Average-Value Analysis of 802.11 WLANs with Persistent TCP flows

R. Bruno, M. Conti, E. Gregori

IIT TR-02/2008
Technical report

Febbraio 2008

Istituto di Informatica e Telematica
Average-Value Analysis of 802.11 WLANs with Persistent TCP flows

Raffaele Bruno, Marco Conti, Enrico Gregori
Institute for Informatics and Telematics (IIT)
Italian National Research Council (CNR)
Via G. Moruzzi, 1 - 56124 PISA, Italy
Email: {r.bruno,m.conti,e.gregori}@iit.cnr.it

Abstract
The widespread use of the IEEE 802.11 MAC as a layer-2 protocol for wireless local area networks has generated an extensive literature on its performance modeling. However, most of the available studies evaluate the capacity of WLANs in saturated conditions, while very little has been done on investigating the interactions between the 802.11 MAC protocol and the various transport protocols that are used to deliver users’ traffic. Recently there have been renewed efforts to understand and model the TCP dynamics in 802.11 WLANs. In general, these models employ multi-dimensional discrete-time Markov chains to analyze the distributions of the number of TCP packets enqueued in the stations’ buffers. Then, they exploit those distributions to derive both the average number of active TCP stations and the aggregate TCP throughput. However, this approach may rapidly lead to the explosion of the model state-space when the number of TCP flows is large. In this technical report we propose a novel modeling approach by developing an average-value analysis of TCP performance in 802.11 WLANs. Our model intuitively characterizes the equilibrium conditions for the network, and this method yields a precise estimate of the throughput of persistent TCP flows. Extensive simulations validate the accuracy of our analysis.

Index Terms
802.11 MAC protocol, TCP, performance analysis.

I. INTRODUCTION
Driven by the standardization and rapid deployment of IEEE 802.11 WLANs, the performance analysis of its contention-based MAC protocol has been studied extensively. Three major seminal models laid the foundations for most of the analytical studies on 802.11 saturation throughput: i) Bianchi [1] proposed a Markov chain to describe the 802.11 backoff behavior; ii) Calí et al. [2] introduced a p-persistent variant of the 802.11 MAC; and iii) Tay [3] used instead an average-value analysis to describe the system properties.

The models in [1]–[3], as well as their various extensions, have many commonalities (e.g., all of them apply regenerative/renewal theory), and provide an accurate estimate of the maximum saturation throughput. However, most of the Internet traffic is carried over the TCP, that is a feedback-based flow-controlled transport protocol. The complex interactions between the collision avoidance mechanisms implemented by the 802.11 MAC and the closed-loop behavior of TCP cannot be correctly described using those popular modeling approaches. For these reasons, recently there have been renewed efforts to analyze the TCP dynamics in 802.11 WLANs [4]–[6]. In general, these models employ multi-dimensional discrete-time Markov chains to analyze the stationary distributions
of the number of TCP packets enqueued in the stations’ buffers. Then, they exploit those distributions to derive both the average number of active TCP stations (i.e., with at least a packet to transmit) and the aggregate TCP throughput. However, this approach may rapidly lead to the explosion of the model state-space when the number of TCP flows is large.

In this technical report, we propose a novel modeling approach to analyze the throughput performance of persistent TCP flows in 802.11 WLANs. We focus on studying the equilibrium behavior of the network, and we derive explicit expressions to mathematically derive the equilibrium conditions for the system. In addition, our model uses the average value for a variable wherever possible. This technique is simple, yet effective because it provides an intuitive procedure to analytically evaluate the average number of active TCP stations in the network, thus simplifying the analysis of the throughput performance. To show the versatility of our analysis we apply it to both the standard 802.11 MAC and an enhanced variant, which allows the access point to transmit bursts of consecutive frames [7]. Extensive simulations validate the accuracy of our analysis.

The rest of this technical report is organized as follows. In Section II we present the system model, and we discuss the modeling assumptions. Section III develops the mathematical analysis of the throughput performance of persistent TCP flows. In Section IV we show numerical results validating the model accuracy. Section V presents concluding remarks and discusses future work.

II. System Model

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

A. Network and station model

We consider a typical WLAN, where \( n \) long-lived TCP stations are associated to an access point (AP). An FTP server is co-located at the AP, and \( n^d \) (\( n^u \)) TCP stations are downloading (uploading) infinite size files from (to) the server. The DCF-based 802.11 MAC regulates the channel access. However, we assume that the AP, differently from regular TCP stations, is allowed to transmit bursts of \( m \) consecutive frames. This variant of the 802.11 MAC protocol has been firstly proposed in our previous paper [7]. Note that all the 802.11 enhancements currently under standardization (e.g., 802.11e and 802.11n [8]) are introducing multi-frame transmissions to improve the network capacity. The standard 802.11 MAC corresponds to \( m = 1 \).

From the perspective of the MAC protocol, the network could be modeled as a set of queues (i.e., a set of MAC buffers). The access control policy employed by the MAC protocol determines the order these queues (i.e., the stations) are served and the related service time, which is the time a frame spends in the MAC layer from the instant of leaving the transmission buffer until its successful transmission. It is evident that frames may have different service times depending on the number of non-empty queues in the network. Indeed, in real 802.11 networks the saturated queue assumption is unlikely to be valid and several factors affect the average number of non-empty queues. One of these factors is clearly the traffic arrival pattern (for instance, TCP flows may generate variable traffic loads and the inter-arrival time between packets can result into large idle periods). In addition, the complex interactions between the collision avoidance mechanisms implemented by the 802.11 MAC and the...
closed-loop behavior of TCP also play a crucial role in determining the state of transmission buffers. However, in this technical report we will show that it is not necessary to exactly describe the evolution of the MAC queues to derive an accurate estimate of the TCP throughput performance in a WLAN.

B. 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 timeouts are sufficiently long to avoid timeout expirations due to 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 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 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 we have adopted to cope with the problem complexity are A.1 and A.2. In fact, basing 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 acknowledgments) for a TCP flow is fixed and equal to the maximum TCP receive window size $W$ (in packets). 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 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 [6], [9] 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 [5], [6], [10], [11] on analysing the TCP dynamics in 802.11 WLANs.
Fig. 1. Time-line of the $k(\nu)$ evolution during $\eta_{\nu}$.

III. THROUGHPUT ANALYSIS OF PERSISTENT TCP FLOWS IN WLANS

In this section we develop our average-value analysis. Specifically, we firstly derive the average number of active TCP stations in the system. Then, we use this result to compute the throughput of persistent TCP flows in WLANs.

A. Average Number of Active TCP Stations

Let $a(k)$ be a random variable representing the number of active TCP stations when there are $k$ TCP packets (i.e., counting both TCP data packets and TCP ACKs) stored in the TCP stations’ buffers. Furthermore, let $k(\nu)$ be the discrete-time stochastic process representing the number of TCP packets stored in the TCP stations’ buffers at the end of the $\nu^{th}$ AP’s successful transmission. Finally, let $\eta_{\nu}$ be the time interval between the end of the $\nu^{th}$ and $(\nu+1)^{th}$ AP’s successful transmission.

Firstly, we elaborate our analysis to derive the steady-state expected value of the $k(\nu)$ process, which is defined as $\lim_{\nu \to \infty} E[k(\nu)] = k^*$, with $E[\cdot]$ being the average operator. To this end, Fig. 1 illustrates how the $k(\nu)$ process evolves during $\eta_{\nu}$. Let $s(k(\nu))$ be a random variable expressing the number of successful transmissions performed by the active TCP stations during $\eta_{\nu}$. It is straightforward to observe that the total number of TCP packets stored in the TCP stations’ buffers decreases by one after each TCP station’s successful transmission. Thus, immediately before the $(\nu+1)^{th}$ AP’s successful transmission, the total number of TCP packets stored in the TCP stations’ buffers will be equal to $k(\nu) - s(k(\nu))$. On the other hand, when the AP gains the channel access it sends a burst of consecutive frames. Intuitively, the burst size is given by the minimum between the system parameter $m$ and the number of available TCP packets in the AP’s buffer, being $nW$ the total number of packets in the network.

To derive the $k^*$ value, we adopt a $p$-persistent model of the 802.11 DCF as in [5] and [6]. From the geometric backoff assumption it follows that all the processes that define the occupancy behavior of the channel are regenerative, with the time instants corresponding to the completion of an AP’s successful transmission being the regeneration epochs. This means that the system behavior is a probabilistic replica of itself after each regeneration epoch. In other words, the $k(\nu)$ process should probabilistically replicate after each AP’s successful transmission, with the $\eta_{\nu}$ interval being the renewal period. Formally, this concept can be expressed as $\lim_{\nu \to \infty} E[k((\nu+1)) - k(\nu)] = 0$, which represents the equilibrium condition for the existence of a steady-state regime for the $k(\nu)$ process. Owing
to the previous considerations, this equilibrium condition can be formulated as

$$k^* = k^* - E[s(k^*)] + \min\{m, nW - k^* + E[s(k^*)]\}. \quad (1)$$

The following lemma, provides an explicit expression for the $E[s(k^*)]$ quantity.

**Lemma 1:** In a WLAN with $n$ stations, under the assumptions that transmission attempts are i.i.d., $k^*$ packets are stored in the stations’ buffers, and $nW$ are the total number of packets in the network, it holds that

$$E[s(k^*)] = \sum_{i=1}^{k^*} i \cdot \frac{I_{AP}(k^*-i)}{I_{AP}(k^* - i) + E[a(k^*-i)]} \cdot \prod_{j=0}^{i-1} \frac{E[a(k^*-j)]}{I_{AP}(k^* - j) + E[a(k^*-j)]}, \quad (2)$$

where $I_{AP}(x)$ is an indicator function equal to one if the AP’s buffer is not empty (i.e., for $x < nW$), and

$$E[a(k)] = n \left[ 1 - \left( 1 - \frac{1}{n} \right)^k \right]. \quad (3)$$

**Proof:** Let $E[s(k)]$ be the average number of TCP stations’ successful transmissions happening during a generic renewal period $\eta_k$ (i.e., between two consecutive AP’s successful transmissions), conditioned to having $k$ TCP packets stored in the TCP stations’ buffers at the beginning of $\eta_k$. By definition, $E[s(k)]$ is given by

$$E[s(k)] = \sum_{i=1}^{k} i \cdot P(i, k), \quad (4)$$

where $P(i, k)$ is the probability that there are exactly $i$ TCP stations’ successful transmissions during $\eta_k$. To compute the $P(i, k)$ probability, we introduce two auxiliary probabilities. Specifically, let $P_{ST A,k}^{\text{succ}}$ be the probability that a TCP station succeeds in a generic transmission attempt, and let $P_{AP,k}^{\text{succ}}$ be the probability that the AP succeeds in a generic transmission attempt, both conditioned to having $k$ TCP packets in the TCP stations’ buffers. Assuming that the transmission attempts are i.i.d. ([1], [2]), it holds that

$$P_{AP,k}^{\text{succ}} = \frac{I_{AP}(k)}{I_{AP}(k) + E[a(k)]}, \quad P_{ST A,k}^{\text{succ}} = 1 - P_{AP,k}^{\text{succ}}, \quad (5)$$

where $E[a(k)]$ is the average number of active TCP stations with $k$ TCP packets enqueued in the TCP stations’ buffers. Now, it is straightforward to write that

$$P(i, k) = \prod_{j=0}^{i-1} P_{AP,k-j}^{\text{succ}} \cdot P_{AP,k-i}^{\text{succ}} = \frac{I_{AP}(k*-i)}{I_{AP}(k*-i) + E[a(k*-i)]} \cdot \prod_{j=0}^{i-1} \frac{E[a(k*-j)]}{I_{AP}(k*-j) + E[a(k*-j)]}. \quad (6)$$

Now, by substituting (6) in (4), and computing $E[s(k=k^*)]$ we obtain formula (2).

Note that in Equation (4) we have implicitly assumed that $k$ is an integer to properly define the summation. However, in principle $k$ might be a real number and not a discrete value. Several techniques can be considered to map a real value on a set of discrete values. The approximation we adopted here is to consider two values, say $k_l$ and $k_h$, defined as the greatest integer smaller than $k$ and the smallest integer greater than $k$, respectively. Then,
we compute $E[s(k_l)]$ and $E[s(k_h)]$ using (4), and we estimate $E[s(k)]$ through a linear interpolation of these two values. For the sake of brevity, we assume the $E[s(k^*)]$ value defined in (2) is the result of this interpolation.

Formula (6) shows that $P(i, k)$ depends on the $E[a(k)]$ value. Thus, the last step in our proof is to derive an explicit expression for this quantity. For the general case of $W > 0$, it may be very complicate to compute the exact mass distribution of the $a(k)$ random variable. For simplicity, in our study we assume that there are no limitations on how $k$ TCP packets can be distributed over the $n$ TCP stations (this is equivalent to assume that $W \to \infty$). Under this assumption, the derivation of the $E[a(k)]$ value recalls the well-studied balls-in-bins problem. Specifically, if we have $n$ bins and we randomly throw balls into them, after $k$ tosses, the average number of bins that contain exactly $h$ balls is given by

$$e(h, k) = n \frac{k!}{(k-h)!} \frac{1 - \frac{1}{n}}{h!} \left(1 - \frac{1}{n}\right)^{k-h} \left(\frac{1}{n}\right)^h.$$  

(7)

According to formula (7), the average number of TCP stations with empty buffer with $k$ total TCP packets can be estimated as $e(0, k)$. Thus, the average number of active TCP stations can be approximated as follows

$$E[a(k)] = n - e(0, k) = n \left[1 - \left(1 - \frac{1}{n}\right)^k\right],$$  

(8)

and this concludes the proof.

By substituting (2) in (1), $k^*$ becomes the only independent variable in equation (1), which can be solved with simple numerical techniques.

Now, let us denote with $E[a|Succ^{AP}]$ the average number of active TCP stations at the end of an AP’s successful transmission, and with $E[a|Pre\ Succ^{AP}]$ the average number of active TCP stations immediately before an AP’s
successful transmission. Using (3) it is immediate to write that $E[a|Succ^{AP}] = E[a(k^*)]$ and $E[a|Pre\ Succ^{AP}] = E[a(k^* - E[s(k^*)])]$. However, our ultimate aim is to derive $E[a|Succ]$, i.e., the average number of active TCP stations after a generic successful transmission. To this end, Fig. 2 illustrates the evolution of the number of active TCP stations during an average renewal period $E[\eta]$. Note that in the following elaboration we use the average value for a variable wherever possible. As expressed in equation (1), during $E[\eta]$ the number of TCP packets enqueued in the TCP stations’ buffers decreases from $k^*$ to $k^* - E[s(k^*)]$, on average. To generalize, let us define $\Omega_i = \{j : k^* - E[s(k^*)] \leq j \leq k^*, E[a(j)] = i\}$, and let $\sigma(i)$ be the number of elements in the set $\Omega_i$. Thus, $\sigma(i)$ represents the number of TCP stations’ successful transmissions that are performed during $E[\eta]$ with $i$ active TCP stations, on average. Let $\mu_i$ be the long-term proportion of successful transmissions occurring with $i$ active TCP stations in the network. By definition, $\mu_i$, is given by

$$\mu_i = \sigma(i)/E[s(k^*)].$$  

(9)

To complete the derivation of the $E[a|Succ]$ value, let us define $\Delta = \{i : E[a|Pre\ Succ^{AP}] \leq i \leq E[a|Succ^{AP}]\}$ and $\exists j \in [k^* - E[s(k^*)], k^*] : a(j) = i\}$, which is the set of possible values for the process representing the number of active TCP stations during $E[\eta]$. Now, it is straightforward to write that

$$E[a|Succ] = \sum_{i \in \Delta} i \cdot \mu_i.$$  

(10)

B. Aggregate TCP Throughput

To derive the aggregate TCP throughput, say $S$, we exploit the knowledge of the average number of active TCP stations in the WLAN. Specifically, we introduce an equivalent network composed of an equivalent saturated AP and $E[a|Succ]$ equivalent saturated TCP stations. The equivalent saturated AP transmits bursts of consecutive frames, and the average burst size is $min\{m, nW - k^* + E[s(k^*)]\}$ frames. Moreover, the percentage of TCP data packets and TCP ACKs in a burst is $n_d/n$ and $n_u/n$, respectively, where $n_d$ is the number of TCP downloads and $n_u$ is the number of TCP uploads. On the other hand, each equivalent saturated TCP station will asymptotically transmit TCP data packets and TCP ACKs with probability $n_u/n$ and $n_d/n$, respectively. Note that introducing these equivalent saturated traffic sources completely hides the complexities of the TCP flow-control dynamics [6].

By applying the $p$-persistent model proposed in [2] to the saturated network composed of $n^* = E[a|Succ]+1$ nodes described above, we can easily derive the TCP throughput. Specifically, once the per-slot transmission probability $p$ is known (e.g., using the algorithm described in Appendix B of [2]), the throughput $S$ can be computed as [2]:

$$S = \frac{E[P]}{(E[N_c]+1) \cdot E[Idle\ P] + E[N_c] \cdot E[T_C] + E[T_S]},$$  

(11)

where the denominator represents $E[T_v]$, i.e., the average time between two successful transmissions, and $E[P]$ is the average number of TCP-payload bytes transmitted with each successful transmission. Note that the $E[P], E[T_c]$ and $E[T_s]$ values differ with respect to the corresponding ones defined in [2] because they depend on the packet size distributions. Explicit expressions for those quantities are derived in Appendix I.
IV. Numerical Validation

In this section, we check the accuracy of our analysis by comparing the model predictions with the results obtained from an event-driven simulator. Following curves show both average values and 95% confidence intervals, which are very tight (lower than 1%).

Fig. 3 shows the $E[a|Succ]$ value computed using formula (10) (dashed lines), and the simulation results (solid lines), for different numbers of downlink TCP flows and $W$ values. The numerical results indicate that there is a good match between the analysis and the simulations. In addition, the numerical results show that the $E[a|Succ]$ value is weakly dependent on the TCP receive window size $W$. Another interesting observation derived from the plotted curves is that the average number of TCP active stations is very close to $n$ when $m\geq n$. We have investigated other traffic configurations, with different $n^u$ and $n^d$ values, obtaining similar results that are not shown here due to space limitations.

Fig. 4 and Fig. 5 show the aggregate download and upload TCP throughput, respectively, versus the burst size, for $W = 16$ and fixed TCP payload-size equal to 1448 bytes. First of all, we can observe that our model accurately predicts the total TCP throughput, and small discrepancies are appreciable only when $n$ is large, i.e., when the $E[a|Succ]$ estimate is less accurate (see Fig. 3). The second important observation is that there is a small increase of the downlink TCP throughput by increasing the burst size, but the TCP throughput rapidly flattens out. Surprisingly, an increase of the burst size may negatively affect the uplink TCP throughput. To explain these counter-intuitive results we should recall that an increase of the burst size yields an increase in the number of active TCP stations. Consequently, the collision probability increases as well, reducing the system efficiency. This effect
is more noticeable for uplink flows than downlink flows because in the former case the TCP stations transmit TCP data packets, and the collisions are more costly in terms of bandwidth wastage.

V. CONCLUSIONS

In this technical report, we have shown that the throughput performance of TCP persistent flows in 802.1 WLANs can be accurately estimated once the average number of active TCP stations is known. Our average-value analysis permits to easily characterize the equilibrium behavior of the network, without using complex and large-scale DTMCs. Our model is versatile, and it can be used to analyze the various enhancements of 802.11 MAC currently under standardization (e.g., multi-frame transmissions), and to derive optimization conditions. For instance, in this report we have shown that allowing multiple-frame transmissions is not always beneficial, and in some configurations it can even negatively affect the throughput performance.

REFERENCES


In this appendix we derive the $E[T_v]$ expression, which was introduced in Section III-B as follows:

$$E[T_v] = (E[N_c] + 1) \cdot E[Idle_p] + E[N_c] \cdot E[T_C] + E[T_S],$$

where $E[Idle_p]$ is the average idle period that precedes each transmission attempt, $E[N_c]$ is the average number of collisions happening between two consecutive successful transmissions, $E[T_C]$ is the average duration of a collision, and $E[T_S]$ is the average duration of a successful transmission.

Our analysis is based on the network model defined in Section III-B. Specifically, we analyze a saturated network composed of $n^* = E[a|Succ] + 1$ nodes. The AP asymptotically transmits bursts of $m^*$ consecutive frames, with $m^* = \min\{m, nW - k^* + E[s(k^*)]\}$. Indicating with $P_{AP}$ the payload length (in bytes) of a frame delivered by the AP, the cumulative distribution function of the $P_{AP}$ random variable is defined as $F_{AP}(l) = Pr\{P_{AP} \leq l\}$. Similarly, each one of the $E[a|Succ]$ TCP stations asymptotically transmits frames with payload length equal to $P_{STA}$ bytes.

Then, the cumulative distribution function of the $P_{STA}$ random variable is defined as $F_{STA}(l) = Pr\{P_{STA} \leq l\}$. Note that we develop the following analysis for the most general case, i.e., assuming general $F_{STA}(l)$ and $F_{AP}(l)$ functions, and at the end of this appendix we instantiate the formulas to the frame distributions considered in Section III-B.

Before developing the analytical derivations, it is useful to introduce some notations adopted during the following analysis:

- $1_{AP}$ is an indicator function that has value 1 when the channel is busy and the AP is transmitting, while is 0 when the channel is busy but the AP is not transmitting.
- **SIFS, DIFS, EIFS, and SLOT** are the interframe spaces and the slot time used in the 802.11b MAC protocol [12], respectively.

- **t_B, t_{H_{MAC}}, and t_{ACK}** are the time needed to transmit a byte, the MAC header, and the MAC acknowledgment frame at the data rate \( r \), respectively. \( t_{H_{PHY}} \) is the duration of the physical preamble.

The key approximation in our model is that the probability, say \( p \), that a station transmits in a randomly chosen slot time during the \( T_v \) period is constant. Following the footprints of [2], for a \( p \)-persistent MAC protocol it holds that:

\[
E[\text{Idle}_p] = \frac{(1-p)^{n^*}}{1-(1-p)^{n^*}} \cdot \text{SLOT},
\]

\[
E[N_e] = \frac{1-(1-p)^{n^*}}{n^*p(1-p)^{n^*}} - 1.
\]

For reducing the notational complexity during the derivation of the \( E[T_C] \) and \( E[T_S] \) expressions, we define two auxiliary variables, say \( OV_S \) and \( OV_C \), which account for the protocol overheads introduced with the Basic Access method [12] during a successful transmission and a collision, respectively. Let \( t_H = t_{H_{MAC}} + t_{H_{PHY}} \) be the total duration of the frame header transmission, and \( \tau \) be the propagation delay. Considering the packet structure and the MAC protocol operations, it follows that:

\[
\begin{align*}
OV_S &= 2\tau + \text{SIFS} + t_{ACK} + t_H + t_B \cdot H_1, \\
OV_C &= \tau + \text{EIFS} + t_H + t_B \cdot H_1,
\end{align*}
\]

where \( H_1 \) is the sum of the IP and TCP header sizes, expressed in terms of bytes.

Furthermore, considering the Basic Access method, we obtain that:

\[
\begin{align*}
E[T_S] &= \text{DIFS} + m^* \cdot (OV_S + E[P_S] \cdot t_B/8) + (m^* - 1) \cdot \text{SIFS} \quad \text{(I.4a)} \\
E[T_C] &= OV_C + E[P_C] \cdot t_B/8, \quad \text{(I.4b)}
\end{align*}
\]

where \( E[P_S] \) is the average number of payload bits delivered with a successful transmission, while \( E[P_C] \) is the average payload length of the longest packet involved in a collision. To derive \( E[T_S] \) we assumed that each frame in the burst is separately acknowledged, and that two consecutive frame exchanges are separated by a \( \text{SIFS} \) interval (see [7] for a detailed description of this variant of the 802.11 MAC protocol).

- **\( E[P_S] \)** computation

The \( E[P_S] \) value can be computed as follows:

\[
E[P_S] = E[P^{AP}] \cdot Pr\{\text{Succ}^{AP} | \text{Succ}\} + E[P^{STA}] \cdot Pr\{\text{Succ}^{STA} | \text{Succ}\},
\]

where \( E[P^{AP}] \) is the average number of payload bits delivered with an AP’s successful transmission, and \( E[P^{STA}] \) is the average number of payload bits delivered with TCP station’s successful transmission, while \( Pr\{\text{Succ}^{AP} | \text{Succ}\} \)
and \( Pr\{Succ^{STA}|Succ\} \) are the probabilities that observing a successful transmission, this is either an AP’s successful transmission or a TCP station’s successful transmission, respectively. By definition, we have that
\[
E[P_{AP}] = \sum_{l=1}^{l_{max}} l \cdot [F^{AP}(l) - F^{AP}(l-1)] , \quad E[P_{STA}] = \sum_{l=1}^{l_{max}} l \cdot [F^{STA}(l) - F^{STA}(l-1)] ,
\]
where \( l_{max} \) is the MTU value, i.e., the largest packet size (measured in bytes) that can be transmitted over the 802.11 network.

To compute the \( Pr\{Succ^{AP}|Succ\} \) and \( Pr\{Succ^{STA}|Succ\} \) probabilities it is sufficient to note that, once the \( p \)-persistence is assumed for the MAC protocol operations, it follows that the successful transmissions on the channel are independent events. Thus, it immediately follows that:
\[
Pr\{Succ^{AP}|Succ\} = \frac{1}{n^*} , \quad Pr\{Succ^{STA}|Succ\} = \frac{n^* - 1}{n^*} .
\] (1.5)

- \( E[P_C] \) computation

By taking the conditional expectation of \( E[P_C] \) on the number \( n_c \) of colliding packets, \( E[P_C] \) can be rewritten as follows
\[
E[P_C] = \sum_{h=2}^{n^*} E[P_C|n_c = h] \cdot Pr\{n_c = h|Coll\} ,
\] (1.6)
where \( n_c = \frac{n^*}{h} \) is the number of stations transmitting simultaneously with the AP.

The \( E[P_C|n_c = h] \) expression can be further expanded as:
\[
E[P_C|n_c = h] = \sum_{l=1}^{l_{max}} l \cdot Pr\{P_C = l|n_c = h\} ,
\] (1.8)
where \( Pr\{P_C = l|n_c = h\} = Pr\{\max(P_1, P_2, \ldots, P_h) = l\} \), i.e., it is the probability that the maximum payload size among the \( h \) colliding frames is equal to \( l \) bytes. To compute this probability it is useful to distinguish between collisions that involve or not involve the AP. Let \( Pr\{1_A = 1|n_c = h\} \) be the probability that among the \( h \) colliding stations there is the AP, and let \( Pr\{1_A = 0|n_c = h\} \) be the probability that among the \( h \) colliding stations there is not the AP. Then, \( Pr\{P_C = l|n_c = h\} \) writes as follows:
\[
Pr\{P_C = l|n_c = h\} = Pr\{P_C = l|n_c = h, 1_A = 1\} \cdot Pr\{1_A = 1|n_c = h\} + Pr\{P_C = l|n_c = h, 1_A = 0\} \cdot Pr\{1_A = 0|n_c = h\} .
\] (1.9)

To compute the probability that the AP is (is not) transmitting, given that \( h \) stations are colliding, we have to count how many ways exist to select \( h \) colliding nodes among the \( n^* \) active nodes. Standard probabilistic arguments yield
In [13] it was derived that the probability cumulative function of the length of the longest packet payload involved in a collision with \( h \) nodes is the product of the probability cumulative functions of the payload sizes of colliding packets. From this observation, the unknown quantities in (I.9) can be written as

\[
Pr\{P_C = l|n_c = h, 1_A = 1\} = F^{AP}(l)[F^{STA}(l)]^{h-1} - F^{AP}(l - 1)[F^{STA}(l - 1)]^{h-1},
\]

and

\[
Pr\{P_C = l|n_c = h, 1_A = 0\} = [F^{STA}(l)]^h - [F^{STA}(l - 1)]^h.
\]

By substituting (I.11), (I.12) in equation (I.9) we finally obtain \( E[P_C|n_c = h] \).

To conclude this appendix we specify the payload-size distributions for the traffic sources described in Section III-B. Specifically, for the equivalent saturated AP introduced in Section III-B, the probability distribution of the payload size of its frame transmissions can be expressed as:

\[
F^{AP}(l) = \begin{cases} 
0 & l < 0 \\
\frac{n_d}{n} & 0 \leq l < l_T \\
1 & l \geq l_T
\end{cases}
\]

This distribution follows from the observation that the real AP generates TCP data segments for the \( n_d \) TCP downlink flows and TCP acknowledgments for the \( n_u \) TCP uplink flows. In this study we assume that the TCP data segments have a fixed-size data content equal to \( l_T \) bits and that the TCP ACK packets have no useful payload. Similarly, for each equivalent saturated TCP station the payload-size probability distribution can be expressed as

\[
F^{STA}(l) = \begin{cases} 
0 & l < 0 \\
\frac{n_d}{n} & 0 \leq l < l_T \\
1 & l \geq l_T
\end{cases}
\]

This distribution follows from the observation that the equivalent TCP stations generate TCP ACK packets for the \( n_d \) TCP downlink flows and TCP data segments for the \( n_u \) TCP uplink flows.

By means of the distributions provided with (I.13) and (I.14), we can compute the \( E[P_C] \) value as defined in (I.4b).

Remark: note that in all the above derivations we have implicitly assumed that \( n^* \) 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 \( n^* \) parameter the smallest integer number greater than or equal to the exact value (i.e., \( n^* = \text{ceil}\{1+E[a|Succ]\} \)). Obviously, more sophisticated solutions can be devised, such as to assume that the \( n^* \) parameter follows some general discrete distribution law, whose mean is \( (1 + E[a|Succ]) \). However, our results justify the use of a simpler approach.