On the Processor Sharing Properties of File Transfers in a WLAN Testbed

Gerard Hoekstra[1,2] and Rob van der Mei[1,3]

[1]Centre for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands
[2]Innovation Research & Technology, Thales Nederland B.V., Huizen, The Netherlands
[3]VU University Amsterdam, Department of Mathematics, The Netherlands
E-mail: {hoekstra, mei}@cwi.nl

***Abstract*—802.11-based WLAN deployments have become a commodity to provide today's wireless Internet access. In this paper, we conduct a practical study on the performance of FTP file transfers over real WLAN equipment. To this end, we propose a new analytic model that translates the highly complex dynamics of the FTP/TCP/IP/MAC-stack, and their interactions, into a single parameter, which will be called the *effective load*. The effective load is used to describe the flow-level behavior of FTP-based file transfers over WLANs without admission control as a Processor-Sharing (PS)-model. Next, despite the fact that PS models are heavily used in modeling flow-level performance in WLAN networks, an extensive validation of such models with real equipment has not been conducted. Motivated by this, we validate in the present paper our analytic model by comparing the model-based response times against the outcomes obtained from a testbed environment. The results show (a) that the obtained mean download times are fairly insensitive to the file-size distribution, as suggested by the PS-model, and (b) that the model leads to accurate predictions over a broad range of parameters combinations, including different file-size distributions and light- and heavy-load scenarios.**

***Keywords*—Processor-Sharing queues, Wireless LAN, Performance Modeling.**

I. Introduction

Wireless Local Area Network (WLAN) interfaces are available on a wide range of contemporary communication devices to provide users with mobile access, often to the Internet. For practical deployments, there is a need for accurate, explicit, though simple models that can predict the performance of WLANs under particular load conditions.

In the literature, WLAN performance models have appeared that can be categorized in packet and flow-based models. Initial studies concentrated on packet-level models. The seminal work by Bianchi [3] is the most prominent contribution to analyze the packet-level performance in a saturated WLAN, using fixed-point iteration. With the ongoing efforts in IEEE 802.11-standardization, new physical and MAC-layer functionality has been defined. As a result, different parametrization of Bianchi's model appeared [13], [5], [9] to capture the saturation throughput for the higher data-rate WLAN standards. At a later stage, Processor Sharing (PS)-based flow-level models have appeared that incorporated the packet-level details to obtain the performance of non-persistent data flows with [12], [10] or without [8] using the TCP protocol. PS-models are assumed applicable to a variety of communication networks, including CDMA 1xEV-DO, WLAN, UMTS-HSDPA and ADSL (see [4], [15], [7], [8], [2]).

Many studies, however, have validated the applicability of the PS-model to a WLAN network based on simulations [8], [12], [13], [10], rather than experiments with real network equipment. Despite the fact that these models generally lead to accurate performance predictions, they do not lead to simple explicit expressions for the performance of WLANs. Furthermore, simplifying assumptions have been made in these contributions that do not occur in practice, for example the use of a flow admission control mechanism, WLAN channel reservation methods (RTS/CTS), and allowing stations to download only one file simultaneously. These assumptions do not meet the conditions in practical deployments where stations may have multiple downloads in progress in a network without any form of admission control operating in basic access mode without the use of channel reservations. However, previous work on flow-level performance [12], [13], [10], [8] meets at most partially these constraints. This has motivated us to develop a new, simple, yet accurate model that *does* take into account *each* of these constraints imposed by practical deployments. A partial and preliminary version of this model appeared in [7], followed by detailled model in [6] that was thoroughly validated by OPNET simulations. In the present paper, the latter model is further extended and validated against testbed experiments.

In this paper, our main interest is in the performance at the application layer that depends on the underlying protocol dynamics, including that of the TCP and the 802.11 MAC. In [12], [13] it is observed that the combined dynamics of the MAC and TCP-layer simplify the analytic modeling when compared to separate and generally complex MAC [3] and TCP-layer models.

The contribution of this paper threefold. First, we present a detailed analytic model, based on the preliminary model in [7], that is used to obtain the *effective load* of the WLAN network. Second, the effective load is used to describe the flow-level performance of file transfers over WLANs by an

$M/G/1$ Processor Sharing (PS) model, where the load is taken equal to the effective load. In this way, the combined dynamics and protocol overhead of the 802.11 MAC, IP, TCP and application layer are accounted for by the PS-model. Third, we validate our model in a testbed environment for different file-size distributions under light and-heavy load scenarios.

II. Analytic model

The modeling of throughput performance of WLAN has received much attention from the research community of which the vast majority on the Distributed Coordination Function (DCF), specified in the IEEE 802.11 standard [1]. The most prominent analytic models are based on the Markov chain approach of Bianchi [3]. Where Bianchi's parametrization of the model concerned FHSS, others [13], [5], [9] have applied the PHY/MAC parameters of the IEEE 802.11b-,g- and a-standard, respectively. For a detailed description of the PHY/MAC level aspects we refer to these contributions.

As WLAN networks all rely on the same MAC protocol basis, the MAC aspects of our model are based on the IEEE 802.11b standard amendment for its wide availability and lower system requirements involved in high-load experimental validation.

For our analytic model we consider a population of users that download files from an FTP server that is located close to the AP, and make the following assumptions:

- File download requests (passive) arrive from the FTP users according to a Poisson process at the FTP server.
- In the WLAN network there is no admission control mechanism present to limit the number of downloads per station or in total.
- The FTP users are logged in once and maintain their FTP session.
- The FTP users are positioned statically and close to the Access Point (AP).
- All application traffic is carried by TCP that uses delayed acknowledgments for every two data segments.
- The packet loss and delay in the network is negligibly small.
- The contention on the medium is low (all stations draw their backoff time from the minimum interval Cw_{min}).

The analytic model is an extended version of the one presented in [7] in that the FTP and TCP overhead is modeled in greater detail to match the behavior of a real FTP session more realistically. In the sequel, we assume that the reader is familiar with FTP, TCP and 802.11 protocol behavior and compose our model along the lines of those in [7] by first considering the packet level details, followed by a flow-level model.

A. Packet-level Model

For the IEEE 802.11b standard, the following equations apply to the time spent on transmitting the same TCP *data* segment, $T_d(x)$, and on the associated WLAN *acknowledgment*, T_a:

$$T_d(x) = phy + \frac{mac + X_{tcp/ip} + x}{R}, \tag{1}$$

with x representing the TCP segment size in bits, $X_{tcp/ip}$ the TCP/IP overhead bits, mac the 802.11 MAC-layer overhead bits, and R is the WLAN transmission rate (in bits-per-second) used for data units. Additionally, a fixed amount of time, phy, is consumed on the medium by transmitting the overhead associated to the Physical Layer Convergence Protocol (PLCP). We refer to Table I for the 802.11b parameters used for our model.

TABLE I
Selected IEEE 802.11b Parameters

Parameter	Value	Parameter	Value
R_c	10^6 bps	R	11×10^6 bps
mac	224 bits	$sifs$	10μs
τ	20μs	$difs$	50μs
phy	96μs	$eifs$	268μs
mac_ack	112μs	Cw_{min}	31 slots

After the successful transmission of a data segment, the destination WLAN station will reply by transmitting a MAC *acknowledgment*. This occupies the medium for T_a seconds:

$$T_a = phy + \frac{ack}{R_c}, \tag{2}$$

where instead of the mac overhead bits a smaller ack overhead is used. R_c represents the WLAN transmission rate (in bps) for the WLAN acknowledgments.

When transmitting data packets on a WLAN in basic access mode, stations (STA) must sense if the medium is busy. If the medium is found idle and remains so for a time equal to the Distributed InterFrame Space (DIFS) period, the STA may transmit its packet. If the medium is busy, the station must wait until the end of the current transmission. If the medium is not used for a consecutive DIFS period, the STA will draw a random backoff period, unless the backoff-timer already was non-zero. This backoff-timer is decreased for every time slot, τ, waited. The STA may transmit its packet, encapsulated in a MAC data frame, when the backoff-timer reaches zero. If the destination STA correctly receives the MAC frame after a propagation delay, δ, it will send a MAC acknowledgement (ACK) to the source STA after waiting for a time equal to the Short InterFrame Space (SIFS). If this MAC ACK is not received within the ACK timeout, the source STA must retransmit the corresponding MAC data frame. When an erroneous MAC frame is received, stations must backoff from the medium for a time equal to the Extended InterFrame Space (EIFS). For more details we refer the reader to the standard [1]. The time needed for a (correctly received) MAC-acknowledged reception of a TCP segment consisting of x bits is denoted as:

$$T_{da}(x) = difs + T_d(x) + \delta + sifs + T_a + \delta. \tag{3}$$

This transmission cycle is repeated for every data packet that needs to be transmitted from a higher layer protocol and used as in e.g., [12], [11], [7]. A similar transmission cycle operates at the TCP layer, where one endpoint transmits data segments and the receiving end acknowledges (using TCP ACKs) those.

For TCP with delayed acknowledgments, one TCP ACK is transmitted for every two TCP data segments. As a result of the TCP transmission cycle typically one station and the AP contend for the medium at the WLAN layer to send a TCP ACK (encapsulated in 802.11 and IP packets) and two TCP data segments respectively. During the transmission of these three TCP segments, it can be expected that there is one backoff period per cycle in which both stations collide with probability $\frac{1}{Cw_{min}+1}$. This allows formulating the total expected time of a TCP transmission cycle, T_{cycle}, as:

$$\begin{aligned} T_{cycle} &= 2T_{da}(X_{MSS}) + T_{da}(X_{tcp/ip}) \\ &+ \frac{T_{col}}{Cw_{min}+1} \\ &+ \frac{Cw_{min}\,(7Cw_{min}+8)\,\tau}{6(Cw_{min}+1)}, \\ T_{col} &= T_d(X_{MSS}) + \delta + eifs, \end{aligned} \tag{4}$$

where T_{col} is the time consumed by a collision on the WLAN medium because multiple stations used the same backoff time. The largest MAC frame involved in a collision determines the expected time in which the medium is occupied. The remaining parameters are Cw_{min} (minimum contention window), τ (slot time), δ (propagation delay), X_{MSS} (TCP Maximum Segment Size (MSS)), and $X_{tcp/ip}$ (TCP/IP overhead bits). The last term in T_{cycle} represents the total time that the WLAN medium is idle because stations are decrementing their contention window. Following the analysis from [13], this contribution consists of three periods, one in which the AP has scheduled the second TCP data segment for transmission and two other periods in which the station and the AP rival for the medium. Because each station draws its backoff time from a uniform distribution $[0, ..., Cw_{min} - 1]$, the expected contribution of the first period equals $Cw_{min}/2$. The expected contribution of the other two periods equals $\frac{Cw_{min}(4Cw_{min}+5)\tau}{6(Cw_{min}+1)}$ and corresponds to the expectation of the maximum of two independent observations from the uniform backoff distribution (c.f. [13]) .

It is known in the literature that TCP regulates the WLAN medium access because transmissions only occur when there is a data segment to acknowledge [12], [13], which sustains the assumption of low medium contention. The work in literature clearly explains that under these circumstances the medium contention is very low and not sensitive to the size of the user population.

Before the TCP transmission cycle is repeatedly used to transfer the file contents additional FTP and TCP protocol messages are exchanged. As we assume that all users have logged in to the FTP server already, the message sequences needed to setup a file download involves (c.f. [14]) the transmission of an FTP passive command (PASV) with size $X_{ftp-pasv}$ by the station, followed by a 227 (entering passive mode) response (with a piggy-backed TCP ACK) of size $X_{ftp-227}$ by the FTP server (and hence the AP) to confirm the passive file download mode. This 2-way handshake is modeled similar to the first two sequences of the TCP setup in [7], but with the appropriate TCP segment payload values for the PASV request and the 227 response.

$$\begin{aligned} T_{ftp\_setup} &= T_{da}(X_{ftp-pasv}) + T_{da}(X_{ftp-227}) \\ &+ \frac{Cw_{min}(2Cw_{min}+1)\tau}{6(Cw_{min}+1)} \\ &+ \frac{(T_{shortcol}+T_{col})}{Cw_{min}+1}, \\ T_{shortcol} &= T_d(X_{tcp/ip}) + \delta + eifs. \end{aligned} \tag{5}$$

Here, the idle time of the medium due to backoff is expected to be the minimum of two backoff observations because the AP and the statio have both a packet to transmit. The PASV request may collide with a TCP data segment from the AP with the probability of both stations drawing the same backoff interval, whereas the 227 response by the AP may only collide with the smaller packets from the stations. After receiving the 227 (entering passive mode) response from the FTP server, the station initiates a new TCP connection for the data transfer and issues an FTP retrieve command (RETR) with size $X_{ftp-retr}$ to obtain a certain file.

During the TCP set-up cycle, a station starts by initiating a TCP SYN segment, which is followed by a SYN ACK segment by the AP and is concluded by a TCP ACK from the station. The AP will acknowledge on the WLAN medium the packet carrying the TCP SYN from the station. As the AP has always packets to transmit, the SYN ACK is likely to be transmitted amidst a pair of TCP data segments from other flows.

$$\begin{aligned} T_{tcp\_setup} &= 3T_{da}(X_{tcp/ip}) \\ &+ \frac{Cw_{min}(2Cw_{min}+1)\tau}{6(Cw_{min}+1)} \\ &+ \frac{(2T_{shortcol}+T_{col})}{Cw_{min}+1}, \\ T_{shortcol} &= T_d(X_{tcp/ip}) + \delta + eifs. \end{aligned} \tag{6}$$

As soon as the TCP connection is established, the station issues the FTP RETR command and will receive a 150 (opening binary mode data connection) response message of size $X_{ftp-150}$ from the FTP server. Since the station's backoff time does not inhibit the AP from using the medium it is not explicitly modeled, the average time spent on using the medium for the FTP RETR Request and Response (RR) equals:

$$\begin{aligned} T_{ftp\_rr} &= T_{da}(X_{ftp-retr}) + T_{da}(X_{ftp-150}) \\ &+ \frac{Cw_{min}(2Cw_{min}+1)\tau}{6(Cw_{min}+1)} \\ &+ \frac{T_{col}+3T_{shortcol}}{2(Cw_{min}+1)}. \end{aligned} \tag{7}$$

Directly after the 150 response is transmitted by the FTP server, the TCP transmission cycle is repeated up to the point where the last cycle is transmitted. Unlike the model in [7] we assume in the present paper that the user's FTP session remains active and does not require the transmission of an FTP closure command. Hence the last transmission cycle may consist of

one or two TCP data segments, with equal probability, and the size of the last TCP data segment will on average be $X_{MSS}/2$ bits. We therefore conclude that the last TCP transmission cycle transports on average X_{MSS} bits in an expected time equal to:

$$\begin{aligned} T_{lastcycle} &= \frac{T_{da}\Big(X_{MSS}\Big)}{2} + T_{da}\Big(\frac{X_{MSS}}{2}\Big) \\ &+ T_{da}(X_{tcp/ip}) + \frac{T_{halfMSSCol}}{Cw_{min}+1} \\ &+ \frac{Cw_{min}\,(11Cw_{min}+13)\,\tau}{12(Cw_{min}+1)} \\ T_{halfMSSCol} &= T_d\Big(\frac{X_{MSS}}{2}\Big) + \delta + eifs. \end{aligned} \quad (8)$$

In (8) the last term of $T_{lastcycle}$ represents the idle time of the medium due to backoff and accounts for the fact that the last cycle may consist of one or two TCP segments. As a result, two or three backoff periods may occur during the last cycle with equal probability.

After the station has acknowledged the last TCP data segment, it will close the TCP data connection by transmitting a TCP FIN that is acknowledged by the FTP server, followed by an FTP 226 (transfer complete) response of size $X_{ftp-226}$ to indicate that the transfer has completed. Finally, the station acknowledges this response, which concludes the data transfer. Similar to the TCP connection setup (6) and the FTP retrieve request (7) the AP contends with at least one station for the medium, causing the medium idle time to be equal to the minimum of two backoff windows and possible collisions occur with small packets. Accordingly, the station's backoff does not inhibit the medium from being used and the transmissions may collide with the larger TCP data segments from the AP. The expected time to close the FTP data transfer can be approximated by:

$$\begin{aligned} T_{TCP\_close} &= 3T_{da}(X_{tcp/ip}) + T_{da}(X_{ftp-226}) \\ &+ \frac{2Cw_{min}(2Cw_{min}+1)\tau}{6(Cw_{min}+1)} \\ &+ \frac{2T_{shortcol}+2T_{col}}{Cw_{min}+1}. \end{aligned} \quad (9)$$

Note that the TCP closure was not commonly accounted for in WLAN performance models from the literature e.g., [10], because the closure of the TCP connection would not affect the download response time. However, it needs to be noted that the TCP set-up, closure and FTP commands do contribute to the contention and overall load on the network, and thus indirectly to higher file-download times. We refer to Section IV for a brief discussion on accounting traffic that can be attributed to a specific download that does not directly affect the file-download time.

The time consumed by transmitting the FTP overhead and TCP session opening/closing as defined by Equation (5), (6), (7) and (9) is calculated by:

$$\begin{aligned} T_{tcpftp_{OH}} &= T_{ftp\_setup} + T_{tcp\_setup} + T_{ftp\_rr} \\ &+ T_{TCP\_close}. \end{aligned} \quad (10)$$

This includes, in addition to the file contents, the following FTP application commands and responses:

$$\begin{aligned} X_{ftp\_cmds} &= X_{ftp-pasv} + X_{ftp-227} \\ &+ X_{ftp-retr} + X_{ftp-150} + X_{ftp-226}. \end{aligned} \quad (11)$$

Now, the overall *effective throughput*, denoted TP_{eff}, at the application layer is obtained by:

$$TP_{eff} = \frac{X_{file}}{\frac{\Big(X_{file}-X_{MSS}\Big)T_{cycle}}{2X_{MSS}} + T_{lastcycle} + T_{tcpftp_{OH}}}, \quad (12)$$

with X_{file} the mean file-size in bits.

B. Flow-level model

For the flow-level abstraction we consider a classical $M/G/1$ PS-model, with flow-arrival rate λ, and where the service time B is generally distributed with mean β. In this model, incoming flows immediately enter the system, thereby receiving a fair share of the available capacity. Then the occupation rate is $\rho := \lambda\beta$, and the expected sojourn time is known to be $E[S] = \beta/(1-\rho)$. Note here that the sojourn time is *insensitive* to the service-time distribution, i.e., depends on the service-time distribution only through its mean β.

To include the packet-level details in our analytic model for WLAN file downloads into an $M/G/1$ PS-model, we define the following notion of *effective load*:

$$\rho_{eff} := \lambda \cdot \left(\frac{X_{file}}{TP_{eff}}\right), \quad (13)$$

where ρ_{eff} can be viewed as the effective medium utilization resulting from the load introduced by processing λ file download requests per second. Since the file download in the WLAN network encompasses the file transfer and the introduced overhead of FTP and TCP, the expected file-download time (denoted $E[R]$) is modeled as the expected sojourn time in an $M/G/1$ PS-model with load ρ_{eff}:

$$E[R] = \frac{\rho_{eff}/\lambda}{1-\rho_{eff}}. \quad (14)$$

Thus, to apply the analytic model the average file-download time $E[R]$ is obtained from (14), where ρ_{eff} is given by (13), and TP_{eff} follows from Equation (12).

III. Model validation for FTP downloads in WLAN Testbed

A. Experimental Setup

This section summarizes the results of extensive testbed experiments. Our aim is to validate whether (1) our analytic model accurately predicts the outcomes from our testbed, and (2) whether the mean download time in our testbed is indeed not very sensitive to the file-size distribution, as suggested by the M/G/1 PS-model. To this end, we have connected two powerful PCs; one functioning as FTP server, and the other as FTP clients. These PCs are interconnected by two independent and identical access networks to measure the download response

times of file transfers in each network separately and compare the obtained values with the expected file-download time from the analytic model. In the experimental setup, the PC with the FTP clients generates all download requests, according to independent Poisson processes. It is important to state that with a larger number of WLAN client devices there is, in addition to the AP, typically only one station contending for the medium at the same time, as reported in [12] and observed during the experiments in [7]. This justifies the choice of using one client device for our downloads rather than a large population.

At the PC serving as FTP server, files were been generated according to an exponential and a hyper-exponential distribution with high variability (with a squared coefficient of variation equal to four, $c^2 = 4$), each of which consisted of 40,000 different files with mean size 2×10^5 bytes that are retrieved in a random order by the FTP clients.

In our testbed environment, each wireless access network consists of a Linksys (WAP54G) access point and an Ethernet bridge (WET54G) of the same hardware and firmware version connected by a power splitter to avoid interference and yield a small propagation delay, δ. The access point uses a modified firmware program, called OpenWrt, that is specifically designed for embedded devices such as residential gateways and routers. This firmware offers detailed WLAN-MAC configuration options. Table II summarizes the parameters of our testbed configuration that are used in addition to those specified in Table I for the WLAN equipment.

TABLE II
TESTBED ENVIRONMENT AND MODEL PARAMETERS.

Parameter	Value	Parameter	Value
$X_{ftp-pasv}$	48 bits	$X_{ftp-227}$	392 bits
$X_{ftp-retr}$	272 bits	$X_{ftp-150}$	704 bits
$X_{ftp-226}$	184 bits	X_{MSS}	11680 bits
w	70080 bits	$X_{tcp/ip}$	320 bits
δ	10^9s		

In Table II it is shown that the TCP stack in our testbed environment was configured to use an MSS, indicated as X_{MSS}, of 1460 bytes and a window-size, w, of 8760 bytes. Therefore we use in our analytic model 40 bytes of TCP/IP overhead, represented by variable $X_{tcp/ip}$. In addition, the FTP commands and responses that were introduced in Section II are also specified in Table II.

B. Experimental Results

We have conducted extensive experimentation to validate our model. Runs were performed for two different file-size distributions, for two different networks and eight different load values. Each run was executed until a sufficiently small 95% Confidence Interval (CI) was obtained, often requiring durations of over 55 hours to gather up to 450.000 observations per run. The representative outcomes of our experiments are outlined below.

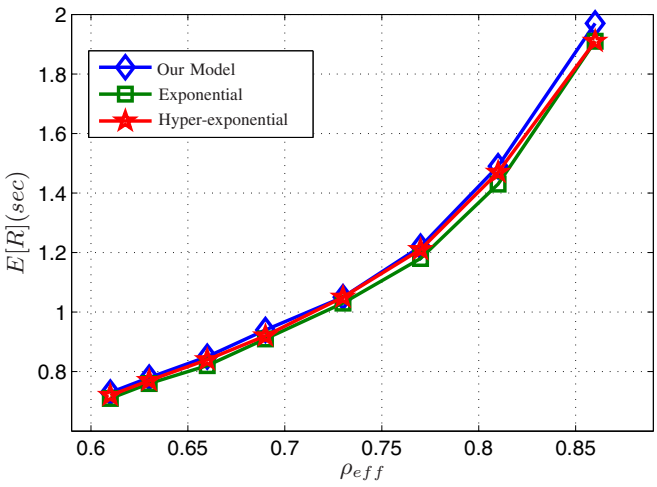

Fig. 1. Analytic and experimental results on the average download response time $E[R]$ for exponential and hyper-exponential (H_2, with c^2 = 4) file-size distributions with an average of 200kbytes.

TABLE III
ANALYTIC AND EXPERIMENTAL E[R] FOR $X_{file} = 200kByte$.

ρ_{eff}	A.Model	Exp	95%-CI	H_2,c^2=4	95%-CI	Observations
0.60	0.73	0.71	1.97%	0.72	2.67%	15×10^4
0.63	0.78	0.76	1.54%	0.77	2.16%	25×10^4
0.66	0.85	0.82	1.72%	0.84	2.42%	25×10^4
0.69	0.94	0.91	1.40%	0.92	1.87%	45×10^4
0.72	1.05	1.03	1.49%	1.05	2.04%	45×10^4
0.76	1.22	1.18	1.77%	1.21	2.37%	45×10^4
0.81	1.49	1.43	1.95%	1.47	2.37%	45×10^4
0.85	1.97	1.91	2.27%	1.91	2.50%	45×10^4

The results in Table III and Figure 1 demonstrate that the analytic results closely match the outcomes from the experimental testbed for a broad range of model parameters with typical errors of $1 - 3\%$ up to a maximum of 5.6%. Indeed, there is no significant dependence of the mean download time in our experiments to the file-size distribution, which extends the applicability of this PS-modeling approach towards real network equipment.

IV. DISCUSSION ON OBSERVED DOWNLOAD TIME

In the interest of obtaining an accurate expression for the effective load of the network (13), the model described in Section II-A captures all data traffic that can be attributed to a file download. Some modeling approaches from the literature [10] do not consider the traffic associated to a file download that has no (direct) impact on the file-download time. The approach presented in the present paper, however, does consider all data traffic associated to a file download to calculate the accurate expected sojourn times from realistic effective load values, even under high traffic loads. The download response time as observed by the FTP application includes only those interactions from the setup of the TCP data connection up to the one in which the last data segment is received, and thus excludes the interactions related to the FTP PASV request

(5) and the FTP/TCP closure (9). Therefore, an alternative approach is to define the observed service time to transfer a file of X_{file} bits through the network considered in this paper as:

$$\beta_{obs} = \frac{\left(X_{file} - \frac{X_{MSS}}{2}\right)T_{cycle}}{2X_{MSS}} + T_{lastcycle} + T_{tcp\_setup} + T_{ftp\_rr}, \quad (15)$$

where only those data exchanges are included that directly contribute to the file-download time as observed by the FTP application. When subsequently applying the $M/G/1$ PS-model from Section II-B we obtain the following expression for the expected observed file-download time (denoted $E[R]_{obs}$) in a network with an effective load ρ_{eff}:

$$E[R]_{obs} = \frac{\beta_{obs}}{1 - \rho_{eff}}. \quad (16)$$

In this way, the contribution of the traffic that only has a direct impact on the load of the network is already included in ρ_{eff} from (13) and omitted from the download-time that is observed by the FTP application (16). Applying this alternative approach further increases the accuracy of the analytic model specified in Table III to obtain outcomes with typical errors of $1-2\%$ up to a maximum of 4.5% and thus reduces the slightly over-estimated outcomes from the flow-level model from (14). It needs to be noted, however, that in (16) the observed service time β_{obs} is not equal to ρ_{eff}/λ, in contrast with what one would expect from an $M/G/1$ PS-model application.

V. Conclusions

In this paper, we have proposed a detailed analytic model for the download response time FTP-based file transfers over WLANs that matches practical deployments. The model translates the complex dynamics of the FTP/TCP/IP/MAC-stack, and their interactions, into a single parameter, called the effective load. We use this parameter to describe the flow-level behavior of FTP-based file transfers over WLANs without admission control as an M/G/1 processor-sharing (PS)-model. The model is validated by experiments in a testbed environment. The results show that the model leads to accurate predictions over a broad set of parameter combinations, and as a by-product, the results show that the mean download times are indeed fairly insensitive to the file-size distribution, as suggested by the PS-model. Further improvements to the model accuracy can be achieved by defining the notion of observed service time, in which the traffic that has no direct impact on the file-download time is omitted but still captured in the effective load of the network.

VI. Acklowledgments

The work reported in this paper was supported by the Netherlands Organisation for Scientific Research (NWO) under the Casimir project: Analysis of Distribution Strategies for Concurrent Access in Wireless Communication Networks.

References

[1] ANSI/IEEE Standard 802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. 1999.

[2] J.V.L. Beckers, I. Hendrawan, R.E. Kooij, and R.D. van der Mei. Generalized processor sharing models for internet access lines. In *Proceedings of IFIP Conference on Performance Modelling and Evaluation of ATM and IP networks*, pages 101–112, Budapast, 2001.

[3] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[4] S.C. Borst, O.J. Boxma, and N. Hegde. Sojourn times in finite-capacity processor-sharing queues. In *Proceedings NGI 2005 Conference*, 2005.

[5] N.T. Dao and R.A. Malaney. Throughput performance of saturated 802.11g networks. In *AUSWIRELESS '07: Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communications*, 2007.

[6] G.J. Hoekstra and R.D. van der Mei. Effective load for flow-level performance modeling of wireless LANs. *to appear in Computer Communications*, 2010.

[7] G.J. Hoekstra and R.D. van der Mei. On the Processor Sharing of File Transfers in Wireless LANs. In *Proceedings of the 69th IEEE Vehicular Technology Conference, VTC Spring 2009*, Barcelona, Spain, 26-29 April 2009.

[8] R. Litjens, F. Roijers, J.L. Van den Berg, R.J. Boucherie, and M.J. Fleuren. Performance analysis of wireless LANs: An integrated packet/flow level approach. In *Proceedings of the 18th International Teletraffic Congress - ITC18*, pages 931–940, Berlin, Germany, 2003.

[9] P. Mahasukhon, M. Hempel, S. Ci, and H. Sharif. Comparison of throughput performance for the IEEE 802.11a and 802.11g networks. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications (AINA 2007), May 21-23, 2007, Niagara Falls, Canada*, pages 792–799. IEEE Computer Society, 2007.

[10] D. Miorandi, A.A. Kherani, and E. Altman. A queueing model for HTTP traffic over IEEE 802.11 WLANs. *Computer Networks*, 50(1):63–79, 2006.

[11] F.J.M. Panken and G.J. Hoekstra. Multi-service traffic profiles to realise and maintain QoS guarantees in wireless LANs. *Computer Communications*, 32(6):1022 – 1033, 2009.

[12] F. Roijers, J.L. van den Berg, and X. Fang. Analytical modelling of TCP file transfer times over 802.11 wireless LANs. In *Proceedings of the 19th International Teletraffic Congress - ITC19*, Beijing, China, 2005.

[13] T. Sakurai and S. Hanley. Modelling TCP flows over an 802.11 wireless LAN. In *Proceedings of European Wireless Conference*, 2005.

[14] W.R. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley Professional, New York, NY, USA, fourth edition, 1993.

[15] Y. Wu, C. Williamson, and J. Luo. On processor sharing and its applications to cellular data network provisioning. *Performance Evaluation*, 64(9-12):892–908, 2007.

.