Volume: 01 Issue: 03 Pages: 198-204 (2009)

Throughput Analysis in 802.11 WLANs using TCP

Chandra Mouli Venkata Srinivas, Akana,
Associate Professor, AMC Engineering College, Bangalore, Karnataka, India.
Email: mouliac@yahoo.co.in
Dr. C. Divakar
Professor & Principal, Visakha Institute of Technology, Visakhapatnam, AP, India
Email: divakar_c@yahoo.com
Mr. G. Sunil Kumar
Student, UVCE, Bangalore, Karnataka, India.
Email: gsunil_nsj@yahoo.com
Prof. K Sundeep kumar
Assistant Professor, AMC Engineering College, Bangalore, Karnataka, India.
Email: sundeepkk@yahoo.co.in
Prof. K.P.Sirisha
Assistant Professor, AMC Engineering College, Bangalore, Karnataka, India.
Email: phanisirisha76@yahoo.com

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 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.

Keywords: MAC protocol, TCP traffic, Throughput, UDP datagrams, WLAN.

Date of Submission: September 14, 2009 Accepted : November 09, 2009

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, metro scale 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 these research areas 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 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 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. 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 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.

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.

2. System Model for Network and Station Model

In this paper, 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.

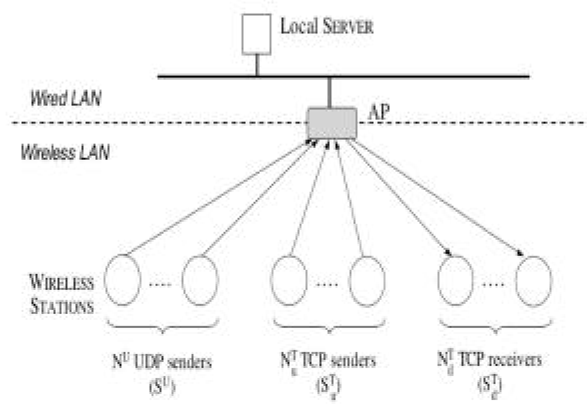

Fig. 1. The network model: network and traffic configurations.

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^Td TCP downstream flows (that is, N^Td stations are the receivers of TCP downlink connections) and NT u TCP upstream flows (that is, N^Tu 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^Tu a station behaving as the sender for TCP upstream flows, and with S^Td 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 widely recognized that drastic unfairness occurs when TCP flows share the same buffer with unresponsive UDP flows, competing for scarce bandwidth. 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. 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 back off process.

2.1 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.

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.

3. Throughput Analysis of UDP And TCP Flows

Our analysis is based on a Markov renewal-reward approach. 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. 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 station's output queues without considering UDP streams. 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.

3.1 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, 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.

We start our analysis from the consideration that, under the assumptions listed, it holds that the state of each TCP connection 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 back off 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. . . .N^T), we indicate with $X^i_{tcp}(t)$ the stochastic process representing the number of TCP data segments stored in the TCP sender's output queue at time t and with $X^i_{ack}(t)$ 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

For every t,

: $X^i_{tcp}(t) + X^i_{ack}(t) = W$ i= 1..........N^T ——— (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 $X^i_{tcp}(t)$ and $X^i_{ack}(t)$ However, instead of developing a per-connection model, we may describe the system evolution by using the aggregate stochastic processes $X^i_{tcp}(t) = {}_{i=1}?^{NT}\ X^i_{tcp}(t)$ and $X^i_{ack}(t) = {}_{i=1}?^{NT}\ X^i_{ack}(t)$. In addition, the bidimensional process { $X^i_{tcp}(t)$, $X^i_{ack}(t)$ } can be modeled using the equivalent bidimensional process { $Y^i_{tcp}(t)$, $Y^i_{ack}(t)$ }, where $Y^i_{tcp}(t)$ expresses the total number of TCP data segments that are stored in the transmission queues of all the $S^T u$ stations at time t, and $Y^i_{ack}(t)$ expresses the total number of TCP ACKs stored in the transmission queues of all the $S^T d$ 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 $S^T u$ and $S^T d$ stations. Our aim is to compute the probability that, in the steady state conditions, there are i TCP data segments in the $S^T u$ stations' transmission queues and j TCP ACK packets in the $S^T d$ stations' transmission queues. This probability can be formally defined as follows:

: $b_{i,\ j\ =} \lim_{t?\ 8}$ P{ $Y_{tcp}(t)$ =i , $Y_{ack}(t)$ =j} i ? $(0.m_u)$, j ? $(0.m_d)$, ----- (2)

Where $m_u = N^T u$ .W and $m_u = N^T d$ .W. For the sake of brevity, in the following, we adopt the short notation $Y_{i,j}$ to denote the system state such that {$Y^i_{tcp}(t)$ =i,$Y^i_{ack}(t)$ =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, $S^T u$ and $S^T d$ 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 ? (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.

3.2. 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. 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[$K^{\wedge}$] 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[$K^{\wedge}$] are reported). Table 1 compares the measured E[K] and E[$K^{\wedge}$] 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 1 through real experiments.

W	N_u^T	N_d^T	$E[\hat{K}]$ analysis	$E[\hat{K}]$ simulation	$E[K]$ analysis	$E[K]$ 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

Table 1 : Study of the Average Network Activity

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.

Volume: 01 Issue: 03 Pages: 198-204 (2009)

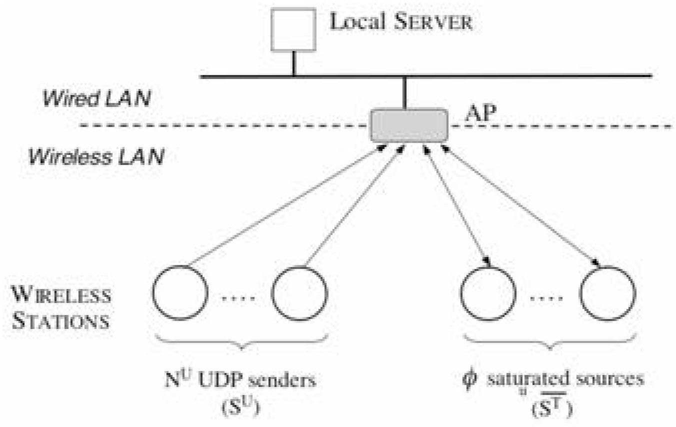

Fig. 2. The equivalent network model: network and traffic configurations

4. 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.

4.1 Performance Evaluation for N^U

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

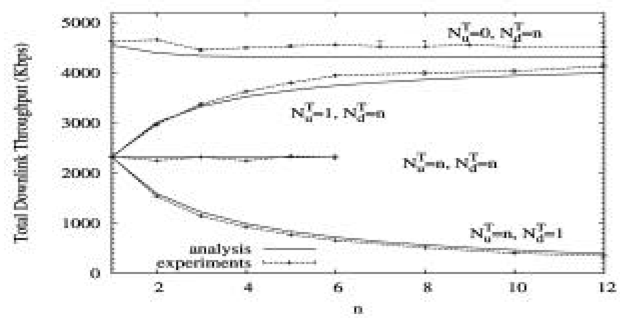

Fig. 3. Comparison of model predictions and measured l^{d}.

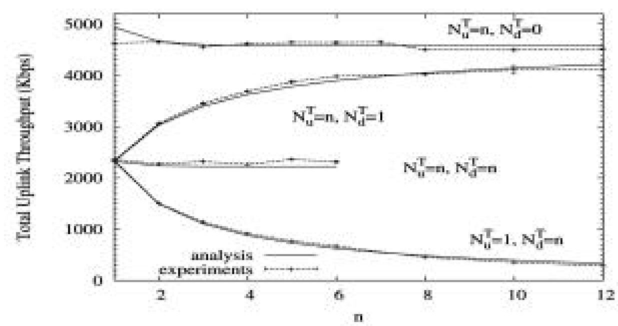

Fig. 4. Comparison of model predictions and measured l^{u}.

Fig. 3 shows the total throughput obtained by the TCP downlink connections (that is, the l^D value, as defined in (5)), whereas Fig. 6 shows the aggregate throughput of theTCP uplink connections (that is, the l^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.

Volume: 01 Issue: 03 Pages: 198-204 (2009)

5. Conclusion

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.

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,

[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' 00.

[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

[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,

[12] G. Cantieni, Q. Ni, C. Barakat, and T. Turletti, "*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. 2, 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,

[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.

Volume: 01 Issue: 03 Pages: 198-204 (2009)

Authors Biography

Chandra Mouli Venkata Srinivas Akana, received the Masters Degree in Computer Science & Engineering from Guru Jambheshwar University Of Science & Technology, Hisar, Haryana, India. He is currently working as Associative Professor in the Department of Computer Science & Engineering, AMC Engineering College, Bangalore, Karnataka, India. He has occupied various academic responsibilities like practical examiner, project adjudicator for various Universities and Colleges in and around in Andhra Pradesh & Karnataka. His current research is focused on Software Engineering, Ad-hoc Networks, Network Computing, Image Processing & Database Management System. He has published several papers in National & International Journals. He has guided to more than 60 academic projects at Post graduate & Under Graduate level. He is a member of the IEEE, CSI & ISTE.

Dr. Divakar C. received his Masters Degree as well as Doctorate in Computer Science & Engineering from Andhra University, Visakhaptnam, Andhra Pradesh, India. He is currently working as Professor & Principal for Visakha Institute of Technology, Visakhapatnam, Andhra Pradesh, India. He has occupied various academic responsibilities like practical examiner, project adjudicator for various Universities and Colleges in and around in Andhra Pradesh. He is guiding 4 Ph.D. research Scholars. His current research is focused on Bio-informatics, Software Engineering, Ad-hoc Networks, Network Computing, Image Processing & Database Management System. He has published several papers in National & International Journals. He has guided several academic projects at Post graduate & Under Graduate level.

G Sunil Kumar pursuing ME (CSE) from UVCE, Bangalore and B.Tech (IT) from JNTU, Hyderabad. He presented more than 4 papers in International and national Conferences. . He worked as a lecturer in the department of Information Technology, ADAMS Engineering College, AP. His research interests include Software Engineering, Wireless Sensor Networks and Data Warehousing. He is a member in ISTE.

K Sundeep Kumar received the M.Tech(IT) from Punjabi University in 2004, and ME (CSE) from Anna University in 2009. He is with the department of Information Science & Engineering and as an Asst. Professor, and as a Placement Officer, AMC Engineering College, Banglaore. He presented more than 10 papers in International and national Conferences. His research interests include OOAD, Software Engineering and Data Warehousing. He is a member in ISTE.

Phani Sirisha K. received the Masters Degree in Computer Applications from Andhra University, Visakhapatnam, Andhra Pradsh, India. She is currently working as Assistant Professor in the Department Professor in the of Master of Computer Applications, AMC Engineering College, Bangalore, Karnataka, India. She has occupied various academic responsibilities like practical examiner, project adjudicator for Visvesvaraya Technological University and Colleges in and around in Karnataka. Her current research is focused on Software Engineering, Ad-hoc Networks, Network Computing, Image Processing & Database Management System. She has guided to more than 20 academic projects at Post graduate level.