INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2014)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2832

An analysis of energy consumption for TCP data transfer with burst transmission over a wireless LAN

Masafumi Hashimoto*,†, Go Hasegawa and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871, Japan

SUMMARY

A common strategy for energy saving in wireless network devices is to remain in sleep mode for as long as possible. The timing of packet transmission and reception depends on the behavior of the transport-layer protocols used by upper-layer applications. Therefore, understanding the relation between the behavior of the transport-layer protocols and energy efficiency using sleep mode is important for effective energy saving, especially when a wireless network interface (WNI) is activated in sleep mode at packet interarrivals. In this paper, we analyze the energy consumption of a client's WNI in Transmission Control Protocol (TCP) data transfer over a wireless LAN by focusing on the detailed behavior of TCP congestion control mechanisms. This model considers three situations: the WNI is activated in continuously active mode, in sleep mode, and in sleep mode with *burst transmission*. The latter is proposed as an effective method to improve energy efficiency, which lengthens sleep periods by transmitting and receiving multiple packets in a bursty fashion. Through numerical examples, we show that sleeping without modification of transmission timing reduces energy consumption in TCP data transfer by only around 10%, and that the burst transmission can contribute further 50% energy reduction. Copyright © 2014 John Wiley & Sons, Ltd.

Received 13 December 2012; Revised 20 May 2014; Accepted 16 June 2014

KEY WORDS: Transmission Control Protocol (TCP); wireless LAN; energy consumption model; energy efficiency

1. INTRODUCTION

With recent developments in wireless network technologies, the Internet is increasingly accessed via IEEE 802.11-based wireless LANs (WLANs) by using mobile devices, such as mobile phones, smartphones, laptops, and tablet PCs. Wireless communication through a mobile device can account for approximately between 10% and 50% of the device's total energy consumption [1–3]. Therefore, there is a great deal of interest in reducing the energy consumed through wireless communication, particularly because most mobile devices are battery-driven.

For energy saving in media access control (MAC) layer protocols, the IEEE 802.11 standard defines a power saving mode (PSM) [4], as opposed to the mode under normal operation, which is referred to as the continuously active mode (CAM).

Although PSM can significantly reduce energy consumption, it can also degrade network performance characteristics, such as throughput and latency [5]. Many researchers have proposed energy-efficient methods in WLANs [5–18].

Some of them achieve high-energy efficiency by mainly modifying MAC protocols. The others are energy-efficient solutions for the specific applications. In contrast, in this paper, we aim to derive general-purpose transport-layer solution for energy saving without requiring any modifications for wireless network interface (WNI) hardware.

*Correspondence to: Masafumi Hashimoto, Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871, Japan.
†E-mail: m-hasimt@ist.osaka-u.ac.jp

Other researchers have constructed energy consumption models for WLAN clients to find factors that affect energy efficiency and network performance in WLANs [19–21]. However, they mainly focus on the behavior at only the MAC level and do not consider detailed TCP behavior. In contrast, TCP congestion control mechanisms primarily determine the timing of packet transmission and reception, which has a large impact on the energy efficiency of sleep behavior.

To assess the impact of TCP on sleep efficiency, in this paper, we construct a new energy consumption model for upstream flow in TCP data transfer over a WLAN. The proposed model consists of the combination of two layer models: a MAC-level model and a TCP-level model.

We derive energy consumption models for a device with CAM and with *ideal sleep* mode to reveal the sleep efficiency while considering detailed TCP behavior. Ideal sleep mode implies that a WNI knows the schedules of both the transmission and reception of TCP packets such that it can sleep and wake up with exact timing. Furthermore, in order to improve the sleep efficiency, we propose TCP-level *burst transmission* behavior, which lengthens sleep periods by transmitting multiple data packets in groups.

The main contributions of this paper are as follows. We first show that sleeping at a short time scale, but not at a long time scale, is a meaningful method to save energy. Second is that we present a possibility of transmission control at transport-layer level to contribute to effective energy saving.

The remainder of this paper is organized as follows. First, we show recent works on energy saving in WLAN clients in Section 2. We describe the network model and our assumptions in Section 3. Then, in Section 4, our models of energy consumption during TCP data transfer over a WLAN are introduced. Section 5 describes the simulation-based validation for our models. Section 6 shows numerical results of the analysis based on our models. Finally, conclusions and future research directions are presented in Section 7.

2. RELATED WORK

In IEEE 802.11 PSM, wireless clients only wake up at the beacon interval, which is typically 100 ms. Because of this, PSM can achieve high-energy efficiency while degrading network performance such as throughput and latency. To overcome this, IEEE 802.11e defines new power saving mechanisms, which is called automatic power save delivery [6]. In addition to modification of IEEE 802.11 power saving mechanism, many researchers have proposed energy-efficient methods in WLANs by mainly modifying MAC protocols or WNI hardware [5, 8–13].

Some solutions for energy saving of wireless clients focus on the behavior of the upper-layer protocols [14–17]. Kim *et al.* [14] presented an energy-aware transmission mechanism with split TCP connections. Namboodiri and Gao [15] proposed GreenCall algorithm for VoIP applications, which derives sleep and wake-up schedules for the wireless client to save energy during VoIP calls. Dogar *et al.* [16] developed the Catnap proxy for data-oriented applications such as web browsing and file transfer.

Yan *et al.* [17] presented a client-centered method in TCP over WLANs, with burst transmission realized by manipulating the TCP receiver's window size.

In terms of sleep granularity, the Catnap proxy allows wireless clients to stay sleep mode at the whole data transmission scale, whereas the client-centered method allows them to stay sleep mode at the round trip time (RTT) scale. Wireless clients with our proposed method can sleep at an interarrival time of acknowledgement (ACK) packets within one RTT, which can be changed as a function of the number of packets sent in a bursty fashion, because the recent development in RF circuit design has resulted in sleep mode of shorter transition time [11].

In order to investigate the factors that affect energy consumption of wireless clients in WLANs, many researchers have constructed energy consumption models for WLAN clients [19–21]. Anastasi *et al.* [19] modeled a single wireless client in PSM downloading a file from a server in the presence of multiple wireless clients. Ergen and Varaiya [20] presented the results of an analysis of energy consumption during different MAC operations for a wireless client with multiple clients in a WLAN; they found that 80% of the total energy in saturated situations is wasted. Agrawal *et al.* [21] created a discrete-time Markov chain model of the energy consumption for TCP transfers

DOI: 10.1002/dac

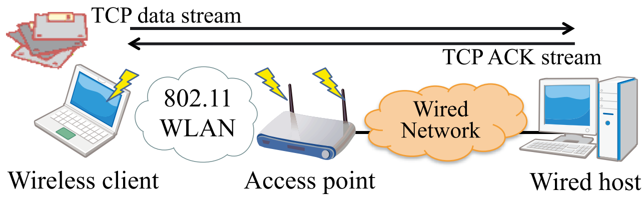


Figure 1. Wireless LAN environment.

in CAM and in PSM in the presence of TCP background traffic. However, most of the aforementioned researches mainly focus on the behavior at only the MAC level and do not consider detailed TCP behavior. Because TCP congestion control mechanisms primarily determine the timing of packet transmission and reception, analysis of the detailed TCP behavior is necessary to assess sleep efficiency at packet interarrivals.

3. NETWORK MODEL AND ASSUMPTIONS

Suppose we have a WLAN environment in which a single wireless client associates with an access point (AP) connected to a host via a wired network (Figure 1)[‡]. In the WLAN, the wireless client sends a file of S_d bytes to the wired host by TCP; that is, we consider upstream TCP data transfer. Note that our model can be easily adjusted to deal with downstream TCP data transfer. In Section 4, energy consumption is taken to be from the transmission of the first packet of the file until the reception of the ACK packet for the last packet of the file. We assume that at the MAC level, the wireless client and the AP do not use the request to send/clear to send (RTS/CTS) mechanism because it is an optional mechanism.

Suppose that at the hardware level the WNI has four communication modes—*transmit*, *receive*, *idle* or *listen*, and *sleep*—and each of these modes has a different power consumption denoted by P^t, P^r, P^l, and P^s, respectively [1]. Furthermore, the WNI consumes power when transiting between active and sleep modes, and we define P^{as} and P^{sa} as the power consumption when changing from and to active mode, respectively. The periods of these power consumptions are then denoted by T^{as} and T^{sa}, respectively. In order to focus on the impact on sleep efficiency by TCP, we assume idle sleep mode, which means that a WNI knows the schedules of both the transmission and reception of TCP packets such that it can sleep and wake up with exact timing.

We assume that data frames are lost randomly at MAC-level of the WLAN because of channel error and collisions. In addition, TCP data packets are lost in the wired network due to network congestion. For simplicity, the probability of frame transmission failures at MAC-level, q, and the probability of TCP-level packet loss events in the wired network, p_l, are given. Let p_w denote the probability of TCP-level packet loss events in the WLAN, which is calculated as $p_w = q^{N+1}$ where N is the maximum number of frame retransmissions at MAC-level. Therefore, the TCP at the wireless client experiences the packet loss events at a probability of p, which is defined as $p = 1 - (1 - p_w)(1 - p_l)$.

Other assumptions are as follows.

- TCP Reno is used.
- No TCP ACK packets are lost in the wired network.
- The available bandwidth in the WLAN is larger than that in the wired network, that is, a link in the wired network is bottleneck, because it is a necessary condition to sleep at packet interarrivals.
- Energy consumption due to retransmission of data packets is not considered because it has a small impact on sleep efficiency.

[‡]In this paper, our model does not consider situations in which multiple clients share a single AP. Nevertheless, our model is reasonably validated because we can ignore the effect of frame collisions when congestion-level is moderate (Section 5).

DOI: 10.1002/dac

Table I. Notation of given variables.

Symbol	Description
P^t, P^r, P^l, P^s	Power consumed during transmit, receive, listen, and sleep modes, respectively
P^{as}, P^{sa}	Power consumed during state transitions to and from sleep modes, respectively
T^{as}, T^{sa}	Duration during state transitions to and from sleep modes, respectively
p	Probability of TCP-level packet loss events at a wireless client
q	Probability of frame transmission failures at MAC-level
p_l, p_w	Probabilities of TCP-level packet loss events in the wired network and in the WLAN, respectively
RTT	RTT between a client and a wired host
m	Number of packets sent in a bursty fashion
N	Maximum number of data frame retransmissions
S_d	File size in bytes
S_p	TCP data packet size in bytes

RTT, round trip time; MAC, media access control; WLAN, wireless LAN.

- TCP-level burst transmission is achieved by using TCP delayed ACK [22]. When the delayed ACK is used, growth of the TCP congestion window has been reported to be inhibited [23]. In this paper, we assume that this problem has been resolved.
- Unless otherwise noted, we follow the assumptions of Padhye *et al.* [24] and Cardwell *et al.* [25] for TCP congestion control behavior.

Some researchers modeled the behavior of the TCP congestion mechanisms under the existence of channel error [26–28]. Zorzi and Rao [26] modeled energy consumption for different TCP versions while considering a two-state Markov model for frame losses. Vacirca *et al.* [27] constructed an energy consumption model that consists of combination of a TCP behavior model and a MAC model under a Rayleigh fading channel. These two models consider energy consumption for a device activated in CAM. In this paper, we mainly focus on the impact of the TCP behavior for sleep efficiency when the WNI sleeps at packet interarrivals. To this end, we assume that the available bandwidth of the WLAN is larger than that of the wired network, which is a necessary condition for sleeping at packet interarrivals in the WLAN. In this situation, frame losses due to collision or channel error are moderate in the WLAN to obtain the sufficient available bandwidth in the WLAN. Therefore, we consider a simple frame error model, as mentioned in this section, in the WLAN against the aforementioned models [26, 27].

4. ENERGY CONSUMPTION MODELS

In this section, we formulate our models which describe expected energy consumed at the WNI of a wireless client during TCP transfer of S_d bytes data. The behavior of MAC-layer protocols determines the energy consumption when a single data frame is transmitted and received, whereas the number of data packets sent and received per RTT is dependent on the behavior of TCP congestion control mechanisms. Thus, using the MAC-level model developed in Subsect. 4.1, we derive an energy consumption model in TCP data transfer in Subsect. 4.2.

The main notation used in this section is defined in Tables I, II, and III. Table I lists given variables used throughout this section. Tables II and III summarize notation used mainly in Subsect. 4.1 and 4.2, respectively. Other symbols are defined as required.

4.1. Energy consumption during frame exchanges for IEEE 802.11 MAC

We first present a model of energy consumption at the MAC level when a wireless client transmits or receives one data frame[§]. Specifically, we derive $E[e^t]$ and $E[e^r]$. Figure 2 shows details of the

[§]Because we consider upstream TCP data transfer, TCP data and TCP ACK packets are contained in the data frame sent from and received at the client, respectively.

DOI: 10.1002/dac

Table II. Notation used in MAC-level model.

Symbol	Description
$E[e^t]$, $E[e^r]$	Expected energy consumptions for a client to transmit and receive a data frame, respectively
$E[t^t]$, $E[t^r]$	Expected period for a client to transmit and receive a data frame, respectively
$e^t(i)$, $e^r(i)$	Energy consumptions for the ith data frame transmission and reception after $(i-1)$ failures, respectively
$t^t(i)$, $t^r(i)$	Average period for a client to send and receive a data frame for the ith transmission, respectively
$CW(i)$	Contention window size of ith transmission after $(i-1)$ consecutive transmission failures
CW_{min}, CW_{max}	Minimum and maximum values of the contention window size, respectively
$T_{BO}(i)$	Expected backoff time of ith transmission after $(i-1)$ consecutive transmission failures
T_{slot}	Slot time
T_{SIFS}, T_{DIFS}	Short interframe space (SIFS) and DIFS time, respectively
T^{client}_{DATA}, T^{AP}_{DATA}	Transmission duration of a data from a client and an AP, respectively
T_{ACK}	Transmission duration of an ACK frame
τ	Radio propagation delay between a client and an AP
$Q(i)$	Probability distribution that a data frame is transmitted i times until it becomes successful

WNI, wireless network interface; AP, access point; DIFS, distributed interframe space; RTO, retransmission timeouts; RTT, round trip time.

frame exchanges between a wireless client and an AP. When the wireless client sends one data frame to the AP, it sends the data frame after a distributed interframe space (DIFS) and a random backoff time. It receives an ACK frame from the AP when the transmission succeeds. When the client does not receive the ACK frame, it lengthens the backoff time and retries the data frame transmission. Data frame is sent from the AP to the client in a similar way to the aforementioned sequence.

In the aforementioned sequences, the expected backoff time of ith transmission after $(i-1)$ consecutive transmission failures is determined by the following equations:

$$T_{BO}(i) = CW(i)T_{slot}/2 \tag{1}$$

where T_{slot} is the slot time. From the IEEE 802.11 standard [4], $CW(i)$ is given by

$$CW(i) = \min\left((CW_{min}+1)2^{i-1}-1,\ CW_{max}\right). \tag{2}$$

From Figure 2 and Equation (1), $t^t(i)$ and $t^r(i)$ are calculated as follows:

$$t^t(i) = T_{SIFS} + T_{DIFS} + T_{BO}(i) + 2\tau + T^{client}_{DATA} + T_{ACK} \tag{3}$$

$$t^r(i) = T_{SIFS} + T_{DIFS} + T_{BO}(i) + 2\tau + T^{AP}_{DATA} + T_{ACK}. \tag{4}$$

Here, we consider $Q(i)$. Using q, a probability that a data frame is transmitted i times is calculated as $q^{i-1}(1-q)$. Because the frame transmission is limited up to $N+1$ times, a probability that the number of frame transmissions reaches $N+1$ is obtained as $(1-\sum_{i=1}^{N} q^{i-1}(1-q) = q^N)$. Thus, $Q(i)$ can be calculated as follows.

$$Q(i) = \begin{cases} q^{i-1}(1-q) & \text{if } i \leqslant N \\ q^N & \text{if } i = N+1 \end{cases} \tag{5}$$

Table III. Notation used in TCP-level model.

Symbol	Description
$E[J_{cam}]$, $E[J_{sleep}]$	Expected energy consumption with CAM and with sleeping during the whole TCP data transfer, respectively
$E[T^{all}]$, $E[\hat{T}^{all}]$	Expected period of the whole data transfer without and with burst transmission, respectively
$E[J^t]$, $E[J^r]$	Expected energy consumed in transmission and reception of TCP packets during the whole data transfer, respectively
$E[J^l_{cam}]$, $E[J^l_{sleep}]$	Expected energy consumed during idle periods with CAM and with sleeping, respectively
$E[J^s]$	Expected energy consumed during sleep periods
$E[J^{st}]$	Expected energy consumed due to state transitions between active and sleep modes
$E[T^t]$, $E[T^r]$, $E[T^s]$, $E[T^{st}]$	Expected periods of packet transmission, of packet reception, of sleep periods, and of state transitions during the whole data transfer, respectively
$E[r_\zeta]$, $E[r_\eta]$	Expected number of rounds during the initial slow start phase and during the steady phase, respectively
$E[r_{ss}]$, $E[r_{td}]$, $E[r_{to}]$	Expected numbers of rounds during the data transfer of the initial slow start phase, of a TD period, and of a TO period, respectively
$E[n^t]$, $E[n^r]$	Expected numbers of TCP packets sent and received during the whole data transfer, respectively,
$E[n^r_\zeta]$, $E[n^r_\eta)]$	Expected number of TCP packets received during the initial slow start phase and during the steady phase, respectively
$E[n_{cycle}]$	Expected numbers of cycles in the steady phase
w^{ss}_k, w^{ca}_k	Congestion window size of the kth round in the initial slow start phase and in the TD period, respectively
$E[t^s_\zeta]$, $E[t^s_\eta]$	Expected sleep periods during the initial slow start phase and during the steady phase, respectively
$t^a(w_k)$	An interarrival time of TCP ACK packets as a function of the congestion window size w_k in the kth round
$E[t^s(w_k)]$	Expected sleep period that the WNI sleeps at an interarrival time of ACK packets
$E[t^s_d(r,\ w_k)]$	Expected sleep period from the first to the rth rounds
$E[t^s_{td}(r,\ w_k)]$	Expected sleep period in the round in which a packet loss event is detected by triple duplicate ACK packets
$E[t^s_{to}(r,\ w_k)]$	Expected sleep period of a sequence of RTOs
$E[n^{st}_\zeta]$, $E[n^{st}_\eta]$	Expected numbers of state transitions during the initial slow start phase and during the steady phase, respectively
$E[n^{st}(w_k)]$	Expected number of state transitions between active and sleep modes at an interarrival time of ACK packets
$E[n^{st}_d(r,\ w_k)]$	Expected number of state transitions from the first to the rth rounds
$E[n^{st}_{td}(r,\ w_k)]$	Expected number of state transitions during the round in which a packet loss event is detected by triple duplicate ACK packets
$E[n^{st}_{to}]$	Expected number of state transitions during the duration of a sequence of RTOs
$E[d_{ss}]$	Expected number of packets sent during the initial slow start phase
$E[W_{ss}]$, $E[W]$	Expected size of the congestion widow when a packet loss event occurs in the initial slow start phase and in the TD period, respectively
W_{bdp}	Maximum number of packets injected into the WLAN per RTT at the TCP-level
$E[R]$	Expected number of packets sent during RTOs
$E[n]$	Expected number of TD periods in a cycle
$Q(w, p)$	Probability that a packet loss is detected by a RTO as functions of window size w and probability p of packet loss events
l_{ss}	Probability that at least one packet is lost in the initial slow start phase

CAM, continuously active mode; WNI, wireless network interface; RTO, retransmission timeout.

On the other hand, periods in which the ith transmission and reception becomes successful after $(i-1)$ failures are obviously given by $\sum_{j=1}^{i} t^t(j)$ and $\sum_{j=1}^{i} t^r(j)$, respectively. Thus, using Equations (3), (4), and (5), $E[t^t]$ and $E[t^r]$ are obtained as

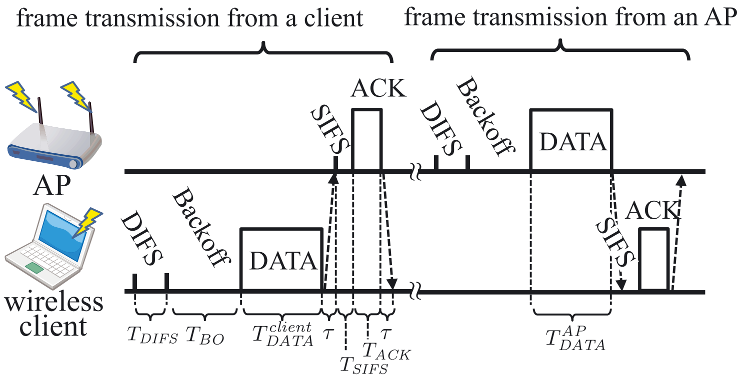


Figure 2. Frame exchange in IEEE 802.11 MAC.

$$E[t^t] = \sum_{i=1}^{N+1} \sum_{j=1}^{i} t^t(j)Q(i), \quad E[t^r] = \sum_{i=1}^{N+1} \sum_{j=1}^{i} t^r(j)Q(i). \tag{6}$$

Next, we determine the expected energy consumed during transmission ($E[e^t]$) and reception ($E[e^r]$) of one data frame. From Figure 2 and Equation (1), $e^t(i)$ and $e^r(i)$ are obtained by

$$e^t(i) = P^l(T_{SIFS} + T_{DIFS} + T_{BO}(i) + 2\tau) + P^t T_{DATA}^{client} + P^r T_{ACK}, \tag{7}$$

$$e^r(i) = P^l(T_{SIFS} + T_{DIFS} + T_{BO}(i) + 2\tau) + P^t T_{ACK} + P^r T_{DATA}^{AP}. \tag{8}$$

In a similar way to the calculation process for Equation (6), using Equations (5), (7), and (8), $E[e^t]$ and $E[e^r]$ are calculated as follows:

$$E[e^t] = \sum_{i=1}^{N+1} \sum_{j=1}^{i} e^t(j)Q(i), \quad E[e^r] = \sum_{i=1}^{N+1} \sum_{j=1}^{i} e^r(j)Q(i). \tag{9}$$

4.2. *Energy consumption of TCP data transfer*

We now construct a TCP-level model of energy consumption during TCP data transfer. Expected energy consumed during the whole data transfer is calculated by the sum of energy consumption of each state of a WNI. Thus, $E[J_{cam}]$ and $E[J_{sleep}]$ are obtained as

$$E[J_{cam}] = E[J^t] + E[J^r] + E[J_{cam}^l], \tag{10}$$

$$E[J_{sleep}] = E[J^t] + E[J^r] + E[J_{sleep}^l] + E[J^s] + E[J^{st}]. \tag{11}$$

In what follows, we derive each term in Equations (10) and (11) by utilizing TCP analysis models [24, 25]. Padhye *et al.* [24] formulated the average TCP throughput analyzed based on the detailed behavior of TCP congestion control mechanisms. By extending [24], Cardwell *et al.* [25] derive the expected TCP latency of finite-size data transfer. The derivations of both models are based on a *round* that starts when the first packet of a window is transmitted and ends when the corresponding ACK packet is received. Note that a round makes a RTT. We determine the terms in Equations (10) and (11) by using the expected number of rounds during the data transfer of S_d bytes, which is derived in [24, 25].

DOI: 10.1002/dac

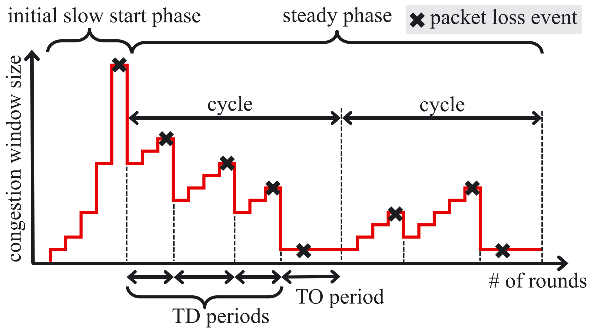


Figure 3. An example of evolution of congestion window size.

In the following analysis, in Section 4.2.1, we first formulate the number of rounds required for S_{itd} bytes data transfer. We then derive $E[J^t]$ and $E[J^r]$ in Section 4.2.2. In addition, $E[J^l_{cam}]$ and $E[J^l_{sleep}]$ are formulated in Section 4.2.3. Sections 4.2.4 and 4.2.5 show the derivations of $E[J^s]$ and $E[J^{st}]$, respectively. Finally, we formulate the increase in data transfer latency when bust transmission is used in Section 4.2.6.

4.2.1. Number of rounds during TCP data transfer. In TCP data transfer, the number of packets sent per round is determined by the growth of the TCP congestion window size. Figure 3 depicts the typical evolution of the congestion window size of a TCP connection from the beginning of the transmission. The data transfer starts with the *initial slow start phase* that ends because of the occurrence of a packet loss event. After that, data packets are sent in the *steady phase* until the data transfer ends. TCP continues increasing its congestion window size until a packet loss event is detected and decreases it after detecting the packet loss event. The packet loss events are detected either by triple duplicate ACK packets or by Retransmission Timeouts (RTOs). Here, we define a Triple Duplicate (TD) period as the duration between two consecutive packet loss events detected by triple duplicate ACK packets. Further, the duration of a sequence of RTO is referred to as a Timeout (TO) period. In the steady phase, we can observe one TO period appears after multiple TD periods, and this sequence appears repeatedly [24]. Here, we define a *cycle* as the duration between two consecutive packet loss events detected by RTOs. Then, the steady phase is constructed by multiple cycles.

We mainly follow the TCP latency formulations derived in [24, 25] to calculate the expected number of rounds required for the whole data transfer. Using $E[r_\xi]$ and $E[r_\eta]$, the expected number of rounds during the whole data transfer is obtained as $(E[r_\xi] + E[r_\eta])$. Refer to Appendix A for the derivations of $E[r_\xi]$ and $E[r_\eta]$. However, [24, 25] assume situations in which the packets sent per round is constrained by the maximum window size, whereas we assume situations in which the packets sent per round is constrained by the wireless network bandwidth. Due to this, we modify some equations in [24, 25].

The modification of the constraint affects the evolution of the congestion window size. When the number of packets sent per round is constrained by the maximum window size, the evolution of the congestion window size stops when the congestion window size reaches the maximum window size. However, when the number of packets sent per round is constrained by the wireless network bandwidth, the congestion window continues increasing while the number of packets injected into the network is limited by the bandwidth. In the following, considering the aforementioned situations, we derive the expected numbers of rounds during the data transfer of the initial slow start phase $(E[r_{ss}])$ and during the TD period $(E[r_{td}])$.

Using $E[t^t]$ and $E[t^r]$ derived in Section 4.1, W_{bdp} is calculated as follows. When the delayed ACK is used, the TCP receiver sends an ACK packet every received m data packets. Therefore,

DOI: 10.1002/dac

when the burst transmission is used, the number of packets sent per unit time at the TCP-level is calculated as $1/(E[t^t] + E[t^r]/m)$. Then, W_{bdp} is calculated as

$$W_{bdp} = RTT/(E[t^t] + E[t^r]/m). \tag{12}$$

Then, $E[r_{ss}]$ can be obtained simply by replacing the term of maximum window size of the equation derived in [25] with W_{bdp}, $E[r_{ss}]$ is calculated as

$$E[r_{ss}] = \begin{cases} \log_2\left(\frac{E[d_{ss}]}{w_1} + 1\right) & \text{if } E[W_{ss}] \leqslant W_{bdp} \\ \log_2\left(\frac{W_{bdp}}{w_1}\right) + 1 & \\ \quad + \frac{1}{W_{bdp}}\left(E[d_{ss}] - 2W_{bdp} + w_1\right) & \text{otherwise} \end{cases} \tag{13}$$

where w_1 is the initial window size.

We then calculate $E[r_{td}]$. Let $E[W]$ be the expected window size when a packet loss event occurs in the TD period. Then, the number of packets sent per round is dependent on the relation between $E[W]$ and W_{bdp}, thereby changing $E[r_{td}]$. Therefore, we obtain $E[r_{td}]$ by dividing into three cases: $E[W] \leqslant W_{bdp}$, $W_{bdp} < E[W] \leqslant 2W_{bdp}$, and $2W_{bdp} < E[W]$. When $E[W] \leqslant W_{bdp}$, $E[r_{td}] = (E[W]/2 + 1)$ [24]. When $W_{bdp} < E[W] \leqslant 2W_{bdp}$, the expected number of packets sent per round increases from $E[W]/2$ up to W_{bdp}, and after that, it is fixed to W_{bdp} until the TD period ends. On the other hand, when $2W_{bdp} < E[W]$, it is constantly W_{bdp} for all rounds. Following calculation processes [24], $E[r_{td}]$ is calculated as

$$E[r_{td}] = \begin{cases} \frac{E[W]}{2} + 1 & \text{if } E[W] \leqslant W_{bdp} \\ \frac{1-p}{pW_{bdp}} + \frac{E[W](E[W]+8)}{8W_{bdp}} + \frac{W_{bdp}-E[W]}{2} + \frac{3}{4} & \text{if } W_{bdp} < E[W] \leqslant 2W_{bdp} \\ \frac{1-p}{pW_{bdp}} + \frac{E[W]}{W_{bdp}} + \frac{3}{4} & \text{otherwise.} \end{cases} \tag{14}$$

4.2.2. Energy consumption of packet transmission and reception. Let $E[n^t]$ and $E[n^r]$ denote the expected number of packets sent and received during the data transfer of S_d bytes, respectively. Then, $E[J^t]$ and $E[J^r]$ are obtained as

$$E[J^t] = E[n^t]E[e^t], \quad E[J^r] = E[n^r]E[e^r]. \tag{15}$$

For simplicity, the number of retransmitted packets is not counted. Then, $E[n^t]$ is given by S_d/S_p. On the other hand, the number of received packets equals that of received ACK packets, which depends on timing of the occurrence of packet loss events. Therefore, $E[n^r]$ is calculated as follows. Using $E[n^r_\zeta]$ and $E[n^r_\eta]$, $E[n^r]$ is calculated as $E[n^r] = E[n^r_\zeta] + E[n^r_\eta]$.

We next determine $E[n^r_\zeta]$. In the initial slow start phase, $E[d_{ss}]$ data packets are transmitted. When packet loss events are detected by triple duplicate ACK packets, we assume that $E[W_{ss}]/2$ of $E[d_{ss}]$ data packets are lost averagely[¶]. In contrast, when packet loss events are detected by RTOs, $E[W_{ss}]$ data packets are lost. When m data packets are sent in a bursty fashion at a time by using the delayed ACK, the TCP sender receives an ACK packet every m data packets sent from it. From the aforementioned discussion, in a similar way to the calculation process for Equation (43), $E[n^r_\zeta]$ is obtained as

$$E[n^r_\zeta] = \left(E[d_{ss}] - l_{ss}(1 + Q(E[W_{ss}], p))\frac{E[W_{ss}]}{2}\right)/m. \tag{16}$$

[¶]This assumption follows [24].

DOI: 10.1002/dac

Similarly, $E[n^r_\eta]$ is obtained as follows. $E[W]/2$ packets of $\left(\frac{1-p}{p} + E[W]\right)$ packets sent during the TD period are lost averagely. Thus, $\left(\frac{1-p}{p} + E[W]/2\right)$ ACK packets are received in the TD period. According to the calculation process for Equation (53), $E[n^r_\eta]$ is calculated as

$$E[n^r_\eta] = E[n_{cycle}]E[n]\left(\frac{1-p}{p} + \frac{E[W]}{2}\right)\frac{1}{m} \tag{17}$$

where $E[n_{cycle}]$ is the expected number of cycles in the steady phase and $E[n]$ is the expected number of TD periods in a cycle.

4.2.3. Energy consumption during idle period. $E[J^l_{cam}]$ and $E[J^l_{sleep}]$ are calculated by multiplying P^l by the expected length of idle period during the whole data transfer, which is obtained by subtracting the expected period of the other states from the expected period of the whole data transfer, $E[T^{all}]$. As a result, $E[J^l_{cam}]$ and $E[J^l_{sleep}]$ are obtained as

$$E[J^l_{cam}] = P^l\left(E[T^{all}] - E[T^t] - E[T^r]\right), \tag{18}$$

$$E[J^l_{sleep}] = P^l\left(E[T^{all}] - E[T^t] - E[T^r] - E[T^s] - E[T^{st}]\right) \tag{19}$$

Note that $E[T^s]$ and $E[T^{st}]$ are derived in Section 4.2.4 and 4.2.5, respectively.

$E[T^{all}]$, $E[T^t]$ and $E[T^r]$ are obtained as follows. $E[T^{all}]$ is derived as

$$E[T^{all}] = RTT \cdot \left\{E[r_\xi] + E[r_\eta]\right\}. \tag{20}$$

In a similar form to Equation (15), $E[T^t]$ and $E[T^r]$ are written by

$$E[T^t] = E[n^t]E[t^t], \quad E[T^r] = E[n^r]E[t^r]. \tag{21}$$

4.2.4. Energy consumption during sleep period. In this subsection, we derive $E[J^s]$ based on a bottom-up approach. Figure 4 represents the packet sequence of transmission and reception, and state transitions of WNI in one round when $w_k = 5$ where w_k is the congestion window size of the kth round in either the initial slow start phase or the TD period of the steady phase. Note that sleeping with $m = 1$ means sleeping without burst transmission. In an interarrival time of ACK packets that arrive at the WNI of the client, m data packets are sent and a single ACK packet is received. Then, the idle period is obtained by subtracting packet transmission and reception times from the interarrival time. Furthermore, subtracting the time for state transitions between active and sleep modes, we can obtain sleep period in the interarrival time of ACK packets.

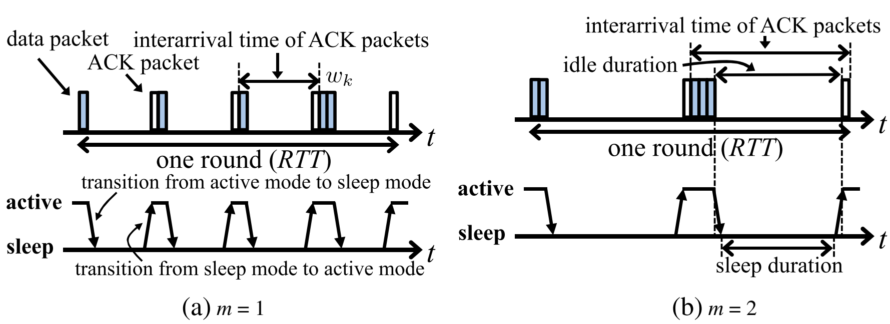


Figure 4. Packet sequence of transmission and reception and state transitions of wireless network interface where $w_k = 5$.

DOI: 10.1002/dac

We first consider an interarrival time of ACK packets. With the delayed ACK, the number of interarrivals of ACK packets in the kth round is $\lceil w_{k-1}/m \rceil$. Assuming ACK packets in a window are arrived at the client at equal intervals, an interarrival time of ACK packets, $t^a(w_k)$, is obtained as

$$t^a(w_k) = RTT / \lceil w_{k-1}/m \rceil . \quad (22)$$

Here, w_k for the data transfer of the initial slow start phase and for the TD period are denoted by w_k^{ss} and w_k^{ca}, respectively, which are given by

$$w_k^{ss} = \begin{cases} w_1 & \text{if } k = 0 \\ 2^{k-1} w_1 & \text{otherwise,} \end{cases} \quad (23)$$

$$w_k^{ca} = \begin{cases} E[W]/2 & \text{if } k = 0 \\ E[W]/2 + k - 1 & \text{otherwise.} \end{cases} \quad (24)$$

We now determine the expected sleep period at one interarrival time of ACK packets. Note that we consider that the WNI can sleep only when the time for state transitions between active and sleep modes is longer than the idle period at an interarrival time of ACK packets. As shown in Figure 4, m data packets are sent and an ACK packet is received at an interarrival time of ACK packets except for the last interarrival time of ACK packets. For simplicity, we regard the number of packets sent in the last interarrival time of ACK packets in one round as the same as that in the other interarrival times of ACK packets. As a result, the expected period that the WNI sleeps at an interarrival time of ACK packets, $E[t^s(w_k)]$, is calculated as

$$E[t^s(w_k)] = \left[t^a(w_k) - mE[t^t] - E[t^r] - T^{as} - T^{sa}\right]^+ \quad (25)$$

where $x^+ = \max(0, x)$.

To derive the expected sleep period of each phase, we determine the expected sleep period of data transfer and that of packet loss events. Using Equation (25), we derive the expected sleep period from the first round to the round in which a packet loss event occurs. Because the number of interarrivals of ACK packets in the kth round is $\lceil w_{k-1}/m \rceil$, the expected sleep period from the first to the rth rounds, $E[t_d^s(r, w_k)]$, is obtained as

$$E\left[t_d^s(r, w_k)\right] = \sum_{k=1}^{r} \left\lceil \frac{w_{k-1}}{m} \right\rceil E[t^s(w_k)] \quad (26)$$

where $r = E[r_{ss}]$ for the initial slow start phase and $r = E[r_{td}]$ for the TD period.

In the round in which a packet loss event is detected, the number of ACK packets received at the client is dependent on which packet in the window is actually lost. When a packet loss event is detected by triple duplicate ACK packets, in average, the idle period of $RTT/2$ occurs after $\lceil w_r/m \rceil /2$ ACK packets are received. Therefore, $E[t_{td}^s(r, w_k)]$ is calculated as

$$E\left[t_{td}^s(r, w_k)\right] = \frac{\lceil w_r/m \rceil}{2} E[t^s(w_{r+1})] + [RTT/2 - T^{as} - T^{sa}]^+ . \quad (27)$$

On the other hand, during the duration of a sequence of RTOs, $E[R]$ data packets are sent, whereas no ACK packet is received. We here assume that WNI can always sleep at idle period of the round because the idle period is larger enough than $(T^{as} + T^{sa})$. Then, $E[t_{to}^s]$ is given by

$$E[t_{to}^s] = RTT \cdot E[r_{to}] - E[R]\left(E[t^t] + T^{as} + T^{sa}\right). \quad (28)$$

DOI: 10.1002/dac

Using Equations (26)–(28), we derive $E[t^s_\xi]$ and $E[t^s_\eta]$. In a similar way to the calculation in Equation (43), $E[t^s_\xi]$ is obtained as

$$E\left[t^s_\xi\right] = E\left[t^s_d\left(r_{ss}, w^{ss}_k\right)\right] + l_{ss}\left((1 - Q(E[W_{ss}], p))E\left[t^s_{td}\left(r_{ss}, w^{ss}_k\right)\right] + Q(E[W_{ss}], p)E[t^s_{to}]\right). \tag{29}$$

Similarly, in the same manner as the calculation in Equation (53), $E[t^s_\eta]$ is calculated as

$$E[t^s_\eta] = E[n_{cycle}] \times \left\{E[n]\left(E[t^s_d(r_{td}, w^{ca}_k)] + E[t^s_{td}(r_{td}, w^{ca}_k)]\right) + E[t^s_{to}]\right\}. \tag{30}$$

Consequently, $E[J^s]$ and $E[T^s]$ are obtained as

$$E[J^s] = P^s \cdot E[T^s], \quad E[T^s] = E[t^s_\xi] + E[t^s_\eta]. \tag{31}$$

4.2.5. Energy consumption due to state transitions. In a similar way to Section 4.2.4, we determine $E[J^{st}]$ based on a bottom-up approach. From the condition, the WNI sleeps during the idle period discussed in Section 4.2.4, $E[n^{st}(w_k)]$ is given by

$$E[n^{st}(w_k)] = \begin{cases} 1 & \text{if } t^a(w_k) > mE[t^t] + E[t^r] + T^{as} + T^{sa} \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

Then, $E[n^{st}_d(r, w_k)]$ is calculated as

$$E[n^{st}_d(r, w_k)] = \sum_{k=1}^{r} \left\lceil \frac{w_{k-1}}{m} \right\rceil E[n^{st}(w_k)] \tag{33}$$

On the other hand, $E[n^{st}_{td}(r, w_k)]$ is obtained as

$$E[n^{st}_{td}(r, w_k)] = \begin{cases} \frac{\lceil w_r/m \rceil}{2} E[n^{st}(w_r)] + 1 & \text{if } RTT/2 > T^{as} + T^{sa} \\ \frac{\lceil w_r/m \rceil}{2} E[n^{st}(w_r)] & \text{otherwise.} \end{cases} \tag{34}$$

In contrast, because the state transitions occur $E[R]$ times during the duration of a sequence of RTOs, $E[n^{st}_{to}]$ is calculated as

$$E[n^{st}_{to}] = E[R] \tag{35}$$

Let $E[n^{st}_\xi]$ and $E[n^{st}_\eta]$ denote the expected numbers of state transitions during the initial slow start phase and during the steady phase, respectively. Following the calculation for Equation (43) and (53), using Equations (33)–(35), $E[n^{st}_\xi]$ and $E[n^{st}_\eta]$ are derived as follows:

$$E\left[n^{st}_\xi\right] = E\left[n^{st}_d(r_{ss}, w^{ss}_k)\right] + l_{ss}\left((1 - Q(E[W_{ss}], p))E\left[n^{st}_{td}(r_{ss}, w^{ss}_k)\right] + Q(E[W_{ss}], p)E\left[n^{st}_{to}\right]\right), \tag{36}$$

$$E[n^{st}_\eta] = E[n_{cycle}] \times \left\{E[n]\left(E\left[n^{st}_d\left(r_{td}, w^{ca}_k\right)\right] + E\left[n^{st}_{td}\left(r_{td}, w^{ca}_k\right)\right]\right) + E\left[n^{st}_{to}\right]\right\}. \tag{37}$$

Using Equations (36) and (37), $E[J^{st}]$ and $E[T^{st}]$ are calculated as

$$E[J^{st}] = \left(E\left[n^{st}_\xi\right] + E\left[n^{st}_\eta\right]\right)\left(P^{as}T^{as} + P^{sa}T^{sa}\right), \tag{38}$$

DOI: 10.1002/dac

$$E[T^{st}] = \left(E\left[n_{\xi}^{st}\right] + E\left[n_{\eta}^{st}\right]\right)(T^{as} + T^{sa}). \tag{39}$$

4.2.6. Increase in data transfer latency by burst transmission. In this subsection, we consider a disadvantage of burst transmission—the increase in the data transfer latency. For simplicity, we assume that the expected window size is less than W_{bdp} and that the duration of a sequence of RTOs is ignored because they have little influence on the increase in latency with burst transmission. With delayed ACK, the TCP receiver does not send an ACK packet until m data packets are received or the delayed ACK timer expires. Because of this waiting period, burst transmission achieved by the delayed ACK increases the latency in the whole data transfer.

Assuming that data packets sent in a bursty fashion are received at the TCP receiver at equal intervals due to the background traffic in the wired network, the RTT observed at the TCP sender is increased averagely by $(m-1)RTT/(\frac{1}{r}\sum_{k=1}^{r} w_k)$. Consequently, $E[\hat{T}^{all}](m)$ is obtained as

$$E[\hat{T}^{all}](m) = \left(RTT + (m-1)\frac{RTT}{E[r_{ss}]w_1}\left(2 - \frac{1}{2^{E[r_{ss}]-1}}\right)\right)E[r_{\xi}] + \left(RTT + (m-1)\frac{4RTT}{3E[W]}\right)E[r_{\eta}]. \tag{40}$$

5. MODEL VALIDATION

In this section, to confirm the accuracy of our model, we compare results of analysis with that of simulation experiments with ns-3 [29]. It is necessary to validate our model in a moderately congested WLAN because the proposed method has a modest effect on energy efficiency in a heavily congested WLAN due to a lack of idle periods. When congestion is moderate in a WLAN, the behavior of frame exchanges in the WLAN is well-modeled by ns-3 [30]. Because of this, the ns-3 simulation has sufficient accuracy to validate our model. Of course, in order to confirm the effectiveness of our proposal in real environments, experiments with real WLAN devices are needed, which are left for future work.

In addition, it is difficult to compare our model with existing models due to difference of assumptions. One of the main aims of the paper is understanding energy efficiency when a WNI sleeps at packet interarrivals, especially by focusing on impacts on the behavior of TCP congestion control mechanisms. Therefore, we assume ideal sleep mode in which a WNI sleeps and wakes up with exact timing, not specific sleep mechanisms such as PSM in IEEE 802.11. Because existing models [19–21] assume the PSM as a sleep mechanism, we cannot correctly compare our model with other models.

The energy efficiency of the proposed method is dependent on the total sleep period and the number of state transitions between active and sleep modes. This implies that we can confirm the accuracy of our model by comparing interarrival time of ACK packets of our model and true values. Therefore, we focus on the distribution of interarrival time of ACK packets at TCP of the client in the results of our model and simulations.

5.1. Parameter settings of simulation and analysis

We conduct simulations with ns-3.19. We assume a WLAN environment depicted in Figure 5. In Figure 5, some hosts are added to the network in Figure 1 to control the background traffic in the wired network and in the WLAN. In the WLAN, six wireless clients associates the AP. One of them sends data to the wired host by TCP and it keeps sending data during the simulation. The other clients sends 1 Mbps of constant bit rate traffic as background traffic in the WLAN; the total rate of background traffic in the WLAN is 5 Mbps. The link between routers A and B has 25 Mbps of physical bandwidth and 50 ms of one-way delay. By sending UDP constant bit rate traffic from the UDP sender to the UDP receiver, we control the congestion level of the link at the wired network so that it becomes the bottleneck of the TCP connection. One-way delays of all links in the wired network except for the bottleneck link are set to small enough.

DOI: 10.1002/dac

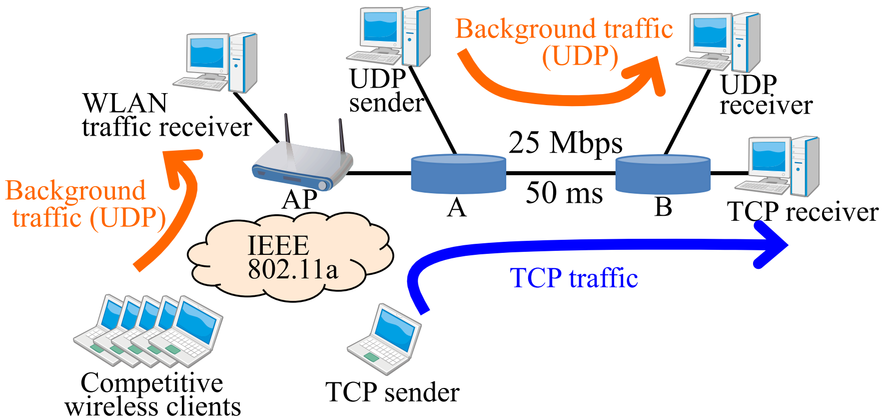


Figure 5. Wireless LAN environment for ns-3 simulation.

We set the probability (q) of transmission failures at MAC level and the maximum number (N) of frame retransmissions at MAC-level to 0.3, respectively, which are intended for a moderate loss rate at TCP-level.

We now determine average RTT and the probability (p) of packet loss events because our model requires both of them. Because the one-way delay of the bottleneck link is 50 ms, we set RTT to 100 ms for our model. p is calculated by solving Equation (50) for p. Now, giving the expected TCP throughput $E[R_{tcp}]$ Mbps, $E[W]$ can be calculated

$$E[W] = \frac{E[R_{tcp}] \cdot RTT \times 1024^2}{8 \times 1500}. \tag{41}$$

Note that $E[R_{tcp}]$ is the available bandwidth at the bottleneck link. $E[R_{tcp}]$ is calculated by subtracting the rate of background traffic from the bandwidth of bottleneck link, for example, $E[R_{tcp}]$ is 10 Mbps for 15 Mbps background traffic.

5.2. Validation results

Figure 6 shows the cumulative distribution function of interarrival time of ACK packets for the various values of m, which is the number of packets sent in a bursty fashion, when the available bandwidth at the bottleneck link is 10 Mbps. Note that regardless of the rate of background traffic in the wired network, the overall tendencies are almost identical. In the figures, we plot simulation results with and without background traffic in the WLAN for comparison purposes.

From Figure 6, we observe that simulation results with and without background traffic in the WLAN are relatively identical. This result implies that the behavior of the other clients has little effect on that of the TCP sender when the congestion level in the WLAN is moderate load.

We can also see in Figure 6 that, regardless of the values of m, the distribution of interarrival time of ACK packets obtained by the analysis is reasonably well matched at that by the simulation. However, our model slightly overestimates interarrival time of ACK packets. In our model, we assume that ACK packets arrives at the client with equal intervals in a RTT, whereas in the simulation, the packet intervals are apt to be compressed after the congestion window size decreases. Thus, the interarrival time of ACK packets in the simulation are distributed to the smaller than that in the analysis.

In order to assess the effects of the available bandwidth at the wired network, we present the average interarrival time of ACK packets, which are calculated from the distribution of interarrival time of ACK packets, for a variety of the available bandwidth at the bottleneck link in Table IV, when $m = 2$. Note that the tendencies of results are almost identical regardless of the

DOI: 10.1002/dac

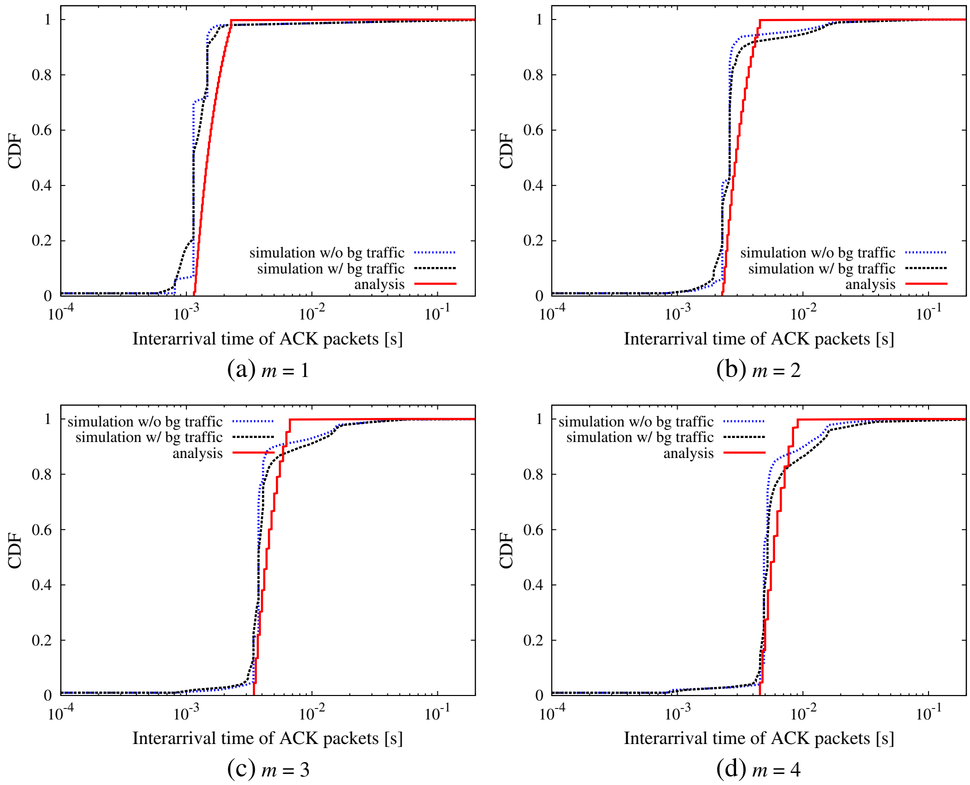


Figure 6. Distribution of interarrival time of ACK packets where the available bandwidth is 10 Mbps.

Table IV. Average interarrival time of ACK packets for simulation and analysis at a variety of the available bandwidth.

Available bandwidth [Mbps]	Avg. interarrival time of ACKs [ms]			Relative error [%]	
	Analysis	Simulation w/o b.g. traffic	Simulation w/ b.g. traffic	w/o b.g. traffic	w/ b.g. traffic
1	36.0	33.4	32.8	-7.89	-9.72
2	16.9	15.7	16.6	-8.06	-1.84
5	6.42	6.89	6.76	6.76	5.05
8	3.95	4.25	3.88	7.15	-1.76
10	3.12	3.18	3.45	2.01	9.56
12	2.57	2.71	2.74	5.18	5.92
15	2.06	2.52	2.30	18.2	10.5
18	1.71	1.81	1.91	5.51	10.3

values of m. Compared with simulation results, the relative error of the average interarrival times of analysis is less than 10% in many cases. The error is caused by the occurrence of consecutive packet losses. Our analysis model assumes that packet loss events occur independently on each other, whereas in the simulation packet loss events are apt to occur repeatedly. This difference affects the growth of the TCP congestion window, resulting in the gap between analysis and simulation results.

From the aforementioned discussion, we conclude that our model is reasonably validated when the congestion level is moderate in the WLAN.

DOI: 10.1002/dac

6. DISCUSSIONS WITH ANALYSIS RESULTS

In this section, we assess the energy efficiency of TCP data transfer with and without burst transmission by means of the energy consumption model described in Section 4.

6.1. Parameter settings and evaluation metrics

We consider TCP data transfer of a 10 MB file from the wireless client to the wired host in Figure 1, by using an IEEE 802.11a WLAN. The WLAN parameters of IEEE 802.11a are summarized in Table V.

To calculate radio propagation delay in Equations (7) and (8), we assume that the wireless client is located 4 m from the AP. From a data sheet for a WNI implemented by the Atheros AR5004 chip [31], we set P^t, P^r, P^l, and P^s to the values shown in Table VI. Following Krashinsky and Balakrishnan [5], who measured power consumption in a specific WNI to determine consumption during transition from active to sleep modes and vice versa, we set $P^{as} = P^l$ and $P^{sa} = P^t$. Moreover, T^{as} and T^{sa} are set equal to 1 μs and 1 ms, respectively, in accordance with Andren *et al.* [32]. The TCP data and TCP ACK packet sizes are set equal to 1500 bytes and 40 bytes, respectively. Unless otherwise noted, we set q to 0.3, which means we assume a moderate frame loss rate.

In Section 6.2, we evaluate energy efficiency and trade-off relationships between energy efficiency and TCP latency in TCP burst transmission. To assess the energy efficiency, we use two metrics: *energy consumptions*, which are given by Equations (10) and (11), and *energy consumption ratio*, which is defined as

$$R_{energy} = E[J_{sleep}]/E[J_{cam}]. \qquad (42)$$

On the other hand, to evaluate the trade-off relationships, we use *TCP transfer latency*, which is obtained by Equation (40), in addition to energy consumption ratio.

6.2. Numerical results

6.2.1. Fundamental characteristics of sleeping. To assess the maximum potential energy saving is attained by sleeping, we first show energy consumption with CAM and with sleeping without burst

Table V. WLAN parameters.

Name	Value
Data rate	54 Mbps
Slot time	9 μs
SIFS	16 μs
DIFS	34 μs
N	7
PLCP preamble	16 μs
MAC header	24 bytes
LLC header	8 bytes
CW_{min}	15
CW_{max}	1023

MAC, media access control.

Table VI. Power consumption of Atheros AR5004 [31].

P^t	P^r	P^l	P^s
1.4 W	0.9 W	0.8 W	0.016 W

DOI: 10.1002/dac

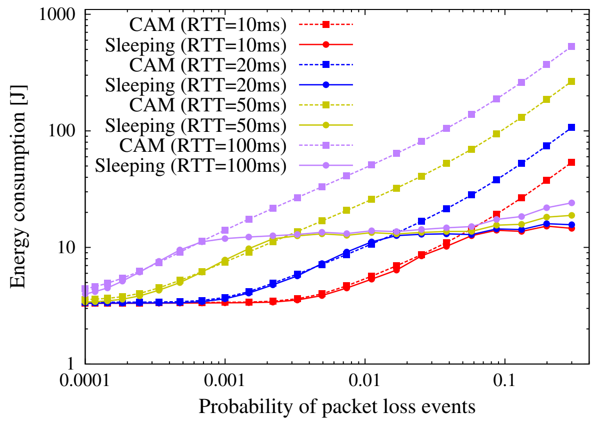


Figure 7. Energy consumption for the complete data transfer.

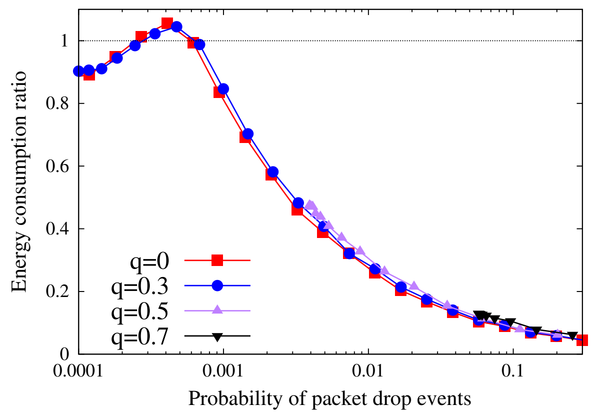


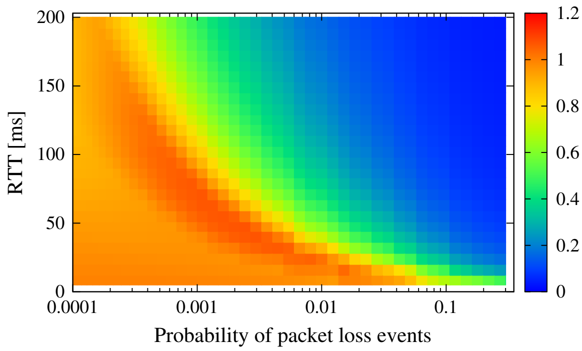
Figure 8. Energy consumption ratio for varying q when $RTT = 100$ ms.

transmission ($m = 1$) in Figure 7. The x-axis in this figure denotes the probability (p) of the packet loss events. We plot the results with various values of RTT.

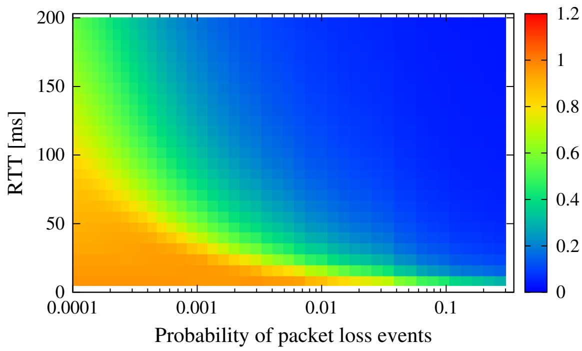
We see that energy consumption decreases as RTT decreases regardless of whether or not sleeping is utilized. This is due to the transfer latency for the whole data becoming small as RTT decreases. Comparing the results for CAM and for sleeping, when p is small, the energy consumptions with CAM and with sleeping are almost identical. For small p, the average congestion window size becomes large, thereby shortening the idle periods at packet interarrivals. Consequently, the WNI cannot sleep at the idle periods. In contrast, for large p, energy consumption with sleeping becomes significantly smaller than that with CAM because the WNI can sleep at all idle periods that occupies a large portion of the transfer latency.

In order to assess the impact on energy consumption due to packet losses at the WLAN, we compare the energy consumption for varying the probability (q) of frame transmission failures at MAC-level. To this end, we depict the energy consumption ratios as a function of p for varying q from 0 to 0.7 in Figure 8, when $RTT = 100$ ms. Note that $q = 0$ means the situation in which packet loss events occur only in the wired network, and that we change p_l to control the value of p. We see that energy consumption ratio with a certain value of p increases slightly as q increase. As q increases, the expected time in which one data frame is sent or received becomes longer due to increased frame retransmissions, resulting in the shorter sleep periods at packet interarrivals. However, the increase in energy consumption due to the shortened sleep periods have a small portion of the total energy consumption. Therefore, the increase in q has a small impact on the sleep efficiency.

DOI: 10.1002/dac



(a) $m = 1$



(b) $m = 5$

Figure 9. Energy efficiency as function of both round trip time and probability of packet loss events.

From the aforementioned results, we conclude that energy consumption is largely affected by RTT and p, which are dominantly determined by wired part of the network. This implies the importance of the analysis of both of wired and wireless parts of the network path to understand the energy efficiency.

6.2.2. Energy efficiency of burst transmission. Figure 9 shows the distribution of the energy consumption ratio as functions of RTT and p with $m = 1$ (Figure 9(a)) and $m = 5$ (Figure 9(b)).

Comparing Figures 9(a) and 9(b), the energy consumption ratio is reduced for the most part with the introduction of burst transmission. In particular, energy efficiency improves significantly with burst transmission when RTT or p is small. When RTT or p is small, energy consumption due to state transitions account a relatively large portion of the total energy consumption. The burst transmission can reduce the number of state transitions, thereby reducing the energy consumed due to state transitions.

Figure 10 shows the energy consumption ratio as a function of p for various values of m, when $RTT = 100$ ms. When $p = 0.001$, sleeping with $m = 1$ can reduce only about 10 % of energy consumption with CAM, whereas sleeping with $m = 5$ can reduce further 50%. Although energy efficiency increases as m increases, the additional energy saving diminishes, implying that good energy efficiency can be obtained with a small value of m.

6.2.3. Trade-off relationships between energy efficiency and data transfer latency. To evaluate trade-off relationships between energy efficiency and data transfer latency, we present the energy efficiency and transfer latency as a function of m in Figures 11 and 12, respectively. We can observe that the latency increases linearly while the energy efficiency converges to a constant. When RTT is large, energy efficiency is high even without burst transmission and it increases slightly as m increases, whereas the transfer latency increases largely compared with that without

DOI: 10.1002/dac

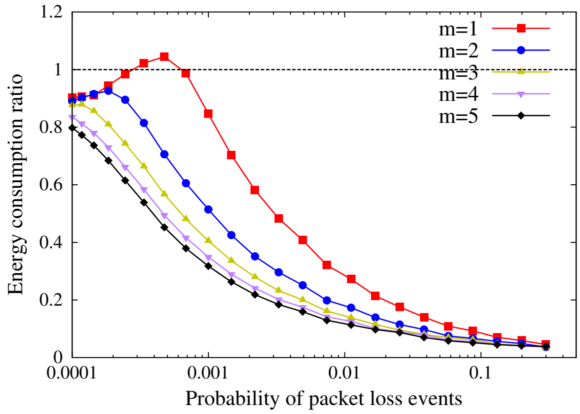


Figure 10. Energy efficiency of burst transmission when $RTT = 100$ ms.

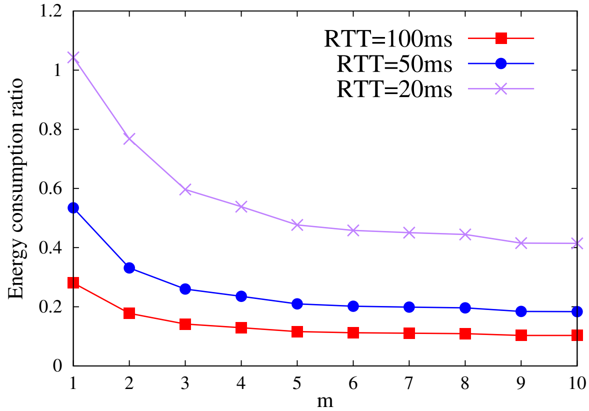


Figure 11. Energy efficiency as a function of m when $p = 0.01$.

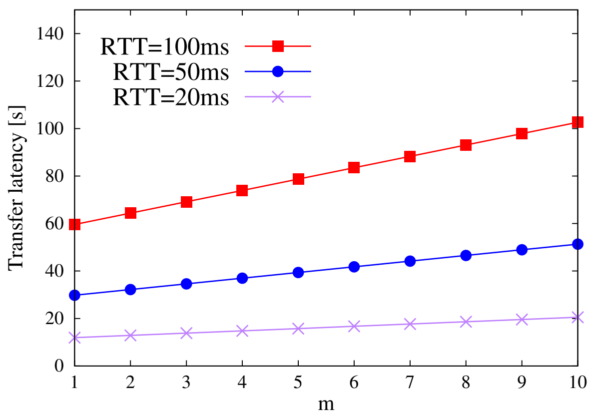


Figure 12. Transfer latency as a function of m when $p = 0.01$.

burst transmission ($m = 1$). Conversely, when RTT is small, energy efficiency improves significantly as m increases, whereas the transfer latency increase slightly from that without burst transmission. This result means that burst transmission is suitable for energy saving when RTT is small.

DOI: 10.1002/dac

7. CONCLUSION

In this paper, we have proposed new models for describing energy consumed in TCP data transfer over a WLAN, which focuses on the detailed behavior of TCP congestion control mechanisms. Furthermore, in order to improve the sleep efficiency, TCP-level burst transmission behavior was also proposed. Through the simulation, our model was reasonably validated when the congestion level is moderate in a WLAN.

From the numerical analyses based on our model, we obtained the following results. The sleep efficiency was largely affected by RTT and the probability of packet losses at transport-layer level, which were dominantly determined by wired part of the network. Sleeping with burst transmission was suitable for situations in which RTT or the probability of packet losses at transport-layer level are small, that is, TCP transmission rate is high. In such a situation, sleeping with burst transmission could significantly reduce energy consumption with increasing moderate delay. In contrast, when RTT or the probability of packet losses at transport-layer level were small, sleeping with burst transmission incurred large delays for TCP packets, although it achieves energy saving.

In the future, we plan to develop transport architecture for energy saving based on burst transmission at the transport-layer level.

APPENDIX A: DERIVATION OF NUMBER OF ROUNDS DURING TCP DATA TRANSFER

In what follows, we formulate $E[r_\zeta]$ and $E[r_\eta]$. We first determine $E[r_\zeta]$. Following [25], $E[r_\zeta]$ is calculated as

$$E[r_\zeta] = E[r_{ss}] + l_{ss}((1 - Q(E[W_{ss}], p)) + Q(E[W_{ss}], p)E[r_{to}]). \tag{43}$$

l_{ss}, $Q(w, p)$, and $E[W_{ss}]$ are derived as

$$l_{ss} = 1 - (1-p)^{S_d/S_p}, \tag{44}$$

$$Q(w,p) = \min\left(\frac{(1+(1-p)^3)(1-(1-p)^{w-3})}{(1-(1-p)^w)/(1-(1-p)^3)}, 1\right), \tag{45}$$

$$E[W_{ss}] = (E[d_{ss}] + w_1)/2. \tag{46}$$

$E[d_{ss}]$ is given by

$$E[d_{ss}] = \min\left\{\frac{(1-(1-p)^{S_d/S_p})(1-p)}{p} + 1, \frac{S_d}{S_p}\right\}. \tag{47}$$

Equations. (44)–(47) can be found in [25]. Using results in [25], $E[r_{to}]$ is obtained by

$$E[r_{to}] = \frac{G(p)}{1-p}\frac{T_0}{RTT} \tag{48}$$

where T_0 is the length of a RTO and $G(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6$.

We can determine $E[r_\eta]$ by utilizing the analysis results in [24]. From Figure 3, in one cycle of the steady phase, multiple TD periods occur before a TO period occurs. Therefore, the expected number of data packets sent during a cycle is given by $(E[n]E[Y] + E[R])$ where $E[Y]$ and $E[R]$ is the expected number of data packets sent during a TD period and during a TO period, respectively. $E[Y]$ and $E[R]$ are obtained as

$$E[Y] = \frac{1-p}{p} + E[W], \quad E[R] = \frac{1}{1-p}. \tag{49}$$

$E[W]$ and $E[n]$ are given by

DOI: 10.1002/dac

$$E[W] = 1 + \sqrt{8(1-p)/(3p) + 1}, \tag{50}$$

$$E[n] = 1/Q(E[W], p). \tag{51}$$

Equations (49) and (51) are found in [24].

Because the expected number of packets sent during the steady phase is $(S_d/S_p - E[d_{ss}])$, the number of cycles during the steady phase is calculated as

$$E[n_{cycle}] = (S_d/S_p - E[d_{ss}])/(E[n]E[Y] + E[R]). \tag{52}$$

Consequently, we can obtain the expected number of rounds during a cycle as $(E[n]E[r_{td}] + E[r_{to}])$. Using the aforementioned notations, $E[r_\eta]$ is calculated as

$$E[r_\eta] = E[n_{cycle}]\left(E[n]E[r_{td}] + E[r_{to}]\right). \tag{53}$$

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Internal Affairs and Communications, Japan, under Promotion Program for Reducing Global Environmental Load through ICT Innovation.

REFERENCES

1. Atheros Communications. Power consumption and energy efficiency comparisons of wlan products, 2003. In Atheros White Papers.
2. Raghunathan V, Pering T, Want R, Nguyen A, Jensen P. Experience with a low power wireless mobile computing platform. *Proceedings of ISLPED 2004,* ACM Press, New York, New York, USA, 2004; 363–368.
3. Agarwal Y, Schurgers C, Gupta R. Dynamic power management using on demand paging for networked embedded systems. *Proceedings of ASP-DAC 2005,* ACM Press, New York, New York, USA, January 2005; 755–759.
4. IEEE 802.11-2007. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE, 2007.
5. Krashinsky R, Balakrishnan H. Minimizing energy for wireless web access with bounded slowdown. *Wireless Networks* 2005; **11**(1-2):135–148.
6. IEEE 802.11e. Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: medium access control (MAC) quality of service enhancements, IEEE, 2005.
7. Camps Mur D, Pérez-Costa X, Sallent Ribes S. An adaptive solution for wireless LAN distributed power saving modes. *Computer Networks* 2009; **53**(18):3011–3030.
8. Lee JR, Kwon SW, Cho DH. Adaptive beacon listening protocol for a TCP connection in slow-start phase in WLAN. *IEEE Communications Letters* 2005; **9**(9):853–855.
9. He Y, Yuan R. A novel scheduled power saving mechanism for 802.11 wireless LANs. *IEEE Transactions on Mobile Computing* October 2009; **8**(10):1368–1383.
10. Ting KC, Lee HC, Lee HH, Lai F. An idle listening-aware energy efficient scheme for the DCF of 802.11n. *IEEE Transactions on Consumer Electronics* 2009; **55**(2):447–454.
11. Liu J, Zhong L. Micro power management of active 802.11 interfaces. *Proceedings of MobiSys 2008,* ACM Press, New York, New York, USA, 2008; 146–159.
12. Zhang X, Shin KG. E-MiLi: energy-minimizing idle listening in wireless networks. *IEEE Transactions on Mobile Computing* 2012; **11**(9):1441–1454.
13. Camps-Mur D, Gomony MD, Pérez-Costa X, Sallent-Ribes S. Leveraging 802.11n frame aggregation to enhance QoS and power consumption in Wi-Fi networks. *Computer Networks* 2012; **56**(12):2896–2911.
14. Kim T, Lee J, Cha H, Ha R. An energy-aware transmission mechanism for WiFi-based mobile devices handling upload TCP traffic. *International Journal of Communication Systems* 2009; **22**(5):625–640.
15. Namboodiri V, Gao L. Energy-efficient VoIP over wireless LANs. *IEEE Transactions on Mobile Computing* 2010; **9**(4):566–581.
16. Dogar FR, Steenkiste P, Papagiannaki K. Catnap: exploiting high bandwidth wireless interfaces to save energy for mobile devices. *Proceedings of MobiSys 2010,* ACM Press, New York, New York, USA, 2010; 107–122.
17. Yan H, Watterson SA, Lowenthal DK, Li K, Krishnan R, Peterson LL. Client-centered, energy-efficient wireless communication on IEEE 802.11b networks. *IEEE Transactions on Mobile Computing* 2006; **5**(11):1575–1590.
18. Yang Wh, Wang Yc, Tseng Yc, Lin BsP. Energy-efficient network selection with mobility pattern awareness in an integrated WiMAX and WiFi network. *International Journal of Communication Systems* 2010; **23**(2):213–230.

DOI: 10.1002/dac

19. Anastasi G, Conti M, Gregori E, Passarella A. Saving energy in Wi-Fi hotspots through 802.11 PSM : an analytical model. *Proceedings of ISCC 2004*, Alexandria, Egypt, March 2004; 466–471.
20. Ergen M, Varaiya P. Decomposition of energy consumption in IEEE 802.11. *Proceedings of ICC 2007*, Glasgow, Scotland, IEEE, June 2007; 403–408.
21. Agrawal P, Kumar A, Kuri J, Panda MK, Navda V, Ramjee R, Padmanabhani VN. Analytical models for energy consumption in infrastructure WLAN STAs carrying TCP traffic. *Proceedings of COMSNETS 2010*, Bangalore, India, IEEE, January 2010; 1–10.
22. Braden R. Requirements for internet hosts – communication layers. *Request for Comments 1122* 1989.
23. Allman M. On the generation and use of TCP acknowledgments. *ACM SIGCOMM Computer Communication Review* 1998; **28**(5):4–21.
24. Padhye J, Firoiu V, Towsley D, Kurose J. Modeling TCP throughput: a simple model and its empirical validation. *ACM SIGCOMM Computer Communication Review* 1998; **28**(4):303–314.
25. Cardwell N, Savage S, Anderson T. Modeling TCP latency. *Proceedings of INFOCOM 2000,* Tel Aviv, Israel, IEEE, Vol. 3, March 2000; 1742–1751.
26. Zorzi M, Rao RR. Energy efficiency of TCP in a local wireless environment. *Mobile Networks and Applications* 2001; **6**(3):265–278.
27. Vacirca F, De Vebductis A, Baiocchi A. Optimal design of hybrid FEC/ARQ schemes for TCP over wireless links with Rayleigh fading. *IEEE Transactions on Mobile Computing* 2006; **5**(4):289–302.
28. Dunaytsev R, Moltchanov D, Koucheryavy Y, Harju J. Modeling TCP SACK performance over wireless channels with completely reliable ARQ/FEC. *International Journal of Communication Systems* 2011; **24**(12):1533–1564.
29. ns-3. (Available from: http://www.nsnam.org/) [Accessed on April 2014].
30. Baldo N, Requena-Esteso M, Núñez Martínez J, Portolès-Comeras M, Nin-Guerrero J, Dini P, Mangues-Bafalluy J. Validation of the IEEE 802.11 MAC model in the ns3 simulator using the EXTREME testbed. *Proceedings of SIMUTools 2010*, March 2010; 64:1–64:9.
31. Wistron NeWeb Corp. *CM9: WLAN 802.11 a/b/g mini-PCI Module*. (Available from: http://site.microcom.us/CM9.pdf) [Accessed on Dec. 2012].
32. Andren C, Bozych T, Road B, Schultz D. *PRISM power management modes: Application note AN9665*, 1997.

DOI: 10.1002/dac