

A Simple Model for the Window Size Evolution of TCP Coupled with MAC and PHY Layers

George Papageorgiou, John S. Baras

Institute for Systems Research, University of Maryland, College Park, MD 20742

Email: {gpapag, baras}@umd.edu

Abstract—In this paper the interaction between the AIMD algorithm of TCP and the random access channel is investigated. In particular, we examine the effects of the MAC and the physical layer of the backward channel on the window size evolution of TCP. The problem of coupling the window size evolution of TCP with a random access channel is addressed using point processes and the theory of martingales.

I. INTRODUCTION

Due to the great importance of TCP in the Internet, various models of it have been developed. These models try to capture the operation of the main mechanisms of TCP and to give insights on how these mechanisms can be improved in order to fine-tune TCP under various networking environments.

TCP is very popular in wired networks and is also used in the first generation of many wireless networks. Thus, it is important to investigate its performance over a wireless communication environment. In a wireless environment the characteristics of communication are quite different compared to those of a wired environment, yielding the existing layering approach of protocols inefficient. It is believed that a closer interconnection between various layers in the protocol stack of a mobile node would allow for a better utilization of the wireless network. One aspect of this cross-layer integration should include the flow-control, which is an end-to-end function, and access control of the shared wireless channel, an operation which is local to each mobile node. The integration of flow-control with Medium Access Control (MAC) would prevent a data source in a mobile node from overloading the network and hence, decrease its overall performance.

In this paper, we consider the window-based mechanism of flow-control of TCP on top of an Aloha-based MAC protocol. All nodes in the network are in hearing distance from each other, thus the hidden terminal problem does not exist. The characteristics of the physical channel are captured through the use of a simple two-state Markov process. Aloha is chosen as the MAC protocol since it provides a very simple channel access mechanism and also because it captures in a simple way the random waiting time before retransmission present in many other random access protocols. In this paper we attempt to develop a simple TCP model for a wireless communications

environment that captures the behavior of TCP in such an environment. To this end, we consider a single persistent TCP connection over one wireless hop, i.e. the TCP sender and the TCP receiver are one hop away. Because of this, no buffering is performed in any intermediate node, and thus, the round-trip time (RTT) consists mainly of the delay incurred by the MAC in its effort to successfully transmit the packet. Since we are interested in examining the effect of timeouts due to MAC and the physical layer, we assume the forward channel (i.e. the channel from the TCP sender to the receiver) to be ideal, in the sense that there are no packet losses. Thus, there are no duplicate acknowledgements (ACKs) received at the TCP sender. The same situation, i.e. detection of packet losses through timeouts rather than duplicate acknowledgements, arises in the case where the bandwidth-delay product is small [2]. Timeout events are produced because of ACK losses in the backward channel (from the TCP receiver to the TCP sender). In the backward channel the MAC layer introduces delays and thus increases the RTTs, while the physical layer is responsible for ACK losses.

II. MODEL DESCRIPTION

The TCP flow-control mechanism is modeled according to the additive increase multiplicative decrease (AIMD) paradigm. Two quantities are defined, the TCP window size W and the slow-start threshold S . The window size W in the sender changes dynamically based on the reception of ACKs. During normal operation of the protocol, where no packet losses occur, the arrival of an ACK causes the increase of the window size. An ACK is sent from the receiver to the sender every time a TCP packet is correctly received by the receiver. An ACK is cumulative in the sense that by sending an acknowledgement packet, the receiver acknowledges that all the packets sent by the sender with sequence number less than the number indicated in the ACK packet were correctly received. In the case where a packet from a series of packets is lost, the receiver produces ACKs for every packet that is correctly received after the lost packet. Each of these ACKs acknowledges that packets before the lost one are correctly received but there is still one packet missing.

If the current window size is less than the slow-start threshold, then TCP is in the slow-start phase and $W_{new} = W_{cur} + 1$ each time an ACK is received, otherwise TCP is in the congestion avoidance phase and $W_{new} = W_{cur} + 1$ per RTT. The latter suggests that the increase in the window size is

The material is based upon research supported by the National Aeronautics and Space Administration under Cooperative Agreements No. NCC8-235 and NAG3-2844, and by Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program Cooperative Agreement DAAD19-01-2-001.

according to $W_{new} = W_{cur} + 1/W_{cur}$ for every received ACK. TCP assumes a packet has been lost either by receiving three duplicate ACKs for a specific packet, or by a timeout. In both cases it is assumed by TCP that the packet was lost because of congestion in the network. In the first case though, since the network is able to deliver packets to the final destination the congestion cannot be severe, so TCP reacts less aggressively. Particularly, the slow-start threshold is updated by taking half the value of the current window size, $S_{new} = W_{cur}/2$, and $W_{new} = S_{new}$. Thus, after the arrival of three duplicate ACKs, TCP enters the congestion avoidance phase. The second case implies a severe congestion in the network, so TCP reacts aggressively by setting $S_{new} = W_{cur}/2$, and $W_{new} = 1$. This reaction causes TCP to enter the slow-start phase.

The packet loss model associated with the wireless channel can be described by a continuous time Markov chain with two states [1]. One state corresponds to the channel being “good”, i.e. packets are not lost w.p. 1, and the other state corresponds to the channel being “bad” having as an effect packet losses w.p. 1. The transition rate from the “bad” to the “good” state is λ_{bg} and the transition rate from the “good” to the “bad” state is λ_{gb} .

The MAC layer model assumed in this work is based on the pure Aloha mechanism [3] (and the references therein). In pure Aloha, if a packet transmission overlaps at all with that of another packet, then the transmission is unsuccessful for all the packets that participated in the collision. A packet that failed to be transmitted successfully is retransmitted after a random delay. It is assumed that this delay is a random variable following the exponential distribution with mean $1/\lambda_{ret}$ and it is independent of any previous delays. It is also assumed that a feedback is immediately available to the nodes informing them about the successful or unsuccessful transmission of a packet.

If the new arrivals to the system are Poisson with rate λ , and the number of nodes that have packets that participated in a collision (backlog) is n , then the initiation times of attempted transmissions follow a time-varying Poisson distribution with rate $G(n) = \lambda + n\lambda_{ret}$. In this setting the probability of a successful transmission of a packet is,

$$p_{mac} = e^{-2G(n)T_p} \quad (1)$$

where $T_p = L/C$ is the transmission time of the packet of constant length L bits over the wireless channel of capacity C bps.

The poor performance of TCP over wireless stems from the fact that TCP cannot distinguish between packet losses due to congestion and packet losses due to poor channel quality. As a result treats all packet losses as if they happen because of the congestion in the network. The TCP sender sets a timeout period for each packet sent to the network. If no ACK for this particular packet is received during this period, the packet is considered lost. Each packet is delayed in the MAC layer because of collisions and probably is lost in the physical layer because of the poor channel quality.

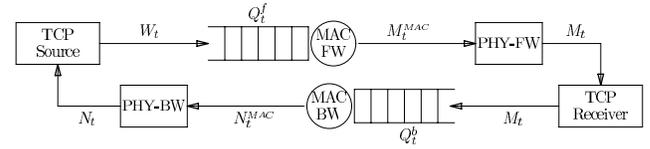


Fig. 1. System model.

In this paper we attempt to develop a simple model for the window size evolution of TCP which incorporates the dependencies from the MAC and the physical layers.

III. ANALYSIS

In this section we give the mathematical models that describe the operation of each of the layers involved in our analysis. In the following we assume there exists a complete probability space (Ω, \mathcal{F}, P) . A representation of the system is given in Fig. 1.

A. Physical Layer

We elaborate on the model for the Physical layer that was briefly introduced in Section II. We define the continuous time Markov chain $P = (P_t)_{t \geq 0}$ with a state space $\mathcal{P} = \{0, 1\}$. When $P_t = 0$, it means the channel is in the “bad” state and the transmitted packet is dropped w.p. 1, when $P_t = 1$, it means the channel is in the “good” state and the transmission of the packet is successful w.p. 1. As was mentioned in Section II, the transition rates from “bad” to “good” and from “good” to “bad” are λ_{bg} and λ_{gb} , respectively. Then, the transition probabilities for the chain in a small time interval $h > 0$ can be given by:

$$\begin{aligned} p_{00}(h) &= 1 - \lambda_{bg}h + o(h), & p_{01}(h) &= \lambda_{bg}h + o(h) \\ p_{10}(h) &= \lambda_{gb}h + o(h), & p_{11}(h) &= 1 - \lambda_{gb}h + o(h) \end{aligned}$$

where $o(h)$ is such that $\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$.

In [6, Appendix A] the probability $p_0(t)$ the chain is in the “bad” state at time t is computed to be:

$$p_0(t) = \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} + \left(p_0(0) - \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} \right) e^{-(\lambda_{bg} + \lambda_{gb})t} \quad (2)$$

for $t \geq 0$ and some initial probabilities $p_0(0)$ and $p_1(0)$ for the chain to be in the “bad” and the “good” state respectively, at time $t = 0$.

Since $p_0(t) + p_1(t) = 1$ for all $t \geq 0$, we also have:

$$p_1(t) = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} + \left(p_1(0) - \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} \right) e^{-(\lambda_{bg} + \lambda_{gb})t} \quad (3)$$

for $t \geq 0$. The stationary distribution for the Markov chain corresponds to the case where $t \uparrow \infty$ in (2) and (3). The stationary probabilities π_b , π_g of being in the “bad” and the “good” states respectively, are computed:

$$\pi_b = \lim_{t \rightarrow \infty} p_0(t) = \frac{\lambda_{gb}}{\lambda_{bg} + \lambda_{gb}} \quad (4a)$$

$$\pi_g = \lim_{t \rightarrow \infty} p_1(t) = \frac{\lambda_{bg}}{\lambda_{bg} + \lambda_{gb}} \quad (4b)$$

In [6, Appendix A] it is also proven that the waiting time $T_i, i \in \mathcal{P}$ in each state is exponentially distributed and more specifically,

$$P\{T_0 \geq t\} = e^{-\lambda_{bg}t}, \quad t \geq 0 \quad (5a)$$

$$P\{T_1 \geq t\} = e^{-\lambda_{gb}t}, \quad t \geq 0 \quad (5b)$$

As was mentioned in Section I we focus on the effect of the MAC and the physical layer on the timeout mechanism of TCP. Thus, we assume the physical channel (PHY-FW) in the forward direction to be ideal. This means that M_t and M_t^{MAC} in Fig. 1, are indistinguishable and there are no duplicate ACKs produced at the TCP receiver. The effect of the physical layer (PHY-BW) in the backward direction is described by (4) and (5). In particular, the process $N = (N_t)_{t \geq 0}$ is a thinned version of the point process $N^{MAC} = (N_t^{MAC})_{t \geq 0}$ and this thinning is done with the stationary probability π_g the channel is in the good state, given by (4b).

B. MAC Layer

In this section we give a more detailed description of the MAC layer model that we use in our analysis. In this work we consider the pure Aloha protocol. Each packet i is successfully transmitted (i.e. without any collisions at the MAC layer) with probability p_{mac} that is given by (1) and the transmission time in this case will be constant and equal to $T_p = L/C$ as was mentioned in Section II. A collision happens with probability $1 - p_{mac}$ and the packet has to wait a random time that is exponentially distributed with mean $1/\lambda_{ret}$. At the end of this time period another transmission is attempted. If there is another collision the packet has to wait again for some time which is exponentially distributed with mean $1/\lambda_{ret}$ and is independent of any previous waiting periods. Since $p_{mac} > 0$, the packet will eventually be transmitted successfully.

Since each packet transmission happens independently of any transmissions of previous packets, if we define D_i^{MAC} to be the service time (the time from the moment the packet goes to the head of the queue till it is successfully transmitted) of packet i in the MAC layer, then the random variables $\{D_i^{MAC}, i = 1, 2, \dots\}$ form an i.i.d. sequence represented by the generic random variable D^{MAC} . We know that it is always true $D^{MAC} \geq T_p$. In particular,

$$D^{MAC} = T_p + \sum_{j=1}^K X_j$$

where $\{X_j, j = 1, 2, \dots, K\}$ are i.i.d. exponentially distributed random variables with mean $1/\lambda_{ret}$, and K is a geometrically distributed random variable with parameter p_{mac} , such that

$$P\{K = k\} = p_{mac}(1 - p_{mac})^k, \quad k = 0, 1, 2, \dots$$

In [6, Appendix A] it is shown that the characteristic function of the random variable D^{MAC} is given by

$$E\left[e^{isD^{MAC}}\right] = p_{mac} e^{isT_p} + (1 - p_{mac}) \frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac} - is} e^{isT_p} \quad (6)$$

From (6) we immediately get the p.d.f. of the random variable D^{MAC} is

$$f_{D^{MAC}}(t) = p_{mac}\delta(t - T_p) + p_{mac}(1 - p_{mac})\lambda_{ret}e^{-\lambda_{ret}p_{mac}(t - T_p)}u_0(t - T_p) \quad (7)$$

for $t \geq 0$, where $u_0(\cdot)$ is the Heaviside function:

$$u_0 = \begin{cases} 0, & t \leq T_p \\ 1, & t > T_p \end{cases}$$

The times between successful packet transmissions at the MAC layer (assuming there are always packets to be transmitted) are independent and distributed according to (7), forming a renewal process. If we define the corresponding point process to be $\{T_n^{MAC}, n = 0, 1, \dots\}$, with $T_0^{MAC} = 0$ P -a.s., then

$$T_n^{MAC} = D_1^{MAC} + \dots + D_n^{MAC} \quad (8)$$

where $D_1^{MAC}, \dots, D_n^{MAC}$ are i.i.d. random variables with p.d.f. given by (7) and

$$D_i^{MAC} = T_p + \sum_{j=1}^{K_i} X_j, \quad i = 1, 2, \dots, n$$

where K_i is geometrically distributed with parameter p_{mac} and X_j is exponentially distributed with parameter λ_{ret} . Then,

$$\begin{aligned} T_n^{MAC} &= nT_p + \sum_{j=1}^{K_1} X_j + \dots + \sum_{j=1}^{K_n} X_j \\ &= nT_p + \sum_{j=1}^K X_j \end{aligned} \quad (9)$$

where K is the sum of n i.i.d. geometrically distributed random variables with parameter p_{mac} . It is shown in [6] that the random variable K has a negative binomial distribution with parameters n and p_{mac} ,

$$P\{K = k\} = \binom{n+k-1}{k} p_{mac}^n (1 - p_{mac})^k$$

for $k = 0, 1, \dots$. Regarding $\sum_{j=1}^K X_j$:

$$\begin{aligned} E\left[e^{is \sum_{j=1}^K X_j}\right] &= \sum_{k=0}^{+\infty} (E[e^{isX_1}])^k \binom{n+k-1}{k} p_{mac}^n (1 - p_{mac})^k \\ &= \left(\frac{\lambda_{ret}p_{mac} - ip_{mac}s}{\lambda_{ret}p_{mac} - is}\right)^n \\ &= p_{mac}^n \sum_{k=0}^n \binom{n}{k} \left(\frac{1 - p_{mac}}{p_{mac}}\right)^k \left(\frac{\lambda_{ret}p_{mac}}{\lambda_{ret}p_{mac} - is}\right)^k \end{aligned}$$

which means that

$$\begin{aligned} f_{\sum_{j=1}^K X_j}(t) &= p_{mac}^n \delta(t) \\ &+ p_{mac}^n \sum_{k=1}^n \binom{n}{k} \left(\frac{1 - p_{mac}}{p_{mac}}\right)^k \\ &\times \frac{(\lambda_{ret}p_{mac})^k}{(k-1)!} t^{k-1} e^{-\lambda_{ret}p_{mac}t} \end{aligned} \quad (10)$$

for $t \geq 0$. From (9) and (10), the p.d.f. for T_n^{MAC} is computed and

$$\begin{aligned} f_{T_n^{MAC}}(t) &= f_{\sum_{j=1}^K X_j}(t - nT_p) \\ &= p_{mac}^n \delta(t - nT_p) \\ &\quad + p_{mac}^n \sum_{k=1}^n \binom{n}{k} \left(\frac{1 - p_{mac}}{p_{mac}} \right)^k \\ &\quad \times \frac{(\lambda_{ret} p_{mac})^k}{(k-1)!} (t - nT_p)^{k-1} e^{-\lambda_{ret} p_{mac} (t - nT_p)} \end{aligned} \quad (11)$$

for $t \geq nT_p$.

We define the counting process $N^{MAC} = (N_t^{MAC})_{t \geq 0}$ that corresponds to the point process $\{T_n^{MAC}, n = 0, 1, \dots\}$,

$$N_t^{MAC} = \sum_{i=1}^{\infty} \mathbf{1}[T_n^{MAC} \leq t], \quad t \geq 0 \quad (12)$$

Although the sequence $\{T_{n+1}^{MAC} - T_n^{MAC}, n = 0, 1, \dots\}$ is an i.i.d. sequence defining a renewal process, the counting process N^{MAC} does not have stationary and independent increments. We define the history of the N^{MAC} process as the right continuous filtration $\mathcal{F}^{MAC} = (\mathcal{F}_t^{N^{MAC}})_{t \geq 0}$, such that,

$$\mathcal{F}_t^{N^{MAC}} = \sigma\{N_s^{MAC}, s \leq t\} = \sigma\{T_{N_s^{MAC}}^{MAC}, s \leq t\}$$

To compute the $\mathcal{F}_t^{N^{MAC}}$ -compensator \mathcal{N}_t of the N^{MAC} process, we define the conditional distribution functions:

$$\begin{aligned} F_1(t) &= P\{T_1^{MAC} \leq t\} \\ F_i(t) &= P\{T_i^{MAC} \leq t \mid T_{i-1}^{MAC}, \dots, T_1^{MAC}\}, \quad i \geq 2 \end{aligned}$$

From (8) we have that

$$T_1^{MAC} = D_1^{MAC}, \quad P\text{-a.s.}$$

and using (7) we compute the conditional distribution $F_1(\cdot)$ and the corresponding p.d.f. $f_1(\cdot)$:

$$\begin{aligned} f_1(t) &= p_{mac} \delta(t - T_p) \\ &\quad + p_{mac}(1 - p_{mac}) \lambda_{ret} e^{-\lambda_{ret} p_{mac} \cdot (t - T_p)} u_0(t - T_p) \end{aligned}$$

and

$$F_1(t) = \begin{cases} 0, & t < T_p \\ p_{mac} + (1 - p_{mac}) \left(1 - e^{-\lambda_{ret} p_{mac} (t - T_p)}\right), & t \geq T_p \end{cases}$$

From (8) we notice that

$$T_i^{MAC} = T_{i-1}^{MAC} + D_i^{MAC}, \quad i \geq 2$$

thus,

$$F_i(t) = P\{D_i^{MAC} \leq t - T_{i-1}^{MAC} \mid T_{i-1}^{MAC}\}, \quad i \geq 2$$

Using (7) we get the conditional distribution $F_i(\cdot)$ and the corresponding p.d.f. $f_i(\cdot)$, $i \geq 2$:

$$\begin{aligned} f_i(t) &= p_{mac} \delta(t - T_{i-1}^{MAC} - T_p) \\ &\quad + p_{mac}(1 - p_{mac}) \lambda_{ret} e^{-\lambda_{ret} p_{mac} \cdot (t - T_{i-1}^{MAC} - T_p)} \\ &\quad \times u_0(t - T_{i-1}^{MAC} - T_p) \end{aligned}$$

$$F_i(t) = \begin{cases} 0, & t < T_{i-1}^{MAC} + T_p \\ p_{mac} + (1 - p_{mac}) \\ \quad \times e^{-\lambda_{ret} p_{mac} (t - T_{i-1}^{MAC} - T_p)}, & t \geq T_{i-1}^{MAC} + T_p \end{cases}$$

We proceed by defining

$$\begin{aligned} \mathcal{N}_t^{(i)} &= \int_0^{t \wedge T_i^{MAC}} \frac{dF_i(u)}{1 - F_i(u^-)} \\ &= \int_0^{t \wedge T_i^{MAC}} \frac{f_i(u)}{1 - F_i(u^-)} du, \quad i \geq 1 \end{aligned}$$

Then,

$$\mathcal{N}_t^{(1)} = \begin{cases} 0, & 0 \leq t < T_p \\ \lambda_{ret} p_{mac} (t \wedge T_1 - T_p), & T_p \leq t \end{cases}$$

and

$$\mathcal{N}_t^{(i)} = \begin{cases} 0, & 0 \leq t < T_{i-1}^{MAC} + T_p \\ \lambda_{ret} p_{mac} \\ \quad \times (t \wedge T_i - T_{i-1}^{MAC} - T_p), & T_{i-1}^{MAC} + T_p \leq t \end{cases}$$

From [4, T7 Theorem, p.61] and [5, Theorem 18.2, p.270] the $\mathcal{F}_t^{N^{MAC}}$ -compensator \mathcal{N}_t of the N^{MAC} process is given by

$$\mathcal{N}_t = \sum_{i \geq 1} \mathcal{N}_t^{(i)}$$

and using $\mathcal{N}_t^{(i)}$, $i \geq 2$ computed above, we have

$$\mathcal{N}_t = \begin{cases} \lambda_{ret} p_{mac} (T_i^{MAC} - iT_p), & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \lambda_{ret} p_{mac} t - \\ \quad \times (i+1) \lambda_{ret} p_{mac} T_p, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \quad (13)$$

The $\mathcal{F}_t^{N^{MAC}}$ -intensity η_t^{MAC} can be computed directly from (13) to be:

$$\eta_t^{MAC} = \begin{cases} 0, & T_i^{MAC} \leq t < T_i^{MAC} + T_p \\ \lambda_{ret} p_{mac}, & T_i^{MAC} + T_p \leq t < T_{i+1}^{MAC} \end{cases} \quad (14)$$

C. Transport Layer

To describe the evolution of the window size, two stochastic processes $W = (W_t)_{t \geq 0}$ and $H = (H_t)_{t \geq 0}$ are defined, where W_t is the window size of the TCP flow, and H_t is the corresponding slow-start threshold at time t .

Underlying Point Processes: Given the description in Section II, there exist two underlying strictly increasing sequences of random variables representing two point processes:

- for the arrival of ACKs $\{T_n, n = 0, 1, \dots\}$ with $T_0 = 0$, P -a.s. and intensity $\lambda_t > 0$, and
- for the timeout events $\{S_n, n = 0, 1, \dots\}$ with $S_0 = 0$, P -a.s. and intensity $\mu_t > 0$.

The point process $\{T_n, n = 0, 1, \dots\}$ represents the arrival of ACKs at the TCP sender, and is closely related to the MAC and the physical layer. With the assumption that there are always ACKs waiting transmission at the MAC layer at the TCP receiver side, it was shown in Section III-B that the

successful (i.e. without collisions) transmissions of ACKs at the MAC layer form a renewal process.

Those ACKs that survived collisions at the MAC layer are subject to the quality of the physical layer. Thus, each of these ACKs is successfully received at the TCP sender with probability π_g that is given by (4b) and this happens independently of the operation of the MAC layer (thinning of the point process).

If $\mathcal{F}^N = (\mathcal{F}_t^N)_{t \geq 0}$ is the right continuous filtration that represents the history of the point process $\{T_n, n = 0, 1, \dots\}$, then the \mathcal{F}^N -intensity λ_t of the process is

$$\lambda_t = \pi_g \eta_t^{MAC} \quad (15)$$

where π_g is given by (4b) and η_t^{MAC} is given by (14).

The point process $\{S_n, n = 0, 1, \dots\}$ represents the time-out events. For each packet sent to the network, TCP expects an ACK back from the receiver acknowledging the receipt of the packet. In the case of poor channel quality such an ACK may be lost. If the TCP sender does not receive the ACK in certain amount of time it will assume the packet was not properly received by the receiver and will retransmit it, minimizing at the same time its window size and in effect the throughput of the connection. In a wireless network though an ACK may experience delays because of the MAC and the collisions that take place when accessing the channel. Thus, the TCP sender should not be anxious declaring a timeout and in effect minimizing the sending rate to the network. On the other hand, the more the TCP sender is waiting for the arrival of an ACK, the more the connection remains idle resulting in performance degradation.

The Slow-Start Threshold Process $H = (H_t)_{t \geq 0}$: Based on the point processes defined above, the stochastic process H that represents the slow-start threshold in TCP is given by:

$$H_0 = h, \quad P\text{-a.s.} \quad (16)$$

$$H_t = h + \sum_{n=1}^{\infty} \mathbf{1}[S_n \leq t] \Delta H_{S_n}, \quad t > 0$$

where h is given. The sample paths of H defined by (16) are piecewise constant and right continuous with left limits (càdlàg process) and non-increasing. The magnitude of each jump at the points of the process $\{S_n, n = 1, 2, \dots\}$ is given by

$$\begin{aligned} \Delta H_{S_n} &= H_{S_n} - H_{S_n^-} \\ &= H_{S_n} - H_{S_{n-1}} \\ &= \frac{W_{S_n}^-}{2} - \frac{W_{S_{n-1}}^-}{2} \\ &= \frac{1}{2} (W_{S_n}^- - W_{S_{n-1}}^-) \end{aligned}$$

for $n = 2, 3, \dots$, and

$$\begin{aligned} \Delta H_{S_1} &= H_{S_1} - H_{S_1^-} \\ &= H_{S_1} - H_{S_0} \\ &= \frac{W_{S_1}^-}{2} - h \end{aligned}$$

and it is zero for all the other time instances. It should be noticed that according to the description in Section II it is always true that $\Delta H_n < 0$ for all n .

The Window Size Process $W = (W_t)_{t \geq 0}$: As was described in Section II, the window size evolution is driven by the two point processes $\{T_n, n = 0, 1, \dots\}$, and $\{S_n, n = 0, 1, \dots\}$.

During the slow-start phase the window size is increased by 1 for every received acknowledgement. Define the counting process $N = (N_t)_{t \geq 0}$ that is associated with the point process $\{T_n, n = 0, 1, \dots\}$ and counts the received acknowledgements:

$$N_t = \sum_{i=1}^{\infty} \mathbf{1}[T_n \leq t], \quad t \geq 0$$

The process N is an \mathcal{F}^N -submartingale and from the Doob-Meyer decomposition we have

$$dN_t = \lambda_t dt + dX_t$$

where X_t is an \mathcal{F}_t^N -martingale with $X_0 = 0$, P -a.s..

Then, in the slow-start phase the window size evolves according to

$$W_t = W_0 + N_t, \quad t \geq 0 \quad (17)$$

where W_0 is given. As was described in Section II the evolution of the window size in congestion avoidance phase is more conservative compared to the case of the slow-start phase. Based on that description the window size evolution in the congestion avoidance phase is described by

$$dW_t = \frac{1}{W_t} dN_t, \quad t \geq 0 \quad (18)$$

IV. CROSS-LAYER INTEGRATION OF TCP, MAC AND PHY

The continuous time evolution of the window size is characterized by (17) and (18) for slow-start and congestion avoidance respectively. It starts in the slow-start phase and if there are no timeouts it switches to the congestion avoidance phase whenever the window size W_t becomes larger than the slow-start threshold H_t . Whenever the window size W_t reaches its maximum allowable value W_{max} , it remains to this value. In any case, whenever a timeout occurs, TCP switches to slow-start and sets the window size to its minimum value W_0 :

$$W_{S_n} = W_0, \quad n = 1, 2, \dots$$

and the window size evolves according to (17). To summarize, the window size evolution is described by:

$$dW_t = \begin{cases} \lambda_t dt + dX_t, & W_t < H_t \\ \frac{1}{W_t} \lambda_t dt + \frac{1}{W_t} dX_t, & W_t \geq H_t \end{cases}$$

$$W_{S_n} = W_0, \quad n = 1, 2, \dots \quad (19)$$

$$W_0 \leq W_t \leq W_{max}, \quad t \geq 0$$

Note that both processes $W = (W_t)_{t \geq 0}$ and $H = (H_t)_{t \geq 0}$ are fully observable by the TCP sender (controller).

V. CONCLUSION

The issue of TCP performance optimization in wireless environments is important because of the increased popularity of the wireless networks and their applications, e.g. web applications on cellular devices. The inherent difficulty of TCP to deal with packet drops due to channel quality fluctuations deem any efforts to alleviate these problems crucial for the success of wireless networking and wireless Internet. This paper is a first step towards the development of optimization techniques that can improve the performance of TCP over wireless channels.

The paper provides a simple dynamic equation that describes the evolution of the window size of TCP, parametrized by quantities that describe the operation of the physical and the MAC layers in a random access wireless network. Such a dynamic equation can later be used to maximize TCP throughput (which is a function of the window size).

REFERENCES

- [1] A. A. Abouzeid, S. Roy, and M. Azizoglu. Comprehensive performance analysis of a TCP session over a wireless fading link with queueing. *IEEE Trans. Wireless Commun.*, 2(2):344–356, Mar. 2003.
- [2] K. E. Avrachenkov, A. A. Kherani, N. O. Vilchevsky, and V. S. Zaborovski. Optimal tuning of the TCP retransmission timeout for small-BDP lossy wireless networks. In *Proceedings of the 6th ITC Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems*, Antwerp, Belgium, Sept. 2004.
- [3] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, Englewood Cliffs, NJ, second edition, 1992.
- [4] P. Brémaud. *Point Processes and Queues, Martingale Dynamics*. Springer-Verlag, New York, 1981.
- [5] R. S. Liptser and A. N. Shiryaev. *Statistics of Random Processes, II. Applications*. Springer-Verlag, second edition, 2001.
- [6] G. Papageorgiou and J. S. Baras. Simple model for the window size evolution of tcp coupled with mac and phy layers. Technical report, <http://www.isr.umd.edu/~gpapag/tr-ciss-2009.pdf>, Feb. 2009.