \end{cases} \quad (3)$$

where $\alpha_{i,j}^D$ ($\alpha_{i,j}^A$) is the probability that the AP successfully delivers a TCP data segment (TCP ACK packet) stored in its transmission queue and $\beta_{i,j}^D$ ($\beta_{i,j}^A$) is the probability that an S_u^T (S_d^T) station successfully transmits a TCP data segment (TCP ACK packet) stored in its transmission queue. To express these four probabilities, we must know how the TCP packets are distributed over the nodes' output queues in order to estimate the number of active S_u^T and S_d^T stations when the system is in state $Y_{i,j}$. To this end, we develop a *worst case* analysis that maximizes the network activity. Specifically, we assume a *deterministic distribution* of the TCP data segments and the TCP ACK packets on the transmission buffers so as to maximize the number of active TCP stations. For clarity, we indicate with n_i^u the number of active S_u^T stations when $Y_{tcp}(t) = i$ and with n_j^d the number of active S_d^T stations when $Y_{ack}(t) = j$. Owing to the previous assumptions, it follows that $n_i^u = \min(i, N_u^T)$ and

$n_j^d = \min(j, N_d^T)$. Furthermore, let $n_{i,j}^a$ be an indicator function that has value 1 when the AP has a nonempty transmission queue; otherwise, it is 0. It is intuitive to note that $n_{i,j}^a = 0$ only in the system state corresponding to $(i = m_u, j = m_d)$ because all the TCP data packets and TCP ACK packets are stored in the active stations' queues. Thus, the total number k of active nodes (that is, the *network activity*) when the network is in state $Y_{i,j}$ is $k = n_{i,j}^a + n_i^u + n_j^d$. Finally, exploiting standard probabilistic arguments described in [44], we can compute the transition probabilities (3) and solve the Markov chain.

4.1.1 Throughput Analysis

Let ρ be the total throughput measured at the transport layer, defined as the average number of TCP payload bits successfully transmitted per unit time. To compute the ρ value, we analyze the channel events between two consecutive successful frame transmissions (that is, our model embedding points). Hereafter, we refer to this time interval as *virtual transmission time* T_v . Furthermore, let us indicate with $T_v(i, j)$ a virtual transmission time that begins with the system in state $Y_{i,j}$. By exploiting the p -persistent assumption and renewal theoretical arguments, it is intuitive to note that the MAC protocol behavior regenerates at the epochs corresponding to the beginning of $T_v(0, 0)$ periods. In fact, in the state $Y_{0,0}$, all the wireless stations have empty transmission queues, and the AP is the only active node in the network. From the properties of the regenerative processes, it follows that the time interval between two consecutive regeneration epochs is a renewal time for the stochastic process describing the sequence of successful transmissions on the channel. Let $E[T_{renewal}]$ be the average duration of this renewal time. From the renewal-reward theory, we obtain that the ρ value is given by $E[P_{renewal}]/E[T_{renewal}]$, where $E[P_{renewal}]$ is the average number of TCP payload bits successfully transmitted during the average renewal period. The following reformulates the ρ expression so as to make its mathematical derivation easier:

Lemma 1. *By assuming the independence of virtual times and considering a p -persistent MAC protocol, it holds that*

$$\rho = \frac{\sum_{i=0}^{m_u} \sum_{j=0}^{m_d} b_{i,j} \cdot E[P_v(i, j)]}{\sum_{i=0}^{m_u} \sum_{j=0}^{m_d} b_{i,j} \cdot E[T_v(i, j)]} = \frac{E[P_v]}{E[T_v]}, \quad (4)$$

where $E[P_v(i, j)]$ is the average number of TCP payload bits transmitted during the $E[T_v(i, j)]$ period.

Proof. The proof is reported in [44]. \square

By inspecting (4), we observe that the throughput analysis can be divided in two distinct phases. First, we need the $b_{i,j}$ distribution. Then, we should compute the average duration of a virtual time, given that the network is in state $Y_{i,j}$, that is, $E[T_v(i, j)]$. The analytical derivation of the $E[T_v(i, j)]$ expression is based on well-known results originally formalized in [6] for a p -persistent model. In fact, in our model, the probability that a station transmits in a randomly chosen slot is assumed constant and approximated with the equivalent per-slot transmission probability, say, p_k , of a saturated network with $k = n_{i,j}^a + n_i^u + n_j^d$ active

2. We adopt the short notation $Pr\{Y_{tcp}(t+1) = i_1, Y_{ack}(t+1) = j_1 | Y_{tcp}(t) = i_0, Y_{ack}(t) = j_0\} = Pr\{i_1, j_1 | i_0, j_0\}$, as is done in [5].

nodes [6]. The k value changes only after a successful transmission and in accordance with the number of TCP data segments and TCP ACK packets stored in the stations' output queues. Thus, the p_k probability is constant during the entire $T_v(i, j)$ period. Due to space constraints, the complete derivation of the $E[T_v(i, j)]$ is omitted here and reported in [44].

Finally, it is important to discern between the throughput obtained by either the TCP upstream flows or the TCP downstream flows. To this end, the ρ expression can be alternatively written as

$$\rho = \rho^U + \rho^D, \quad (5)$$

where ρ^U is the average number of TCP payload bits successfully transmitted per unit time by the S^U stations, and ρ^D is the average number of TCP payload bits successfully received per unit time by the S^D stations. Considering Lemma 1, it is intuitive to recognize that $\rho^U = E[P_v^U]/E[T_v]$ ($\rho^D = E[P_v^D]/E[T_v]$), where $E[P_v^U]$ ($E[P_v^D]$) is the contribution to $E[P_v]$ due to the TCP payload bits transmitted by the S^U (S^D) stations during the average virtual time $E[T_v]$. Closed-form expressions for ρ^U and ρ^D are reported in [44].

4.1.2 Considerations on the Average Number of Active Stations

In this section, we elaborate on the characteristics of the probability distribution of the number of active TCP stations. Specifically, we show numerical results demonstrating that the average number of active TCP stations (excluding the AP) is bounded to less than two stations. In [28], we theoretically proved that two stations is an asymptotic bound for the average number of active TCP stations, considering only TCP downstream flows and $W = 1$. Here, we provide empirical evidence that this result can also be extended to the case of arbitrary TCP receive window sizes. In addition, our analysis clearly highlights that the *average network activity is an invariant figure for the system, and it does not depend on the number of open TCP connections*. This is a counterintuitive result, confirming that the TCP dynamics in an 802.11 WLAN are radically different from what is expected in a saturated network. For this reason, this finding is extremely important both to better understand the measured TCP throughput and to develop the analysis in the case of concurrent UDP and TCP connections.

Let $E[K]$ and $E[\hat{K}]$ be the average numbers of active nodes in the network after a successful transmission, including or not including the AP, respectively (closed-form expressions for $E[K]$ and $E[\hat{K}]$ are reported in [44]). Table 1 compares the measured $E[K]$ and $E[\hat{K}]$ values with the model descriptions for several network configurations. Note that we use discrete-event simulations³ to validate this part of the analysis because it was not possible to read the buffer status in our experimental setup. However, the validation of the throughput analysis is conducted in Section 5.1 through real experiments. Due to the space

TABLE 1
Study of the Average Network Activity

W	N_u^T	N_d^T	$E[\hat{K}]$		$E[K]$	
			analysis	simulation	analysis	simulation
1	1	1	1.00	0.92889	1.75	1.69146
1	1	2	1.30	1.17601	2.20	2.09607
1	1	5	1.49693	1.31688	2.4954	2.31605
1	1	10	1.50	1.31875	2.50	2.31875
32	1	1	1.25385	1.15385	2.25385	2.15385
32	1	2	1.39081	1.20785	2.39081	2.20785
32	1	5	1.53156	1.25558	2.52835	2.25558
32	1	10	1.50877	1.24792	2.50877	2.24792
1	2	1	1.30	1.17535	2.20	2.09562
1	5	1	1.49693	1.31627	2.4954	2.31545
1	10	1	1.50	1.31937	2.50	2.31937
32	2	1	1.39081	1.20723	2.39081	2.20723
32	5	1	1.48578	1.25685	2.48578	2.25685
32	10	1	1.49599	1.27096	2.49599	2.27096
1	2	2	1.4375	1.27527	2.40625	2.25424
1	5	5	1.50	1.31923	2.50	2.31923
1	10	10	1.50	1.31858	2.50	2.31858
32	2	2	1.45096	1.22676	2.45096	2.22676
32	5	5	1.49992	1.25388	2.49992	2.25388
32	10	10	1.50	1.24762	2.50	2.24762

restrictions, we show results related to a few representative network configurations for a small TCP receive window size ($W = 1$ maximum segment size (MSS)) and a large TCP receive window size ($W = 32$ MSS), but similar types of behavior have been observed in all the other cases. From the analytical outcomes, we can note that the proposed model slightly overestimates the average number of active stations. This is an expected result, since we developed a worst case analysis of the network activity. However, this approximate description of the network activity is sufficiently accurate to precisely evaluate the TCP throughput, as shown in Section 5.1.

To better understand the results reported in Table 1, we also investigated the probability distribution of the AP's buffer occupancy. From our results (a comprehensive study can be found in [44]), we found out that the AP's buffer accumulates almost all the frames generated by the TCP connections and only a few frames are stored in the wireless stations' buffers on the average. Since most of the TCP packets (both data and ACKs) are accumulated at the AP, the downlink transmissions constitute the network *bottleneck link*. The reason is that the AP contends with N^T uplink carrier sense multiple access (CSMA) instances. Thus, the frames are drained from the AP's buffer at a lower pace than from the wireless stations' queues and the AP's buffer rapidly fills up. However, the wireless stations have new frames to transmit only when the AP delivers new TCP data segments or TCP ACK packets. Thus, *the closed-loop control implemented by TCP implicitly limits the network activity*.

Before concluding this section, it is useful to discuss the implications of these results on the system modeling. Specifically, our analysis shows that the average number of active TCP stations is basically constant and independent of the number of open TCP connections, direction of TCP transfers, maximum TCP receive window size, and so forth. This finding suggests that the TCP throughput may be

3. The simulation environment is a customized tool [7], [30], implementing the TCP Reno version [46]. The parameter setting is compliant with the IEEE 802.11b PHY layer at 11 megabits per second (Mbps) [47].

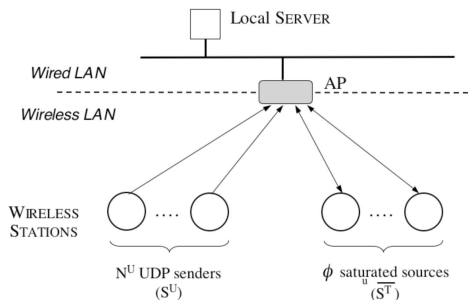


Fig. 3. The *equivalent* network model: network and traffic configurations.

computed by using approximate (and simpler) modeling approaches capable of capturing the basic features of the TCP dynamics without incorporating all the complex details of the TCP flow-control algorithms.

4.2 Analysis of Concurrent UDP and TCP Connections

In this section, we extend the analysis developed in Section 4.1 by including S^U stations, that is, UDP upstream flows. Thus, let us assume that there are N^U UDP-controlled uploads, N_u^T TCP-controlled uploads, and N_d^T TCP-controlled downloads. In our analysis, we consider persistent TCP connections, whereas the UDP streams generate a finite load. A formal definition of the arrival process of the UDP sources is provided later in this section.

To tackle this modeling problem, we may follow the same approach adopted in Section 4.1.1, that is, to derive the probability distribution functions of the TCP data segments, TCP ACK packets, and TCP UDP datagrams stored in the network. However, this approach may lead to an uncontrollable model complexity and to an explosion of the state-space size of the corresponding Markov chain. On the contrary, under the assumption that packet losses are absent and the delays in the network are negligible, *TCP connections exhibit an easily predictable behavior that may be explained by introducing an equivalent and simplified abstraction of the TCP dynamics*. Specifically, two main observations can be derived from our analytical study. First, the AP is most of the time saturated, and the ratio between the TCP data packets and the TCP ACKs stored into its buffer is approximately N_d^T/N_u^T . Second, the N^T TCP stations are sporadically active because the interactions between the CSMA with collision avoidance (CSMA/CA) protocol and the TCP closed-loop nature bound the number of active station to a few nodes on the average. Our claim is that, when modeling the relationship between the TCP and UDP dynamics, we can neglect the complex feedback-based flow-control algorithms employed to regulate the bidirectional TCP transfers. This implies that the *TCP stations can be emulated with a small number of saturated unidirectional traffic sources*. Fig. 3 shows how the original network model depicted in Fig. 1 changes. More precisely, in this equivalent network, the *AP is saturated* and generates packets with variable payload size (say, P^a). The cumulative distribution function of the P^a random variable (that is, the probability that the payload size is less than or equal to l bytes) can be expressed as

$$F^a(l) = P\{P^a \leq l\} = \begin{cases} 0 & l < 0 \\ N_u^T/N^T & 0 \leq l < l_T \\ 1 & l \geq l_T. \end{cases} \quad (6)$$

Here, l_T is the fixed-size payload of TCP data packets (in terms of bytes). This distribution follows from the observation that the real AP generates TCP data segments for the N_d^T TCP downlink flows and TCP ACKs for the N_u^T TCP uplink flows. In this study, we assume that the TCP data segments have a fixed-size data content equal to l_T bytes and that the TCP ACK packets have no useful payload. Concerning the second approximation, we assume that ϕ saturated wireless stations, say, S^T (with $\phi < N^T$), model the overall TCP traffic generated by the real TCP stations. Considering that these equivalent S^T stations generate TCP ACK packets for the N_d^T TCP downlink flows and TCP data segments for the N_u^T TCP uplink flows, the cumulative distribution function of the packet-payload size (say, $\overline{P^T}$) of a generic S^T 's transmission can be expressed as

$$F^t(l) = P\{\overline{P^T} \leq l\} = \begin{cases} 0 & l < 0 \\ N_d^T/N^T & 0 \leq l < l_T \\ 1 & l \geq l_T. \end{cases} \quad (7)$$

Note that introducing equivalent saturated sources implicitly hides the TCP flow-control dynamics. This greatly simplifies the modeling problem because, now, we have to compute only the distribution of the UDP traffic in the network, that is, the probability that h UDP datagrams are stored in the S^U stations' buffers at a given time instant. It is worth pointing out that our modeling approach is different from the ones used in previous studies, in which *all* the TCP flows are assumed saturated [9]. On the contrary, in our work, we define an equivalent saturated traffic model that is used to emulate the TCP traffic and its impact on unresponsive traffic such as UDP. Later in this section, we explain how we can compute the ϕ value such that this equivalent system closely approximates the throughput performance of the original network.

Let $Y_{udp}(t)$ be the total number of UDP datagrams that are stored in the S^U stations' transmission queues at time t . In this part of our analysis, we adopt the same embedding points used in [5]: t and $t+1$ correspond to the end of two consecutive backoff slots.⁴ Our aim is to compute the probability that, in the steady-state conditions, there are h UDP datagrams in the S^U stations' transmission queues at the end of a virtual slot (that is, the observation point of our discrete-time model). This probability can be formally defined as follows:

$$b_h = \lim_{t \rightarrow \infty} Pr\{Y_{udp}(t) = h\} \quad \text{with } h \in (0, h^u), \quad (8)$$

where $h^u = N^U \cdot W^U$ with W^U denoting the memory space allocated by the operating system to the UDP send buffer (in terms of fixed-size datagrams). Since, in our study, we assume finite socket buffers, UDP datagram losses due to buffer overflows at the sender side can occur, depending on the UDP load and the current network contention. For the sake of brevity, in the following, we adopt the

4. The time interval between two consecutive backoff slots, called *virtual slot time*, may be much longer than the slot time size as it may include a packet transmission which can be a success or a collision.

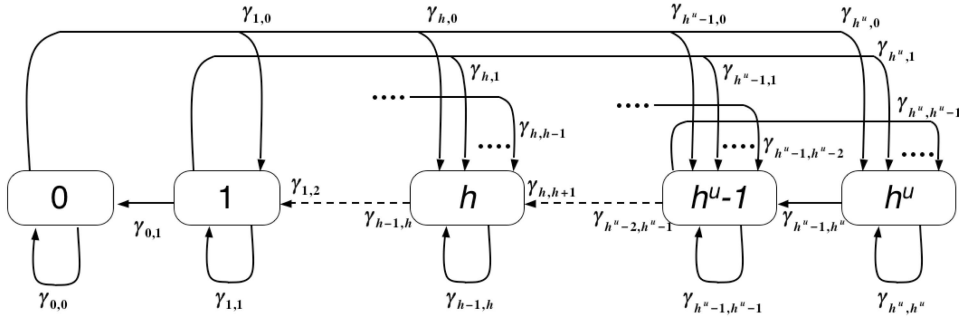


Fig. 4. General Markov model of the $\{Y_{udp}(t)\}$ process.

short notation Y_h to denote the system state such that $\{Y_{udp}(t) = h\}$. In addition, we denote with $E[R]_h$ (also $E[R(h)]$) the expectation of the random variable R conditioning the network to be in state Y_h . Note that R is a general notation referring to any random variable.

To cope with the model complexity, we adopt the same approximation used in Section 4.1.1; that is, for each transmission attempt and regardless of the number of retransmissions suffered, the station transmission probability is a fixed value p_k , where k is the number of active stations given that the network is in state Y_h . For brevity, we omit the subscript h in the parameter k unless ambiguity occurs. To compute the number of backlogged stations in state Y_h , we assume that the UDP datagrams stored in the S^U stations' transmission buffers are *deterministically distributed* over the S^U stations' output queues so as to maximize the number of active S^U stations. Formally, let n_h^u denote the number of active S^U stations when the network is in state Y_h . From the assumptions on the distribution of UDP datagrams over the S^U stations' buffers, it follows that $n_h^u = \min(h, N^U)$. Hence, the total number of active nodes in state Y_h is given by the n_h^u active UDP stations, the saturated AP, and the ϕ equivalent saturated TCP stations. Then, it holds that

$$k = n_h^u + 1 + \phi \quad \text{with} \quad h \in (0, h^u). \quad (9)$$

In order to derive the stationary probability distribution b_h defined in (8), our key approximation is to assume that the MAC protocol behavior regenerates at the epochs corresponding to the completion of two consecutive virtual slots. It is well consolidated that, once the p -persistence is assumed for the MAC protocol operations and the stations operate in saturation conditions, all the processes that define the occupancy pattern of the channel (that is, idle slots, collisions, and successful transmissions) are regenerative with respect to the sequence of time instants corresponding to the end of virtual slot times [5], [7]. However, in the considered network scenario, the regenerative properties are not valid as the number of UDP stations may change during the virtual time. In fact, the real packet arrival process is a continuous time process and the arrival of a new UDP packet may occur at any time in the slot. For simplicity, we assume that the queues' state changes only at the end of the slot so that we may consider p_k constant during the slot time. Nevertheless, the discretization of the arrival process is an approximation that

introduces negligible errors and still provides accurate results, as shown in Section 5.2.

Owing to previous observations, it is possible to model the stochastic process $\{Y_{udp}(t)\}$ with the discrete-time Markov chain depicted in Fig. 4. Note that this is a finite Markov chain because, in the S^U stations', output queues cannot be stored in more than h^u UDP packets. Let γ_{i_1, i_0} be a short notation to denote the transition probabilities $Pr\{Y_{udp}(t+1) = i_1 | Y_{udp}(t) = i_0\}$ of this Markov chain. To compute these transition probabilities, it is useful to introduce some auxiliary probabilities. First, we have to formally define how we can compute the UDP offered load. Our key assumption is that an arbitrary function $\Theta(i, T)$ can be defined to express the probability that i new UDP datagrams arrive in the S^U stations' transmission queues in a generic time interval T . We keep the derivation as general as possible, and then we derive the particular cases relevant for the experimental validation of the analytical outcomes. To derive a closed formula for the γ_{i_1, i_0} transition probability, let us separately consider the different types of virtual slots conditioning to the state Y_h . In particular, a virtual slot can be any of the following:

- **An idle slot.** With probability $P\{Idle\}_h = (1 - p_k)^k$, the virtual slot is empty.⁵ During an idle slot, i new UDP datagrams arrive in the S^U stations' transmission queues with probability $\Theta(i, SLOTT)$, where $SLOTT$ denotes the slot time duration.
- **A successful transmission.** With probability $P\{Succ\}_h = kp_k(1 - p_k)^{k-1}$, the virtual slot contains a successful transmission. In addition, we can further distinguish between three types of successful transmissions: 1) an AP's successful transmission occurring with probability $P\{Succ^A|Succ\}_h$, whose average duration is $E[T_S^A]_h$, 2) an S^U station's successful transmission occurring with probability $P\{Succ^U|Succ\}_h$, whose average duration is $E[T_S^U]_h$, and 3) an $\overline{S^T}$ station's successful transmission occurring with probability $P\{Succ^T|Succ\}_h$, whose average duration is $E[T_S^T]_h$. During a success, i new UDP datagrams arrive in the S^U stations' transmission queues with a probability $\Theta(i, E[T_S^X]_h)$, where X indicates the type of this successful transmission.
- **A collision.** With probability $P\{Coll\}_h = 1 - P\{Idle\}_h - P\{Succ\}_h$, the virtual slot contains a

5. The k value is computed by means of (9).

collision, whose average duration is $E[T_C]_h$. During a collision, i new UDP datagrams arrive in the S^U stations' transmission queues with probability $\Theta(i, E[T_C]_h)$.

From standard probabilistic arguments, it follows that

$$P\{Succ^A|Succ\}_h = \frac{1}{n_h^u + 1 + \phi}, \quad (10a)$$

$$P\{Succ^U|Succ\}_h = \frac{n_h^u}{n_h^u + 1 + \phi}, \quad (10b)$$

$$P\{Succ^T|Succ\}_h = \frac{\phi}{n_h^u + 1 + \phi}. \quad (10c)$$

The derivation of the closed-form expressions of the $E[T_S^X]_h$ and $E[T_C]_h$ quantities involves the packet-size distributions defined in (6) and (7) and requires cumbersome probabilistic machinery. Hence, they are reported in our technical report [44].

By means of the set of probabilities defined above, we are now able to express the not-null one-step transition probabilities of the Markov chain depicted in Fig. 4 as follows:

$$\left\{ \begin{array}{l} \gamma_{h-1,h} = P\{Succ\}_h \cdot P\{Succ^U|Succ\}_h \cdot \Theta(0, E[T_S^U]_h) \\ \quad h \in (1, h_u), \\ \gamma_{h+\delta h,h} = P\{Succ\}_h \cdot P\{Succ^U|Succ\}_h \cdot \Theta(\delta h + 1, E[T_S^U]_h) \\ \quad + P\{Succ\}_h \cdot P\{Succ^A|Succ\}_h \cdot \Theta(\delta h, E[T_S^A]_h) \\ \quad + P\{Succ\}_h \cdot P\{Succ^T|Succ\}_h \cdot \Theta(\delta h, E[T_S^T]_h) \\ \quad + P\{Coll\}_h \cdot \Theta(\delta h, E[T_C]_h) \\ \quad + P\{Idle\}_h \cdot \Theta(\delta h, SLOTT) \\ \quad h \in (0, h_u), \delta h \in (0, h_u - h - 1), \\ \gamma_{h_u,h} = P\{Succ\}_h \cdot P\{Succ^U|Succ\}_h \sum_{\delta h=\Delta h_u}^{\infty} \Theta(\delta h + 1, E[T_S^U]_h) \\ \quad + P\{Succ\}_h \cdot P\{Succ^A|Succ\}_h \sum_{\delta h=\Delta h_u}^{\infty} \Theta(\delta h, E[T_S^A]_h) \\ \quad + P\{Succ\}_h \cdot P\{Succ^T|Succ\}_h \sum_{\delta h=\Delta h_u}^{\infty} \Theta(\delta h, E[T_S^T]_h) \\ \quad + P\{Coll\}_h \sum_{\delta h=\Delta h_u}^{\infty} \Theta(\delta h, E[T_C]_h) \\ \quad + P\{Idle\}_h \sum_{\delta h=\Delta h_u}^{\infty} \Theta(\delta h, SLOTT) \\ \quad h \in (0, h_u), \Delta h_u = h_u - h. \end{array} \right. \quad (11)$$

The first equation in (11) expresses the probability that the number of UDP datagrams stored in the S^U stations' transmission buffers decreases by 1. This event occurs when there is a successful transmission carried out by an S^U station, and no new UDP datagram arrives during the $E[T_S^U]_h$ interval. The second equation in (11) expresses the probability that the number of UDP datagrams stored in the S^U stations' transmission buffers rises from h to $h + \delta h$, with $\delta h \in (0, h_u - h - 1)$. To derive this transition probability, we have considered the different types of possible virtual slots and computed the probability that, during each type of virtual slot, the number of UDP datagrams stored in the S^U stations' transmission buffers grows from h to $h + \delta h$. Finally, the third equation models the transition probability from a generic state Y_h to the state Y_{h_u} . Note that, when the $\{Y_{udp}(t)\}$ process reaches the state h_u , all the

S^U stations' buffers are full. Thus, a new UDP datagram arriving in this state cannot be admitted in the network. Consequently, any $\delta h \geq h_u - h$ produces the transition from the state Y_h to the state Y_{h_u} . This explains the summations used in the $\gamma_{h_u,h}$ formula.

By means of (11), we can write the flow-balance equation at the generic state Y_h of the Markov chain depicted in Fig. 4. Then, using the normalization condition, that is, $1 = \sum_{h=0}^{h_u} b_h$, and solving the resulting linear system of equations, we finally calculated the stationary chain distribution for each ϕ value in $(1, N^T)$. By embedding the b_h knowledge in the model of the MAC protocol operations, we can now develop the throughput analysis.

4.2.1 Throughput Analysis

Let ρ^* be the overall system throughput measured at the transport layer, defined as the average number of payload bits successfully transmitted per unit time. Once p -persistence is assumed for the MAC protocol operations and regenerative properties are assigned to the stochastic processes describing the channel occupations, following the line of reasoning used in Section 4.1.1 to prove Lemma 1, we can prove that

$$\rho^* = \frac{E[P_{vs}]}{E[T_{vs}]}, \quad (12)$$

where $E[T_{vs}]$ is the average duration of a virtual slot and $E[P_{vs}]$ is the average number of information payload bits successfully transmitted during the average virtual slot $E[T_{vs}]$. By taking the conditional expectation of $E[P_{vs}]$ and $E[T_{vs}]$, given that the network is in state Y_h , $E[P_{vs}]$ can be computed as $\sum_{h=0}^{h_u} b_h \cdot E[P_S]_h \cdot Pr\{Succ\}_h$, where $E[P_S]_h$ is the average payload size of a successful transmission, whereas $E[T_{vs}]$ can be computed as $\sum_{h=0}^{h_u} b_h \cdot E[T_{vs}]_h$. The mathematical derivation of these quantities is obtained following probabilistic arguments similar to the one used in [5], and it is detailed in [44].

Finally, it is important to discern between the throughput obtained by the UDP and TCP connections and between upstream flows and downstream flows. To this end, the ρ^* expression can be alternatively written as

$$\rho^* = \rho_{udp}^* + \rho_{tcpU}^* + \rho_{tcpD}^*, \quad (13)$$

where ρ_{udp}^* is the average number of UDP payload bits successfully transmitted per unit time by the S^U stations, ρ_{tcpD}^* is the average number of TCP payload bits successfully received per unit time by the $\overline{S^T}$ stations, and ρ_{tcpU}^* is the average number of TCP payload bits successfully transmitted per unit time by the $\overline{S^T}$ stations. The closed-form expressions for ρ_{udp}^* , ρ_{tcpD}^* , and ρ_{tcpU}^* are reported in [44].

Note that the ρ^* value is implicitly a function of the ϕ parameter. In principle, the ϕ value should describe the contribution of the wireless stations acting as TCP receivers and TCP senders to the average network activity. Thus, it should be $\phi = E[\bar{K}]$. However, the mathematical framework that we have developed requires that the ϕ parameter is a discrete value. Several techniques can be considered to map a real value on a set of discrete values. One possibility is to assign to the ϕ parameter the greatest

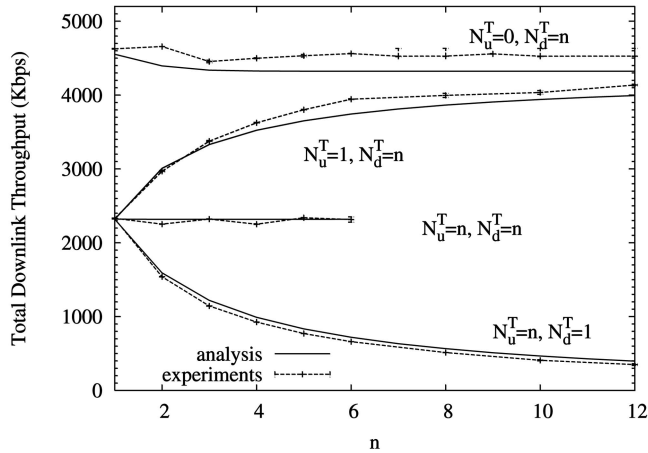


Fig. 5. Comparison of model predictions and measured ρ^D .

integer number less than or equal to the exact value (that is, $\phi = \text{floor}\{E[\hat{K}]\}$). Obviously, more sophisticated solutions can be devised, such as assuming that the ϕ parameter follows some specific distribution law whose mean is $E[\hat{K}]$. However, our experimental results justify the use of a simpler approach.

5 EXPERIMENTAL VALIDATION OF ANALYTICAL RESULTS

To both verify the accuracy of our throughput analysis and the correctness of the underlying modeling assumptions, we conducted a series of performance tests on a real 802.11b network consisting of eight IBM R50 ThinkPad laptops equipped with an Intel Pro-Wireless 2200 wireless card and six Acer Aspire 5633WLMi laptops equipped with an Intel Pro-Wireless 3945 wireless card. Note that we did not use a commercial base station but a computer instrumented as an AP to have full access to all the implementation details of the protocol stack, which are typically not made public by manufacturers. All nodes, including the AP, use a Linux 2.6.18 kernel. In all the experiments, the wireless interfaces are configured to select the IEEE 802.11b mode at 11 Mbps, with disabled data rate auto fallback and no request to send (RTS) or clear to send (CTS). In our experimental setup, all the laptop PCs were configured to use the basic TCP Reno protocol with a disabled Delayed ACK mechanism.⁶ Finally, in accordance with our analytical study, we set up the default TCP receive buffer equal to W MSS and the default UDP sender buffer equal to W^U datagrams.

It is evident that several factors can impact the experimental results and some of them are impossible to isolate or control. For instance, the wireless link interference affecting the area where the testbed is located can have a relevant effect on the measured throughput performances. Therefore, in order to limit the occurrence of external and uncontrollable radio interference, we carried out the performance tests in an area of the Italian National Research

6. Note that, on Linux machines, protocol parameters are easily configured by exploiting the `/proc` virtual file system [48].

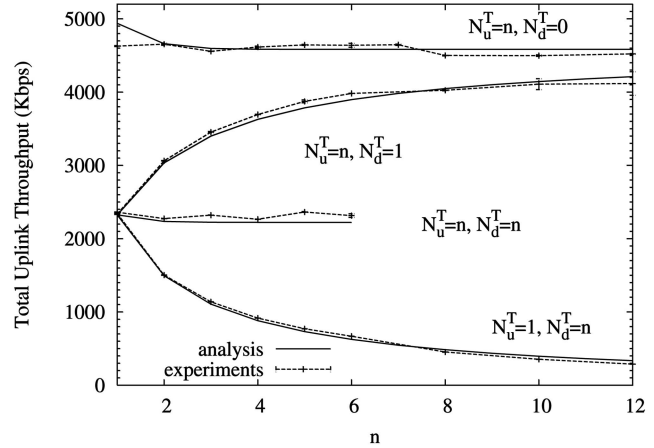


Fig. 6. Comparison of model predictions and measured ρ^U .

Council (CNR) building not covered by other wireless networks. To measure the steady-state performance, we have replicated each test five times. The following graphs report both the averages and the 95 percent confidence intervals. Note that the confidence intervals are often very tight (≤ 1 percent) and are not appreciable from the plots.

5.1 Performance Evaluation for $N^U = 0$

If not otherwise specified, the packet size is constant in all the experiments and the transport layer payload is equal to 1,448 bytes. The header at the Internet Protocol (IP) layer is 20 bytes long, whereas the header at the TCP layer is 32 bytes long because the `iPerf` application introduces in the TCP basic header the optional time stamp field. In the following, we show results obtained using $W = 16$ MSS. We have also carried out tests investigating the impact of the size of the TCP receiver's advertised window, and we obtained similar results that are not reported here for space limitations.

The curves plotted in Figs. 5 and 6 compare the TCP throughput measured during the experiments with the model predictions for different mixes of TCP downstream and upstream flows. More precisely, we consider five representative network layouts:

1. n TCP downlink streams ($N_u^T = 0, N_d^T = n$),
2. n TCP uplink streams ($N_u^T = n, N_d^T = 0$),
3. one TCP downlink stream competing with n TCP uplink streams ($N_u^T = n, N_d^T = 1$),
4. one TCP uplink stream competing with n TCP downlink streams ($N_u^T = 1, N_d^T = n$), and
5. the same number of TCP downlink and TCP uplink streams ($N_u^T = n, N_d^T = n$).

Fig. 5 shows the total throughput obtained by the TCP downlink connections (that is, the ρ^D value, as defined in (5)), whereas Fig. 6 shows the aggregate throughput of the TCP uplink connections (that is, the ρ^U value, as defined in (5)). From the graphs, we can note that our model provides a good correspondence between the analysis and the measured throughput in all the considered settings. The shown results also indicate that the TCP downstream flows and the TCP upstream flows equally share the channel

bandwidth. In addition, the total TCP throughput is basically independent of the number of open TCP connections. The reason is that the interactions between the MAC protocol and the TCP flow-control algorithm limit the average network contention to a few stations. Consequently, the impact of collisions on the throughput performance is negligible and independent of the number of wireless stations in the WLAN. This is in contrast with the typical behavior of a saturated network in which the collision probability increases by increasing the number of wireless stations. These findings further reinforce our claims that the accurate modeling of the TCP dynamics in 802.11-based WLANs cannot be developed using the basic saturation analysis.

5.1.1 Considerations on TCP Fairness

In the following discussion, we adopt the concept of TCP fairness introduced in [15]. Specifically, in a WLAN, the TCP fairness is measured as the ratio r between the rates of downlink and uplink TCP flows, that is, $r = \rho_D / \rho_U$.

It is evident that it is extremely important to ensure that the access to the network by different users remains fair. For this reason, the TCP fairness characteristics in 802.11 WLANs have been recently studied by several researchers identifying a variety of different issues [15], [49], [50], [51], [52], [53], [54]. However, most of the performance problems revealed by these studies are rooted in the location-dependent channel diversities that may characterize the wireless environment [50]. In [51], Heusse et al. observe that slow stations may considerably limit the throughput of other stations roughly to the level of the lowest rate. In [53], it is shown that there is unfairness between long-lived and short-lived TCP flows because the short-lived TCP flows are more susceptible to losses and channel unavailability during the early stages of the TCP window growth. Unfairness between the uplink and downlink TCP flows in 802.11 WLANs is observed in [15], [52], and [54], but different causes are identified. More precisely, those papers investigate the negative impact of using a small-capacity buffer at the AP on TCP fairness. Pilosof et al. [15] propose a solution that is based on the TCP receive window size manipulation at the AP, whereas Bottigliengo et al. [52] and Leith et al. [54] propose solutions based on the contention window adaptation at the MAC layer.

From the results plotted in Figs. 5 and 6, we observe that the average TCP ratio r during our performance tests is in the range [0.98, 1.08]. Hence, in the tested conditions, the MAC protocol is basically fair with the TCP connections. We can offer several arguments explaining why TCP unfairness does not appear in our experiments. First, we perform our tests in a wireless environment characterized by stable and homogeneous channel conditions. Thus, we do not observe asymmetry in the packet transmission rates of the different wireless stations. In addition, the packet losses are rare events. Hence, the congestion window growth is not hindered, and the TCP time-outs are infrequent. Finally, the sizes of interface and driver queues are not artificially set to small values, but we use the system default values. For instance, in the experiments conducted in [54], the network buffers are configured such that the

buffer space is 10 maximum transition unit (MTU)-sized packets. Similarly, simulations carried out in [15] show unfairness only for buffer sizes lower than 100 MTU-sized packets. In our devices, the system default values for the interface buffer size was 1,000 MTU-sized packets, which is sufficient to avoid buffer overflows. In conclusion, both our analytical and experimental results show that, when the flow control performed by the TCP receivers is the main factor limiting the sending rate of corresponding TCP senders, the TCP flows equally share the available channel bandwidth.

5.2 Performance Evaluation for $N^U > 0$

Concerning the TCP traffic, we use the same setting adopted to obtain the results shown in Section 5.1, which is a fixed-size payload equal to 1,448 bytes. On the other hand, the following results have been obtained using *Constant Bit Rate* (CBR) UDP sources. For this type of traffic generator, adjusting the fixed packet interarrival time is sufficient for controlling the offered load.⁷ The UDP sources are configured in such a way to generate fixed-size IP packets with the same length as the IP packets generated by the TCP sources, that is, 1,500 byte long IP packets. However, the UDP header is only 8 bytes, which is much smaller than the TCP header. Therefore, the UDP packet payload size is equal to 1,472 bytes.

As explained in Section 4.2.1, our analytical framework requires that the UDP offered load is modeled through the $\Theta(i, T)$ function, which expresses the probability that i new UDP datagrams arrive in the S^U stations' transmission queues in a generic time interval T . To derive the $\Theta(i, T)$ function for CBR sources, we can note that, given a per-station offered load of OL packets per second, the average total number ψ of UDP packets generated in the time interval T is simply $\psi = N^U \cdot OL \cdot T$. However, we have to map this real value on a set of discrete values, associating to each discrete value an appropriate probability. The discretization method that we employed to transform the continuous $\Theta(i, T)$ variable into a small set of integer values relies on the use of a discrete bimodal distribution. More precisely, we construct a bimodal distribution with the mean equal to ψ : One peak is equal to the largest integer less than or equal to ψ (that is, $\lfloor \psi \rfloor$), and the other peak is equal to the smallest integer not less than ψ (that is, $\lceil \psi \rceil$). Formally, this distribution can be defined as follows:

$$\Theta(i, T) = \begin{cases} p_0 = \frac{\lceil \psi \rceil - \psi}{\lceil \psi \rceil - \lfloor \psi \rfloor} & i = \lfloor \psi \rfloor \\ 1 - p_0 & i = \lceil \psi \rceil \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

In the following, we show results obtained using $W = 16$ MSS and $W^U = 50$ MTU-sized packets.

The first set of experiments was conducted to evaluate the impact of UDP flows on TCP upstream flows. To this end, Fig. 7 compares the UDP throughput with the TCP throughput as a function of the UDP offered load. We report results related to a network with up to six TCP flows and four UDP sources. First, from the shown results,

7. In [44], we replicated the same set of experiments, considering also exponentially distributed interarrival times, that is, *Poisson* UDP sources, obtaining similar results.

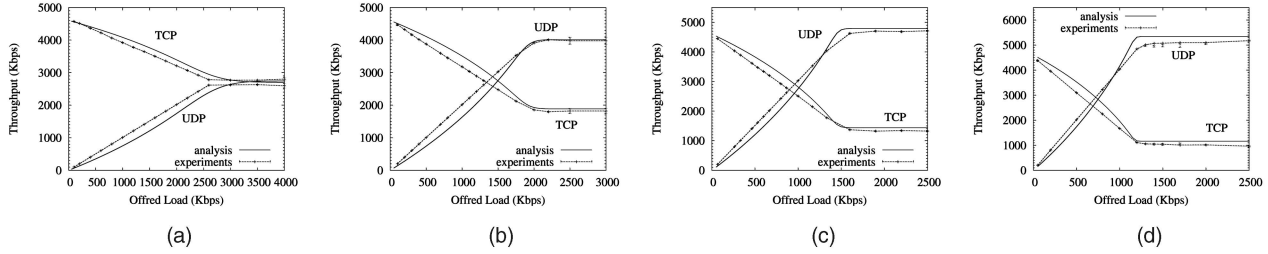


Fig. 7. Impact of UDP flows on TCP upstream flows. (a) $N^U = 1$, $N_u^T = 4$, and $N_d^T = 0$. (b) $N^U = 2$, $N_u^T = 4$, and $N_d^T = 0$. (c) $N^U = 3$, $N_u^T = 4$, and $N_d^T = 0$. (d) $N^U = 4$, $N_u^T = 6$, and $N_d^T = 0$.

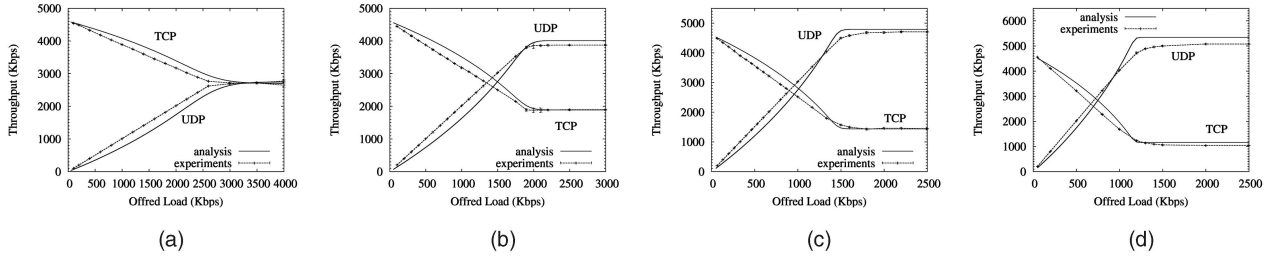


Fig. 8. Impact of UDP flows on TCP downstream flows. (a) $N^U = 1$, $N_u^T = 0$, and $N_d^T = 4$. (b) $N^U = 2$, $N_u^T = 0$, and $N_d^T = 4$. (c) $N^U = 3$, $N_u^T = 0$, and $N_d^T = 4$. (d) $N^U = 4$, $N_u^T = 0$, and $N_d^T = 6$.

we observe that there is a fair match between our model predictions and the experimental measurements for both the TCP and UDP flows. Furthermore, by inspecting the plotted curves, we can identify two distinguishable regions. The first region corresponds to an *unsaturated regime*, in which the UDP throughput is linearly proportional to the offered load. In this case, the TCP throughput decreases in accordance to the amount of channel bandwidth occupied by the UDP traffic. The second region corresponds to a *saturated regime*, in which both the TCP and UDP throughputs stabilize and are independent of the UDP offered load. In addition, in the saturated regime, the N^U UDP connections obtain about N^U times the aggregate throughput achieved by the TCP connections. Thus, the UDP flows are in the network more and the system unfairness is more severe.

From the plotted curves, we notice that, in the unsaturated regime, our analysis slightly underestimates the UDP throughput and overestimates the TCP throughput. This observation can be explained by noting that our model approximates the per-slot access probability when there are n active UDP stations with the equivalent per-slot access probability of a network with $n + 1 + \phi$ saturated nodes. However, the 802.11 specification recommends that a station should not invoke the backoff procedure when a new packet arrives in an empty buffer, and the channel is sensed idle for at least a DCF interframe space (DIFS) period. This implies that, in certain conditions, the unsaturated UDP sources may transmit immediately after an idle DIFS period without using additional backoff delays. Consequently, real unsaturated UDP sources may have a higher per-slot access probability than is assumed in our model, thus obtaining higher throughput.

To explain the results related to the saturated regime, it is useful to remember that, in a saturated network, the standard 802.11 MAC protocol ensures a long-term *per-station throughput fairness* [9]. In other words, in a saturated network, all the stations have roughly the same probability of accessing the channel and performing a successful

transmission. Thus, to better understand the stationary regimes depicted in Fig. 7, we should estimate how many active stations are in the network on the average. It is intuitive to note that, when the UDP offered load exceeds the network capacity, the S^U stations' transmission queues are never empty. Hence, in the saturated network state, we have N^U asymptotic S^U stations. On the other hand, we have demonstrated that persistent TCP connections are emulated by a saturated AP and ϕ saturated TCP stations ($\phi < 2$, as shown in Table 1). For instance, let us consider the case shown in Fig. 7a. In this configuration, we have a single UDP flow and four stations operating as TCP senders, whereas the AP is the TCP receiver for the TCP upstream flows. However, during the saturated regime, the four TCP senders behave as a single saturated source of TCP data packets. This explains why the UDP throughput and the aggregate TCP throughput are basically the same. Similar reasoning can be used to describe the unfairness between TCP and UDP in the other cases shown in Fig. 7.

A second set of experiments has been conducted to evaluate the impact of UDP flows on TCP downstream flows. The comparison of the experimental results with the analytical results is reported in Fig. 8. The plotted curves further validate our model accuracy, confirming that the analysis is capable of characterizing quite precisely the TCP and UDP throughput. In addition, the shown results clearly indicate that the direction of TCP streams does not affect the throughput performance. In fact, by comparing the results shown in Figs. 7 and 8, we can note that the total throughputs obtained by either TCP downstream flows or TCP upstream flows are very similar.

The last set of experiments considers a more complex scenario, with both TCP downstream and TCP upstream flows. Due to space limitations, we show only the results related to a network with two TCP downlink connections, two TCP uplink connections, and a varying number of UDP flows. Fig. 9a compares the aggregate TCP uplink

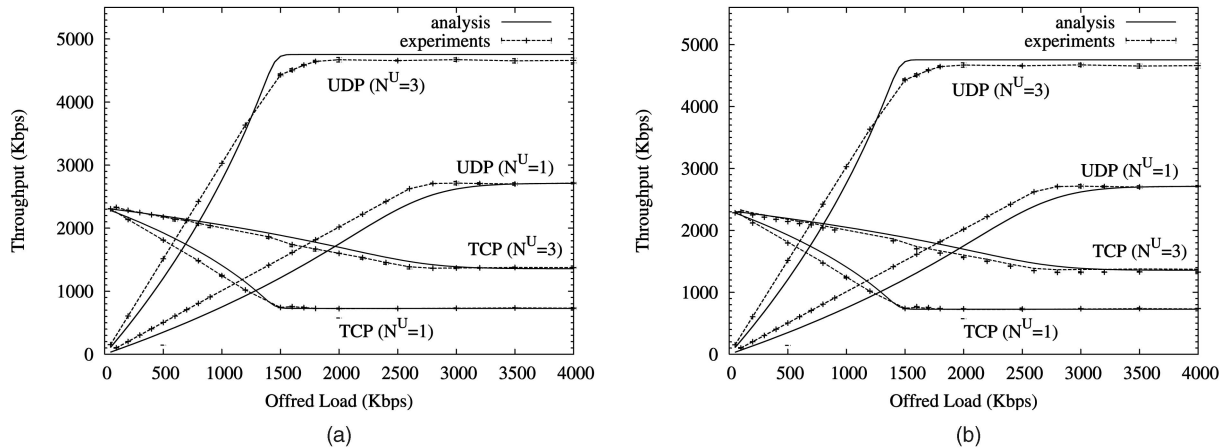


Fig. 9. TCP and UDP throughput for $N_u^T = 2$, $N_d^T = 2$, and various UDP sources. (a) TCP uplink throughput. (b) TCP downlink throughput.

throughput with the UDP throughput, whereas Fig. 9b compares the aggregate TCP downlink throughput with the UDP throughput. From the shown results, we can observe that 1) TCP downstream flows and TCP upstream flows obtain equal throughput, as in the case of the TCP traffic in isolation, and 2) in the single UDP-flow case, the total UDP throughput is almost equal to the aggregate TCP throughput, whereas, in the three UDP-flow cases, the total UDP throughput is roughly three times higher than the aggregate TCP throughput. This last observation can again be explained by noting that there is a critical offered load that saturates the UDP sources, whereas the TCP connections autoregulate their sending rates and only a small number (less than two) of TCP stations are active on the average. This causes a severe unfairness between the UDP and TCP flows.

6 CONCLUSIONS

This paper developed a general analytical framework to derive accurate closed-form expressions for the throughput of finite-load UDP flows and persistent TCP connections in an 802.11 WLAN, under the assumption of no packet losses and negligible delays in the system. We proved that, in an 802.11 WLAN, the TCP stations are sporadically active, whereas the AP stores most of the traffic generated by the TCP connections. Starting from these observations, we formulated an equivalent abstraction of the TCP dynamics that permits us to model an arbitrary number of TCP streams by using a small set of unidirectional and saturated traffic sources. Then, this saturated equivalent model was employed to characterize the interactions between the MAC-layer and the upper layer transport protocols and between flow-controlled TCP flows and unresponsive UDP sources. Our mathematical study was also used to provide a clearer insight into the system behavior and a better understanding of the outcomes of the performance tests conducted in a real 802.11 WLAN. Specifically, we showed that 1) when operating under the flow-control algorithm, the total TCP throughput is basically independent of the number of open TCP connections and the aggregate TCP traffic can be equivalently modeled as two saturated flows, and 2) in the saturated regime, n UDP flows obtain about

n times the aggregate throughput achieved by the TCP flows, which is independent of the total number of persistent TCP connections.

ACKNOWLEDGMENTS

This work was partially funded by the European Commission Programme ICT-2008-215320 through the EU-MESH Project.

REFERENCES

- [1] IEEE 802.11, The Working Group for WLAN Standards, <http://grouper.ieee.org/groups/802/11/>, Apr. 2006.
- [2] IEEE Standard 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification/Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, IEEE, Sept. 2005.
- [3] Y. Xiao, "IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs," *IEEE Wireless Comm. Magazine*, vol. 12, no. 6, pp. 82-91, Dec. 2005.
- [4] R. Karrer, A. Sabharwal, and E. Knightly, "Enabling Large-Scale Wireless Broadband: The Case for TAPs," *ACM Computer Comm. Rev.*, vol. 34, no. 1, pp. 27-34, Jan. 2004.
- [5] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.*, vol. 18, no. 9, pp. 1787-1800, Mar. 2000.
- [6] F. Cali, M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. Networking*, vol. 8, no. 6, pp. 785-799, 2000.
- [7] R. Bruno, M. Conti, and E. Gregori, "Optimization of Efficiency and Energy Consumption in p -Persistent CSMA-Based Wireless LANs," *IEEE Trans. Mobile Computing*, vol. 1, no. 1, pp. 10-31, Mar. 2002.
- [8] Y. Tay and K. Chua, "A Capacity Analysis for the IEEE 802.11 MAC Protocol," *Wireless Networks*, vol. 7, no. 2, pp. 159-171, Mar./Apr. 2001.
- [9] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New Insights from a Fixed-Point Analysis of Single-Cell IEEE 802.11 WLANs," *Proc. IEEE INFOCOM*, vol. 3, pp. 1550-1561, Mar. 2005.
- [10] F. Alizadeh-Shabdiz and S. Subramaniam, "A Finite Load Analytical Model for IEEE 802.11 Distributed Coordination Function MAC," *Proc. First Workshop Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '03)*, Mar. 2003.
- [11] K. Duffy, D. Malone, and D. Leith, "Modeling the 802.11 Distributed Coordinated Function in Non-Saturated Conditions," *IEEE Comm. Letters*, vol. 9, no. 8, pp. 715-717, Aug. 2005.
- [12] G. Cantieni, Q. Ni, C. Barakat, and T. Turetli, "Performance Analysis under Finite Load and Improvements for Multirate 802.11," *Elsevier Computer Comm. J.*, vol. 28, pp. 1095-1109, June 2005.

- [13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 133-145, Apr. 2000.
- [14] A. Abouzeid, S. Roy, and M. Azizoglou, "Comprehensive Performance Analysis of a TCP Session over a Wireless Fading Link with Queueing," *IEEE Trans. Wireless Comm.*, vol. 2, no. 2, pp. 344-356, Mar. 2003.
- [15] S. Pilosof, R. Ramjee, D. Raz, Y. Shavitt, and P. Sinha, "Understanding TCP Fairness over Wireless LAN," *Proc. IEEE INFOCOM*, vol. 2, pp. 863-872, 2003.
- [16] M. Rossi, R. Vicenzi, and M. Zorzi, "Accurate Analysis of TCP on Channels with Memory and Finite Round-Trip Delay," *IEEE Trans. Wireless Comm.*, vol. 3, no. 2, pp. 627-640, Mar. 2004.
- [17] M. Garetto, R. Lo Cigno, M. Meo, and M. Marsan, "Closed Queueing Network Models of Interacting Long-Lived TCP Flows," *IEEE/ACM Trans. Networking*, vol. 12, no. 2, pp. 300-311, Apr. 2004.
- [18] E. Altman, K. Avrachenkov, and C. Barakat, "A Stochastic Model of TCP/IP with Stationary Random Losses," *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 356-369, Apr. 2005.
- [19] R. Bruno, M. Conti, and E. Gregori, "Design of an Enhanced Access Point to Optimize TCP Performance in Wi-Fi Hotspot Networks," *ACM/Springer Wireless Networks J.*, vol. 13, no. 2, pp. 259-274, Apr. 2007.
- [20] P. Chatzimisios, A. Boucouvalas, and V. Vitsas, "IEEE 802.11 Packet Delay: A Finite Retry Limit Analysis," *Proc. IEEE Global Comm. Conf. (GlobeCom '03)*, vol. 2, pp. 950-954, Dec. 2003.
- [21] V. Vishnevsky and A. Lyakhov, "IEEE 802.11 Wireless LAN: Saturation Throughput Analysis with Seizing Effect Consideration," *Cluster Computing*, vol. 5, no. 2, pp. 133-144, Apr. 2002.
- [22] E. Ziouva and T. Antonakopoulos, "CSMA/CA Performance under High Traffic Conditions: Throughput and Delay Analysis," *Elsevier Computer Comm. J.*, vol. 25, pp. 313-321, Feb. 2002.
- [23] C. Foh and W. Tantra, "Comments on IEEE 802.11 Saturation Throughput Analysis with Freezing of Backoff Counters," *IEEE Comm. Letters*, vol. 9, no. 2, pp. 130-132, Feb. 2005.
- [24] G. Bianchi and I. Tinnirello, "Remarks on IEEE 802.11 DCF Performance Analysis," *IEEE Comm. Letters*, vol. 9, no. 8, pp. 765-767, Aug. 2005.
- [25] Y. Xiao, "A Simple and Effective Priority Scheme for IEEE 802.11," *IEEE Comm. Letters*, vol. 7, pp. 70-72, Feb. 2003.
- [26] J. Hui and M. Devetsikiotis, "A Unified Model for the Performance Analysis of IEEE 802.11e EDCA," *IEEE Trans. Comm.*, vol. 53, no. 9, pp. 1498-1510, Sept. 2005.
- [27] O. Tickoo and B. Sikdar, "A Queueing Model for Finite Load IEEE 802.11 Random Access," *Proc. IEEE Int'l Conf. Comm. (ICC '04)*, vol. 1, pp. 175-179, June 2004.
- [28] R. Bruno, M. Conti, and E. Gregori, "Analytical Modeling of TCP Clients in Wi-Fi Hot Spot Networks," *Proc. IFIP-TC6 Networking Conf.*, pp. 626-637, May 2004.
- [29] R. Bruno, M. Conti, and E. Gregori, "Modeling TCP Throughput over Wireless LANs," *Proc. 17th IMACS World Congress on Scientific Computation, Applied Math. and Simulation*, July 2005.
- [30] R. Bruno, M. Conti, and E. Gregori, "Performance Modeling and Measurements of TCP Transfer Throughput in 802.11-Based WLANs," *Proc. Ninth ACM/IEEE Int'l Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '06)*, Oct. 2006.
- [31] S. Choi, K. Park, and C. Kim, "Performance Impact of Interlayer Dependence in Infrastructure WLANs," *IEEE Trans. Mobile Computing*, vol. 5, no. 7, pp. 829-845, July 2006.
- [32] J. Yu and S. Choi, "Modeling and Analysis of TCP Dynamics over IEEE 802.11 WLAN," *Proc. Fourth IFIP/IEEE Ann. Int'l Conf. Wireless On-Demand Network Systems and Services (WONS '07)*, Jan. 2007.
- [33] D. Miorandi, A. Kherani, and E. Altman, "A Queueing Model for HTTP Traffic over IEEE 802.11 WLANs," *Elsevier Computer Networks J.*, vol. 50, no. 1, pp. 63-79, Jan. 2006.
- [34] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of Reliable Transport Protocol over IEEE 802.11 Wireless LAN: Analysis and Enhancement," *Proc. IEEE INFOCOM*, pp. 599-607, June 2002.
- [35] M. Arranz, R. Aguero, L. Munoz, and P. Mahonen, "Behavior of UDP-Based Applications over IEEE 802.11 Wireless Networks," *Proc. 12th Int'l Symp. Personal, Indoor and Mobile Radio Comm. (PIMRC '01)*, vol. 2, pp. 72-77, 2001.
- [36] S. Garg and M. Kappes, "An Experimental Study of Throughput for UDP and VoIP Traffic in IEEE 802.11b Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03)*, vol. 3, pp. 1748-1753, Mar. 2003.
- [37] R. Bruno, M. Conti, and E. Gregori, "Throughput Analysis of UDP and TCP Flows in IEEE 802.11b WLANs: A Simple Model and Its Validation," *Proc. FIRB-Perf Workshop Techniques, Methodologies and Tools for Performance Evaluation of Complex Systems*, pp. 54-63, Sept. 2005.
- [38] H. Balakrishnan, N. Padmanabhan, S. Seshan, M. Stemm, and R. Katz, "TCP Behavior of a Busy Internet Server: Analysis and Improvements," *Proc. IEEE INFOCOM*, vol. 1, pp. 252-262, 1998.
- [39] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Trans. Networking*, vol. 7, no. 4, pp. 458-472, Aug. 1999.
- [40] A. Parekh and R. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. Networking*, vol. 1, no. 2, pp. 344-357, June 1993.
- [41] S. Floyd and V. Jacobson, "Link-Sharing and Resource Management Models for Packet Networks," *IEEE/ACM Trans. Networking*, vol. 3, no. 4, pp. 365-386, Aug. 1995.
- [42] D. Stidialis and A. Varma, "Efficient Fair Queueing Algorithms for Packet-Switched Networks," *IEEE/ACM Trans. Networking*, vol. 6, no. 2, pp. 175-185, Apr. 1998.
- [43] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Networking*, vol. 1, no. 4, pp. 397-413, Aug. 1993.
- [44] R. Bruno, M. Conti, and E. Gregori, "Throughput Analysis and Measurements in IEEE 802.11 WLANs with TCP and UDP Traffic Flows," technical report, IIT CNR, <http://bruno1.iit.cnr.it/bruno/techreport.html>, Sept. 2006.
- [45] D. Heyman and M. Sobel, *Stochastic Models in Operations Research*, vol. 1. McGraw-Hill, 2001.
- [46] W. Stevens, "The Protocols," *TCP/IP Illustrated*, vol. 1. Addison-Wesley, Dec. 1993.
- [47] *ANSI/IEEE Standard 802.11b, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification/Amendment 2: Higher-Speed Physical Layer (PHY) in the 2.4 GHz Band*, IEEE, Nov. 2001.
- [48] D. Bovet and M. Cesati, *Understanding the Linux Kernel*, third ed. O'Reilly, Nov. 2005.
- [49] G. Xylomenos and G. Polyzos, "TCP and UDP Performance over a Wireless LAN," *Proc. IEEE INFOCOM*, vol. 2, pp. 439-446, Mar. 1999.
- [50] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saarinen, "TCP Performance Issues over Wireless Links," *IEEE Comm. Magazine*, vol. 39, no. 4, pp. 52-58, Apr. 2001.
- [51] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *Proc. IEEE INFOCOM*, vol. 2, pp. 836-843, 2003.
- [52] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, and M. Meo, "Smart Traffic Scheduling in 802.11 WLANs with Access Point," *Proc. 58th IEEE Vehicular Technology Conf. (VTC 2003-Fall)*, vol. 4, pp. 2227-2231, Oct. 6-9, 2003.
- [53] M. Bottigliengo, C. Casetti, C.-F. Chiasserini, and M. Meo, "Short-Term Fairness for TCP Flows in 802.11b WLANs," *Proc. IEEE INFOCOM*, vol. 2, pp. 1383-1392, Mar. 2004.
- [54] D. Leith, P. Clifford, D. Malone, and A. Ng, "TCP Fairness in 802.11e WLANs," *IEEE Comm. Letters*, vol. 9, no. 11, pp. 964-966, Nov. 2005.



Raffaele Bruno received the Laurea degree in telecommunication engineering and the PhD degree in computer engineering from the University of Pisa, Italy, in 1999 and 2003, respectively. He is a researcher at the Institute for Informatics and Telematics, Italian National Research Council (IIT CNR). He has served the technical program committee (TPC) of several international conferences and workshops, including the 2006 IEEE International Conference

on Communications (ICC) and the Seventh and Eighth IEEE International Symposia on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2006 and 2007). He was the workshop cochair of the Second IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2006) and is currently the workshop cochair of the First IEEE International Workshop on Mobile Ad Hoc and Sensor Systems for Global and Homeland Security (MASS-GHS07). He is currently a guest editor of a special issue on homeland and global security of the *Pervasive and Mobile Computing Journal*. His main research interests include the modeling and performance evaluation of MAC and transport protocols for wireless networks. He has published more than 30 papers in international journals and conference proceedings.



Enrico Gregori received the Laurea degree in electronic engineering from the University of Pisa in 1980. He joined Centro Nazionale Universitario di Calcolo Elettronico (CNUCE), an institute of the Italian National Research Council (CNR), in 1981. In 1986, he held a visiting position at the IBM Research Center, Zurich, working on network software engineering and heterogeneous networking. He has contributed and is currently working on several national

and international projects on computer networking, including FP5 FET-IST MOBILEMAN and FP6 FET-SAC BIONETS. He is currently with CNR as a research director and the Institute for Informatics and Telematics (IIT)-CNR as the acting director. He was the general chair of the IFIP-TC6 conferences Networking 2002 and Personal Wireless Communications (PWC) 2003 and the Fourth IEEE Pervasive Computing and Communication (PerCom 2006). He served as a guest editor of the *Networking2002 Journal* (special issues on performance evaluation and cluster computing) and the *ACM/Kluwer Wireless Networks Journal*. He is on the editorial board of the *Cluster Computing Journal*, the *Computer Networks Journal*, and the *Wireless Networks Journal*. He is the author of more than 100 papers in computer networks, has published in international journals and conference proceedings, and is a coauthor of *Metropolitan Area Networks* (Springer, 1997).

► **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.**



Marco Conti is a research director at the Institute for Informatics and Telematics, Italian National Research Council (IIT CNR). He served as the technical program committee (TPC) chair of the Fourth IEEE International Conference on Pervasive Computing and Communications (PerCom 2006) and of the IFIP-TC6 conferences Networking 2002 and Personal Wireless Communications (PWC) 2003. He served as a TPC cochair of the Fifth ACM International

Workshop on Wireless Mobile Multimedia (WoWMoM 2002), the Second IEEE Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt 2004), the Sixth IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2005), and the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2006). He served as a general cochair of the IEEE WoWMoM 2006 and as the general chair of the Second ACM International Workshop on Multihop Ad Hoc Networks: From Theory to Reality (REALMAN 2006). He is currently the general chair of the Fourth IEEE International Conference on Mobile Ad Hoc and Sensory Systems (MASS 2007). He is an associate editor-in-chief of *Pervasive and Mobile Computing* and he is on the editorial board of the *IEEE Transactions on Mobile Computing, Ad Hoc Networks*, and *Wireless Ad Hoc and Sensor Networks: An International Journal*. He is a coauthor of the *Metropolitan Area Networks* (1997) and a coeditor of *Mobile Ad Hoc Networking* (2004) and *Mobile Ad Hoc Networks: From Theory to Reality* (2007). He has published in journals and conference proceedings more than 180 research papers related to the design, modeling, and performance evaluation of computer network architectures and protocols.