PAPER

SCTP Tunneling: Flow Aggregation and Burst Transmission to Save Energy for Multiple TCP Flows over a WLAN

Masafumi HASHIMOTO[†a)], ***Student Member***, **Go HASEGAWA**[†], ***and*** **Masayuki MURATA**[†], ***Members***

SUMMARY To raise the energy efficiency of wireless clients, it is important to sleep in idle periods. When multiple network applications are running concurrently on a single wireless client, packets of each application are sent and received independently, but multiplexed at MAC-level. This uncoordinated behavior makes it difficult to control of sleep timing. In addition, frequent state transitions between active and sleep modes consume non-negligible energy. In this paper, we propose a transport-layer approach that resolves this problem and so reduces energy consumed by multiple TCP flows on a wireless LAN (WLAN) client. The proposed method, called SCTP tunneling, has two key features: *flow aggregation* and *burst transmission*. It aggregates multiple TCP flows into a single SCTP association between a wireless client and an access point to control packet transmission and reception timing. Furthermore, to improve the sleep efficiency, SCTP tunneling reduces the number of state transitions by handling multiple packets in a bursty fashion. In this study, we construct a mathematical model of the energy consumed by SCTP tunneling to assess its energy efficiency. Through numerical examples, we show that the proposed method can reduce energy consumption by up to 69%.

key words: *transmission control protocol (TCP), stream control transmission protocol (SCTP), wireless LAN, energy efficiency*

1. Introduction

Owing to recent developments in wireless network technologies, the Internet is increasingly accessed by using mobile devices such as smartphones, laptops, and tablet PCs. Wireless communication accounts for a large portion of energy consumption in mobile devices. For example, wireless communication via a IEEE 802.11 standard wireless LAN (WLAN) is reported to account for up to 50% of the device's total energy consumption [1]–[4]. Therefore, there is a great deal of interest in reducing the energy consumed through wireless communication, particularly because most mobile devices are battery-driven. In the present paper, we focus on a WLAN environment since it has grown in popularity and consumes large energy compared with the other wireless technologies [4].

For energy saving in media access control (MAC) layer protocols, the IEEE 802.11 standard defines a power saving mode (PSM) [5], as opposed to the mode under normal operation, which is referred to as the continuously active mode (CAM). In CAM, the radio devices of a wireless client are constantly activated. Thus, while network performance is high, the energy efficiency is low. In contrast, a wireless client in PSM sleeps when data is not being transmitted or received and periodically wakes up to receive a beacon transmitted from an access point (AP). Although PSM can considerably reduce energy consumption, it can also degrade network performance such as throughput and latency [6].

In order to overcome this issue and to improve further energy efficiency, many researchers have proposed energy-efficient solutions for WLANs [6]–[15]. Some of these methods [6]–[12] achieve high energy efficiency by mainly modifying MAC protocols, whereas the others [13]–[15] are energy-efficient solutions for specific applications. In contrast, we aim to derive a generalized transport-layer solution for energy saving without modifying the applications or MAC protocols.

To maximize energy saving without degrading network performance characteristics, it is important to control a wireless network interface (WNI) of a wireless client to sleep and wake up at appropriate timing. To achieve this, the prediction of timing of packet transmissions and receptions is needed. However, it is difficult due to uncoordinated behavior of applications running concurrently on a single wireless client. In a typical environment where mobile devices are used, multiple TCP connections are established for such applications, resulting in the unpredictable packet transmissions and receptions at MAC-level. Another issue is that the WNI consumes extra energy when transiting active and sleep modes. Therefore, frequent state transitions caused by multiple TCP connections reduce sleep efficiency.

In the present paper, to overcome these issues, we propose SCTP tunneling, which is a transport-layer approach to save energy for TCP data transfer over a WLAN. The proposed method has two key features: *flow aggregation* and *burst transmission* at transport-layer level. In the proposed method, multiple TCP flows are aggregated into a single aggregate flow in order to control the sleep timing. Furthermore, packets from the aggregate flow are sent and received in a bursty fashion to save energy by reducing the number of state transitions. To this end, the proposed method exploits stream control transport protocol (SCTP) [16] that is a general-purpose transport-layer protocol for IP networks. An SCTP association is established between a wireless client and an access point (AP), and all packets of TCP flows at the wireless client are aggregated into the association by means of SCTP multistreaming.

We derive a power consumption model for SCTP tun-

Manuscript received October 29, 2012.

Manuscript revised June 18, 2013.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: m-hasimt@ist.osaka-u.ac.jp

DOI: 10.1587/transcom.E96.B.2615

neling to assess its energy-saving potential. This model is based on our previous energy consumption models for a single TCP flow in a WLAN [17], [18], which focus on both the frame exchanges of a IEEE 802.11 MAC and the detailed behavior of TCP congestion control mechanisms. From the numerical results of the current model, we demonstrate the energy efficiency of SCTP tunneling as functions of aggregate TCP throughput, network environment, and the degree of burst transmission. We also evaluate an increase in transmission delay by using SCTP tunneling.

The remainder of this paper is organized as follows. First, we show related work in Sect. 2, and then describe SCTP tunneling in Sect. 3. In Sect. 4, the proposed power consumption model for SCTP tunneling is introduced. Section 5 shows numerical analysis results for our model. Finally, conclusions and future research directions are presented in Sect. 6.

2. Related Work

In IEEE 802.11 PSM, wireless clients only wake up at the beacon interval, which is typically 100 [ms]. Due to this, PSM can achieve high energy efficiency while degrading network performance such as throughput and latency. To overcome this, IEEE 802.11e defines a new power saving mechanism, which is called automatic power save delivery (APSD) [7]. The main idea of unscheduled APSD (U-APSD), which is one of the power saving modes of APSD, is that a data frame sent from a wireless client, which is referred to as trigger frame, is treated as a request to transmit the buffered data frame at the AP. Raising the effectiveness of U-APSD requires a trigger generation algorithm, which is addressed by Mor et al. [8]. Other researchers have proposed energy-efficient solutions in WLANs by modifying MAC protocols or WNI hardware [6], [9]–[12]. The solutions in the references [6], [9]–[12] achieve energy efficiency on WLAN clients by modifying the MAC protocols and the WNI hardware, whereas our proposed method is the transport-layer solution, which can collaborate with the MAC-layer and physical-layer solutions.

Other solutions for energy saving of wireless clients focus on the behavior of the upper-layer protocols [13]–[15]. Yan et al. [13] presented a client-centered method in TCP over WLANs, with burst transmission realized by manipulating the TCP receiver's window size. Namboodiri and Gao [14] 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. [15] developed the Catnap proxy for data-oriented applications such as web browsing and file transfer. In terms of sleep granularity, the Catnap proxy allows wireless clients to remain in the sleep mode over the whole data transmission scale, whereas the client-centered method allows them to remain in the sleep mode over the round trip time (RTT) scale. With the recent developments in RF circuit design, transition time between active and sleep modes has become shorter [12]. Therefore, wireless clients with our proposed method can sleep at idle duration within one RTT, which can be changed as a function of the number of packets sent in a bursty fashion.

The SCTP multistreaming feature, which is utilized by our proposed method, is mainly used for improving network application performance [19], [20]. Ladha and Amer [19] showed that SCTP multistreaming improves FTP performance compared with TCP. Natarajan et al. [20] demonstrated that HTTP over SCTP has an advantage in terms of transfer latency under lossy networks compared with HTTP over TCP. In contrast, the proposed method utilizes the SCTP multistreaming feature to improve the sleep efficiency in the presence of multiple TCP flows at a wireless client.

In our previous work, we analyzed the sleep efficiency of a wireless client that has a single TCP connection [17], [18]. In [17], we presented an energy consumption model for TCP data transfer over a WLAN, which depends on both the frame exchanges of a IEEE 802.11 MAC and the detailed behavior of TCP congestion control mechanisms. By extending it to accommodate burst transmission at transport-layer level, we demonstrated that burst transmission could improve the sleep efficiency of a wireless client in [18]. In the present paper, we derive a power consumption model for SCTP tunneling for multiple TCP connections on a single WLAN client, based on these models.

3. SCTP Tunneling

Our proposed method, called SCTP tunneling, exploits some features of SCTP to improve energy efficiency. SCTP tunneling has two key features: flow aggregation and burst transmission, which are by means of SCTP multistreaming and delayed ACK mechanisms, respectively. Each is described in turn.

3.1 Flow Aggregation

SCTP is a connection-oriented transport protocol providing a service similar to TCP and has some advanced features to support increased application requirements. In contrast to the stream-oriented nature of TCP, SCTP is message-oriented. It means that user messages are added to DATA chunks in SCTP, and multiple SCTP-DATA chunks are used to construct an SCTP packet. Another feature of SCTP is multistreaming, which enables streams of user messages from multiple upper-layer applications to be multiplexed into a single SCTP association. Additionally, SCTP has the same congestion control mechanisms as TCP except that the use of selective acknowledgments (SACKs) in SCTP is mandatory.

In SCTP tunneling, an SCTP association is established between a wireless client and an AP, as shown in Fig. 1. All packets of multiple TCP flows (e.g., the three flows in Fig. 1) are sent by SCTP tunneling, and each TCP flow is distinguished as a single stream in the SCTP association through multistreaming. Note that SCTP tunneling can also be applicable to UDP flows. For simplicity, we assume that only

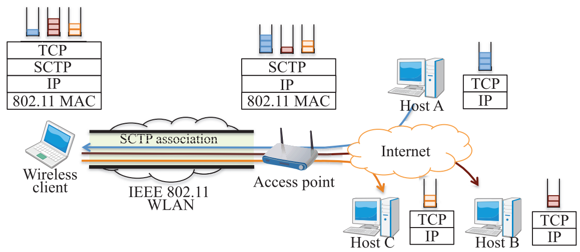
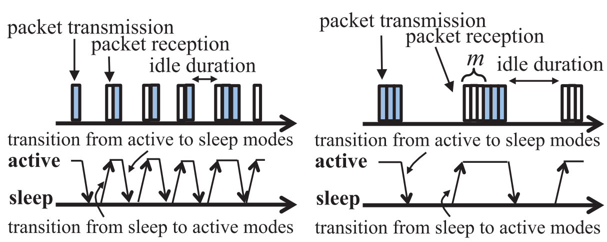


Fig. 1 SCTP tunneling in WLAN environment.



(a) Without burst transmission (b) With burst transmission

Fig. 2 Packet sequences at client's WNI during SCTP tunneling.

TCP flows are established on a client in this paper. A TCP packet generated in a wireless client is encapsulated in an SCTP-DATA chunk and enqueued in a transmission queue of the SCTP association. When a new SCTP packet can be transmitted, an SCTP-DATA chunk is dequeued from the transmission queue and is placed in a single SCTP packet. Transmission of the SCTP packet then obeys SCTP congestion control mechanisms. Once the SCTP packet is received by an AP, the packet is decapsulated and the original TCP packet is forwarded to its destination. At this time, SCTP at the AP generates an SCTP-SACK chunk to acknowledge receipt of the SCTP-DATA chunk. This SCTP-SACK chunk may piggyback with other SCTP-DATA chunks to the client. Data transmission from the AP to the client is conducted in a similar way to the above sequence.

By this method multiple TCP flows are aggregated on a single wireless client, and thus the transmission and reception timing of multiple TCP packets can be controlled.

3.2 Burst Transmission

To reduce energy consumed due to state transitions, SCTP tunneling employs burst transmission of SCTP packets by means of the delayed ACK mechanism [21] applicable to SCTP. Figure 2 shows packet sequences of a client's WNI during SCTP tunneling with and without burst transmission. If m SCTP packets are sent by burst transmission, these packets are transmitted and received consecutively by setting the delayed ACK parameter to m. In this case, once an SCTP has received m SCTP packets, in which the last packet contains an SCTP-SACK chunk, new m packets can be sent simultaneously. Upon receiving the SCTP packets including the SCTP-SACK chunk, another SCTP can consecutively send m new SCTP packets. By this mechanism, burst transmission can be realized. Note that an SCTP-SACK chunk piggybacks with an SCTP-DATA chunk in the mth SCTP packet. When the delayed ACK timer has expired, an SCTP packet containing an SCTP-SACK chunk is transmitted immediately. In the proposed method, we assume that the wireless client informs an AP of the value of m when establishing an SCTP association.

SCTP tunneling thus enables a wireless client to save energy by sleeping during the idle periods lengthened by burst transmission. In practical cases, SCTP tunneling is used by combining it with existing sleep mechanisms at MAC-level such as PSM and U-APSD.

4. Power Consumption Model

In this section, we construct a power consumption model for SCTP tunneling. This model consists of two parts: a MAC-level submodel presented in Sect. 4.2 and an SCTP-level submodel in Sect. 4.3. The assumptions for deriving these models are first described in Sect. 4.1, after which the submodels are outlined. Then, in Sect. 4.4, the expected size of SCTP packets is determined, which is needed in the MAC-level submodel. Finally, we formulate the increase in transmission latency due to buffering delay of SCTP tunneling in Sect. 4.5.

4.1 Assumptions

The environment here is a WLAN in which a single SCTP association is established between a wireless client and an AP. Multiple TCP upstream and downstream flows are established in the wireless client by upper-layer applications. We assume that the average throughputs of TCP flows are given, which are usually dependent on network conditions, e.g., network congestion and physical bandwidth between two end hosts of each flow.

We also assume that data frames are lost randomly at MAC-level of the WLAN due to channel error and collisions. The probability of transmission failures at MAC-level is given. In addition, RTS/CTS mechanisms are used by the wireless client when transmitting a frame to an AP, whereas AP does not utilize RTS/CTS when transmitting a frame to the wireless client.

Suppose that at the hardware level the WNI has four communication modes — *transmit*, *receive*, *idle* or *listen*, and *sleep* modes [1]. Each of these modes has a different power consumption denoted by P^t, P^r, P^l, and P^s, respectively. 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 duration of them is then denoted by T^{as} and T^{sa}, respectively.

We assume that SCTP tunneling uses *ideal sleeping* instead of PSM and U-APSD to assess energy-saving potential by it. Ideal sleeping 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. Note that in practical cases we can control the sleep timing by using SCTP tunneling together with existing sleep mechanisms. For instance, suppose we use U-APSD as a sleep mechanism. In U-APSD, all data frames destined for wireless clients are buffered in the AP. The clients can receive the buffered frames by sending a trigger at arbitrary timing, which means that they can control their sleep timing and duration. Although trigger transmission timing can be estimated based on the numbers of packet transmissions and receptions, the estimation error causes a waste of energy and increase in delay. Trigger transmission algorithms are left for future work.

Finally, although values of the delayed ACK timer may affect performance, this effect is not considered here. Note that when the delayed ACK timer expires, SCTP-SACK packets are delayed up to the value of the timer, which result in longer RTT of the corresponding SCTP packets.

4.2 Energy Consumption during Frame Exchanges for IEEE 802.11 MAC

We first derive the expected energy consumed for a single data frame sent and received by a wireless client, which are defined as $E[J^t]$ and $E[J^r]$, respectively. In our previous work [17], [18], we formulated an energy consumption model for frame exchanges in IEEE 802.11 MAC. However, that model did not consider frame losses in a WLAN, and so we first extend the model to accommodate frame losses.

Figure 3 shows details of the frame exchanges between a wireless client and an AP. To send one data frame to the AP by using RTS/CTS mechanisms (Fig. 3(a)), the wireless client first exchanges RTS/CTS frames after a random backoff time. After that, it sends the data frame and receives the corresponding ACK frame. In contrast, when the AP sends one data frame to the client without RTS/CTS mechanism (Fig. 3(b)), it sends the data frame immediately after a random backoff time.

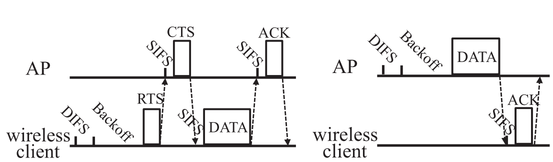
In both of the above sequences, the expected backoff time of ith transmission after $(i-1)$ consecutive transmission failures is determined by the following equations:

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

where T_{slot} is the slot time and $CW(i)$ is the contention window size of ith transmission after $(i-1)$ consecutive transmission failures. $CW(i)$ is given by

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

where CW_{min} and CW_{max} are the minimum and maximum



(a) From client to AP with RTS/CTS mechanisms (b) From AP to client without RTS/CTS mechanisms

Fig. 3 Frame exchange in IEEE 802.11 MAC.

values of the contention window size, respectively.

From Fig. 3(a) and Eq. (1), the average duration for the wireless client to send a single data frame for the ith transmission, $T^t(i)$, is calculated as follows:

$$\begin{aligned} T^t(i) =& 3T_{SIFS}+T_{DIFS}+T_{BO}(i)+4\tau+T_{RTS}+T^{client}_{DATA} \\ &+ T_{CTS} + T_{ACK} \end{aligned} \tag{3}$$

where T_{SIFS} is the short interframe space (SIFS), T_{DIFS} is the distributed interframe space (DIFS). T_{RTS} and T_{CTS} are the transmission duration of the RTS and CTS frame, respectively. T^{client}_{DATA} is the transmission duration of a data frame, T_{ACK} is the reception duration of an ACK frame, and τ is the radio propagation delay between the wireless client and the AP.

Similarly, the average duration by the wireless client to receive a single data frame for the ith transmission, $T^r(i)$, is

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

where T^{AP}_{DATA} is the reception duration of a data frame sent from the AP. Note that the expected values of T^{client}_{DATA} and T^{AP}_{DATA} are functions of the expected SCTP packet size, which is derived in Sect. 4.4.

From Eqs. (3) and (4), we derive the expected values of $T^t(i)$ and $T^r(i)$, denoted by $E[T^t]$ and $E[T^r]$, respectively. Here, we introduce q, the probability of transmission failures at MAC level, and N, a maximum number of data frame retransmissions. Then, using q and N, the probability that a data frame is transmitted i times after $(i-1)$ consecutive failures can be calculated as follows.

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

Since the duration in which the ith transmission becomes successful after $(i-1)$ consecutive failures is given by $\sum_{j=1}^{i} T^t(j)$, from Eqs. (3), (4), and (5), $E[T^t]$ and $E[T^r]$ are given by

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

Next, we determine $E[J^t]$ and $E[J^r]$. From Fig. 3(a) and Eq. (1), the energy consumption for the ith data frame transmission after $(i-1)$ consecutive failures is obtained by

$$\begin{aligned} J^t(i) =& P^l(3T_{SIFS} + T_{DIFS} + T_{BO}(i) + 4\tau) \\ &+ P^t(T_{RTS} + T^{client}_{DATA}) + P^r(T_{CTS} + T_{ACK}). \end{aligned} \tag{7}$$

In a similar way, the energy consumption for the ith data frame reception after $(i-1)$ consecutive failures is

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

Using Eqs. (5), (7), and (8), $E[J^t]$ and $E[J^r]$ are then calculated as follows:

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

4.3 Power Consumption of SCTP tunneling

The congestion control mechanisms in SCTP tunneling are the same as those in TCP, and so the client and an AP are activated independently. However, because the client and AP are under the same wireless conditions, we assume that their average behavior is identical. In addition, SCTP congestion control is applied to the entire association, and not to individual streams. Therefore, we can regard the behavior of SCTP congestion control mechanisms for multiple streams as being that for a single TCP flow. As a result, the power consumption model for SCTP tunneling is formulated based on the energy consumption model for a single TCP flow in [17], [18]. Specifically, we determine the power consumption for a WNI of the wireless client.

In what follows, we first explain the behavior of the congestion control mechanisms based on [17], [18], which is utilized for calculation of the power consumption for SCTP tunneling. After that, we describe duration of data transfer to determine the power consumption, which differs from that in [17], [18].

In SCTP data transfer, the number of packets sent per RTT depends on the evolution of the SCTP congestion window size. Figure 4 depicts the typical evolution of the congestion window size of an SCTP association from the beginning of the transmission. The data transfer starts with the *initial slow start phase* that ends due to the occurrence of a packet loss event. After that, SCTP packets are sent in the *steady phase* until the data transfer ends. For simplicity, the effects of initial slow start phase are not considered here because we focus on power consumption during the steady-state situation. Here, we define a triple duplicate (TD) period as the duration between two consecutive packet loss events detected by triple duplicate SCTP-SACKs. Further, the duration of a sequence of retransmission timeout (RTO) is referred to as a timeout (TO) period. The TO period lasts just before an SCTP packet received successfully at the SCTP receiver. Note that no SCTP-SACKs are received in TO periods. In steady phase, we can observe one TO period appears after multiple TD periods, and this sequence appears repeatedly.

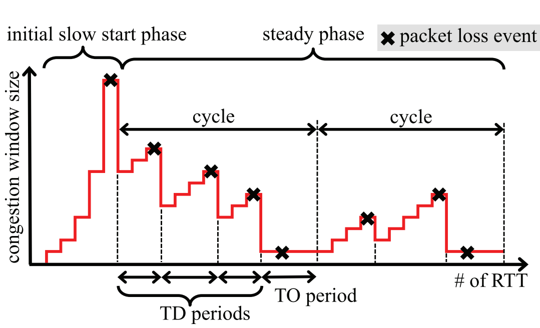


Fig. 4 Evolution of congestion window size.

According to our previous studies [17], [18], the power consumptions for a WNI of the wireless client with CAM and with sleeping, which are defined as P_{cam} and $P_{sleep}(m)$, are given by

$$P_{cam} = \frac{E[J^{TD}_{cam}] + Q(E[W],p)E[J^{TO}_{cam}]}{E[A] + Q(E[W],p)E[Z^{TO}]}, \quad (10)$$

$$P_{sleep}(m) = \frac{E[J^{TD}_{sleep}(m)] + Q(E[W],p)E[J^{TO}_{sleep}]}{E[A] + Q(E[W],p)E[Z^{TO}]} \quad (11)$$

where $E[J^{TD}_{cam}]$ and $E[J^{TD}_{sleep}(m)]$ are the expected energy consumed during a TD period when a WNI is activated in CAM and in sleep mode, respectively. $E[J^{TO}_{cam}]$ and $E[J^{TO}_{sleep}]$ are the expected energy consumed during a TO period when a WNI is activated in CAM and in sleep mode, respectively. $Q(w,p)$ is the probability that a packet loss is detected by an RTO as functions of window size w and probability of packet loss events p at the transport-layer level, and $E[W]$ is the expected window size when a packet loss event occurs in the TD period. Finally, $E[A]$ and $E[Z^{TO}]$ are the expected duration of TD and TO periods, respectively. Equations for $Q(w,p)$, $E[W]$, $E[A]$, and $E[Z^{TO}]$ were derived in [17], [18]. $E[J^{TD}_{cam}]$ are $E[J^{TO}_{cam}]$ are summations of energy consumed due to each state, namely packet transmission, packet reception, and idle period, during a TD period and during a TO period when a WNI is activated in CAM, respectively. Then, they are calculated as

$$E[J^{TD}_{cam}] = E[Y]E[J^t]+(E[Y]-E[W]/2)E[J^r] + P^l\left\{E[A]-E[Y]E[T^t]-(E[Y]-E[W]/2)E[T^r]\right\}, \quad (12)$$

$$E[J^{TO}_{cam}] = E[R]E[J^t]+P^l\left(E[Z^{TO}]-E[R]E[T^t]\right) \quad (13)$$

where $E[Y]$ is the expected number of SCTP packets sent during a TD period and $E[R]$ is the expected total number of SCTP packets sent during a TO period.

Similarly, $E[J^{TD}_{sleep}(m)]$ and $E[J^{TO}_{sleep}]$ are obtained as

$$\begin{aligned} E[J^{TD}_{sleep}(m)] &= E[Y]E[J^t]+(E[Y]-E[W]/2)E[J^r]+P^sE[T^s_{td}(m)] \\ &\quad +E[N^s_{td}(m)](P^{as}T^{as}+P^{sa}T^{sa}) \\ &\quad + P^l\left\{E[A]-E[Y]E[T^t]-(E[N^s_{td}(m)]-E[W]/2)E[T^r]\right. \\ &\quad \left.-E[T^s_{td}(m)]-E[N^s_{td}(m)](T^{as}+T^{sa})\right\}, \end{aligned} \quad (14)$$

$$E[J^{TO}_{sleep}] = E[R]E[J^t]+P^s\left(E[Z^{TO}]-E[R](E[T^t]+T^{as}+T^{sa})\right) + E[R](P^{as}T^{as}+P^{sa}T^{sa}). \quad (15)$$

Here, $E[N^s_{td}(m)]$ is the expected number of state transitions between active and sleep modes during a TD period and $E[T^s_{td}(m)]$ is the expected total sleep duration during a TD period. Note that $E[N^s_{td}(m)]$ and $E[T^s_{td}(m)]$ are as a function of the number m of packets sent in a bursty fashion, which are derived in [18]. Refer [18] for detailed calculation processes of Eqs. (12)–(15).

However, in [17], [18], $E[A]$ is calculated under the assumption that congestion control behavior is dependent on the average RTT of a TCP connection, whereas in SCTP tunneling this behavior is determined by the behavior of aggregate TCP flows. In this work, it depends on the arrival rate of TCP packets since we assume that average throughput of each TCP flow is given.

Let n^u and n^d be the numbers of upstream and downstream TCP flows, respectively, and r_i^u and r_j^d be the average throughputs [byte/s] of the ith upstream and the jth downstream TCP flows. Assuming that delayed ACK is not utilized in TCP[†], the numbers of TCP-DATA and TCP-ACK packets in a single TCP connection are identical. Here, let R^u [packet/s] and R^d [packet/s] are the number of SCTP packets sent from the wireless client to the AP per unit time and that sent from the AP to the wireless client per unit time, respectively. Since an SCTP packet contains a single TCP packet in our SCTP tunneling, R^u and R^d are given as

$$R^u = R^d = \sum_{i=1}^{n^u} \frac{r_i^u}{s_{data}^{tcp}} + \sum_{j=1}^{n^d} \frac{r_j^d}{s_{data}^{tcp}} \tag{16}$$

where s_{data}^{tcp} is the size [bytes] of a TCP-DATA packet. In what follows, R^u and R^d are simply denoted by R.

Next, we consider the SCTP packet sequence in the WNI of the wireless client, as shown in Fig. 5. In the figure, R_{sctp} is the average throughput of SCTP tunneling given by

$$R_{sctp} = \min\left(R,\ R_{sctp}^{max}\right). \tag{17}$$

Here, R_{sctp}^{max} is the maximum throughput achieved by SCTP tunneling, which is given by $1/\left(E[T^t] + E[T^r]\right)$. Note that duration of sent packets in a window is identical with and without burst transmission.

Finally, from Eq. (17) and our previous model [17], [18], $E[A]$ is calculated as

$$E[A] = \left(\frac{1-p}{p} + \frac{3}{2}E[W]\right)\frac{1}{R_{sctp}} \tag{18}$$

where p is the probability of packet drop events at the SCTP level, which is given by $p = q^{N+1}$.

4.4 Expected Size of SCTP Packets

As the mentioned in Sect. 4.2, the expected sizes of SCTP packets sent and received by the wireless client must be computed. To this end, we first calculate the ratios between the numbers of TCP-DATA and TCP-ACK packets in SCTP packets sent from the wireless client and the total number of SCTP packets sent from the client, which are denote by N_{data}^u and N_{ack}^u, respectively. By using the ratio between the aggregate throughputs of upstream TCP flows and downstream TCP flows, $\mu = \sum_{j=1}^{n^d} r_j^d / \sum_{i=1}^{n^u} r_i^u$, N_{data}^u and N_{ack}^u are calculated as follows:

$$N_{data}^u = \frac{1}{1+\mu},\quad N_{ack}^u = \frac{\mu}{1+\mu}. \tag{19}$$

Similarly, N_{data}^d and N_{ack}^d, which are the ratios between the numbers of TCP-DATA and TCP-ACK packets in SCTP packets sent from an AP and the total SCTP sent from the AP, respectively, are given by

$$N_{data}^d = \frac{\mu}{1+\mu},\quad N_{ack}^d = \frac{1}{1+\mu}. \tag{20}$$

Disregarding the length of an SCTP-SACK chunk, the expected sizes of SCTP packets sent from the wireless client and from the AP, which are defined as $\hat{s}_{sctp}^u$ [bytes] and $\hat{s}_{sctp}^d$ [bytes], are

$$\hat{s}_{sctp}^u = \frac{1}{1+\mu} s^{sctp}(s_{data}^{tcp}) + \frac{\mu}{1+\mu} s^{sctp}(s_{ack}^{tcp}), \tag{21}$$

$$\hat{s}_{sctp}^d = \frac{\mu}{1+\mu} s^{sctp}(s_{data}^{tcp}) + \frac{1}{1+\mu} s^{sctp}(s_{ack}^{tcp}). \tag{22}$$

Here, s_{ack}^{tcp} is the size of a TCP-ACK packet and $s^{sctp}(s)$ is the size of a SCTP packet as a function of payload size s [bytes], and

$$s^{sctp}(s) = 12 + 20 + s + padding \tag{23}$$

where *padding* is a padding byte to ensure that the length of $s^{sctp}(s)$ is a multiple of four bytes.

4.5 Increase in Transmission Latency due to Buffering Delay

Burst transmission, which is utilized by SCTP tunneling, causes transmission latency to each packet. Specifically, SCTP tunneling buffers SCTP packets at the tunnel inlet until m TCP packets arrive, which results in an additional delay for each TCP packet. Here, by using R, which is obtained by Eq. (16), the average buffering delay at the tunnel inlet is calculated as

$$D = \frac{m-1}{2}\frac{1}{R}. \tag{24}$$

5. Discussion with Numerical Results

In this section, we assess the energy efficiency of SCTP tunneling by means of the power consumption model described

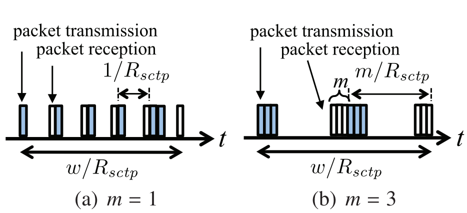


Fig. 5 SCTP packet sequences in WNI of wireless client (w: congestion window size of SCTP).

[†]This assumption can be relaxed easily. When delayed ACK is used, the number of TCP-ACK packets sent from a TCP receiver decreases.

in Sect. 4.

5.1 Parameter Settings and Evaluation Metrics

We consider an IEEE 802.11a WLAN in which multiple upstream and downstream TCP flows are established between a wireless client and wired hosts (Fig. 1). The WLAN parameters of IEEE 802.11a are summarized in Table 1. To calculate τ, which is the radio propagation between the wireless client and the AP, we assume that the wireless client is located four [meters] from the AP. From a data sheet for a WNI implemented by using the Atheros AR5004 chip [22] and measurement studies [6], [23], we set parameters of power consumption to the values listed in Table 2. The TCP-DATA and TCP-ACK packet sizes are set to 1500 [bytes] and 40 [bytes], respectively. The SCTP packet sizes for both directions are calculated by Eqs. (21) and (22). The maximum number of frame retransmissions, N, is set to seven.

In Sect. 5.2, we evaluate energy efficiency and trade-off relationships between energy efficiency and average buffering delay of SCTP tunneling. In order to assess the energy efficiency, we use power consumptions obtained by Eqs. (10) and (11). In contrast, to evaluate the trade-off relationships, we use *average buffering delay*, which is obtained by Eq. (24), in addition to *energy reduction ratio*, which is defined as

$$P_{ratio}(m) = \left(P_{cam} - P_{sleep}(m)\right)/P_{cam}. \tag{25}$$

Table 1 WLAN parameters.

Name	Value
Data rate	54 [Mbps]
Slot time	9 [μs]
SIFS	16 [μs]
DIFS	34 [μs]

Name	Value
PLCP preamble	16 [μs]
MAC header	24 [bytes]
LLC header	8 [bytes]
CW_{min}	15
CW_{max}	1023

Table 2 Power consumption of Atheros AR5004 [22] and parameters of state transitions [6], [23].

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

P^{as}	P^{sa}	T^{as}	T^{sa}
0.8 [W]	1.4 [W]	1 [μs]	1 [ms]

Energy efficiency is high when the ratio is large.

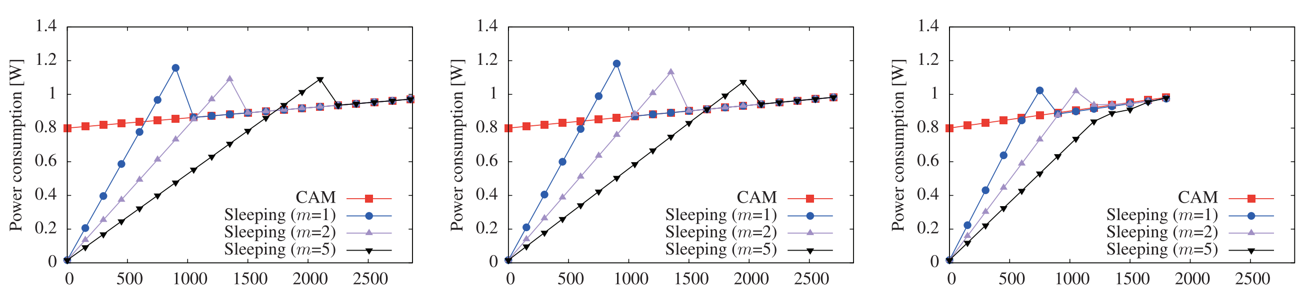
5.2 Numerical Results

5.2.1 Energy Efficiency

Figure 6 shows the power consumption results in the case that only upstream TCP flows exist when q = 0.1, 0.2, and 0.5. Here, we evaluate the performance of CAM and sleeping with burst transmission for m = 1, 2, and 5. Note that $m = 1$ signifies sleeping without burst transmission, while $m > 1$ is sleeping with burst transmission. In this figure, the x-axis represents the aggregate throughput of upstream TCP flows, where the average throughput of each TCP flow is 150 [kbyte/s]. Note that the maximum value of the aggregate throughput of upstream TCP flows is limited by the maximum throughput achieved by SCTP tunneling, R^{max}_{sctp}, which is dependent on the value of q. We also note that the results when upstream and downstream TCP flows coexist show a similar trend to Fig. 6. Although the difference between the number of upstream and downstream TCP flows affects the expected sizes of SCTP packets obtained by Eqs. (21) and (22), the packet size has only a small influence on the power consumption compared with protocol overheads, e.g., backoff and control frame exchanges.

From Fig. 6, we observe that the power consumption when utilizing CAM is increased by an increase in the aggregate throughput of upstream TCP flows. As the aggregate throughput grows, the duration of packet transmission and reception increases while idle duration decreases, which increases the power consumption. When sleeping is employed, the power consumption is considerably reduced regardless of the value of m. The power consumption increases for large aggregate throughput, whereas the increase rate of power consumption is low at large m values. For instance, when the aggregate throughput of upstream TCP flows is 450 [kbyte/s] in Fig. 6(a), sleeping without burst transmission reduces power consumption by 27% compared with CAM. In contrast, the reduction is around 69% for sleeping with m = 5. These results means that the smaller number of state transitions resulting from burst transmission has a large impact on energy reduction.

However, when aggregate throughput further increases,



(a) $q = 0.1$ ($p = 1.00 \times 10^{-8}$) (b) $q = 0.2$ ($p = 2.56 \times 10^{-6}$) (c) $q = 0.5$ ($p \approx 3.91 \times 10^{-3}$)

Fig. 6 Power consumption as a function of aggregate throughput of upstream TCP flows.

the power consumption with sleeping exceeds the power consumption of CAM. The power consumption required for state transitions exceeds the reduction realized by sleeping since the idle duration is short. Note that such a situation can be avoided by staying in active mode when the idle duration is insufficient. After that, aggregate throughput further increases, the power consumption with sleeping eventually becomes the same value of that with CAM. At this time, there is no idle duration to sleep.

Comparing Figs. 6(a), 6(b), and 6(c), we can observe that changes in q have little effect on the power consumption when CAM is employed. The power consumption during idle duration, whose length is affected by protocol overheads at MAC-level, provides a large contribution to the total power consumption compared with that during data frame retransmission even though the number of frame retransmissions increases for large q. In contrast, when sleeping is employed, the value of q has a greater impact on the power consumption with large m than that with small m. An increase in q decreases idle duration to sleep, which results in the decrease in the energy reduction for large m. Note that it has a smaller impact on the power consumption compared with the impact of aggregate throughput.

From the above results, we conclude that the power consumption of SCTP tunneling is hence predominantly determined by the aggregate throughput of TCP flows, while data frame retransmission provides only a small contribution to power consumption.

5.2.2 Trade-off Relationship between Energy Efficiency and Buffering Delay

Figure 7 shows the energy reduction ratio for various m values and $q = 0.1$. The corresponding average buffering delay is presented in Fig. 8.

As m increases, the energy reduction ratio converges to a constant value, whereas the average buffering delay increases linearly. On the other hand, as the aggregate throughput of TCP flows increases, the improvement rate of energy reduction ratio becomes large, whereas the increase rate of the delay becomes low. The power consumption of state transitions, which is reduced by burst transmission, provides a large portion of the total power consumption compared with the power consumption of state transitions when the aggregate throughput is low.

To further understand the trade-off relationships, we consider a situation in which acceptable buffering delay is given by a user or application. To this end, we introduce D_{th}, which denotes an upper limit of an acceptable average buffering delay. Figure 9 depicts the energy reduction ratio achieved subject to $D \leq D_{th}$. In the figure, the average throughput of each TCP flow is 50 [kbyte/s]. We observe that, as the aggregate TCP throughput increases, the energy reduction ratio when $D_{th} = 0$ decreases linearly and finally reaches zero, whereas the higher energy reduction is obtained for larger D_{th}. For example, when only 3 [ms] of additional delay is acceptable, we obtain roughly 0.3 and 0.4 reductions from the ratio for $D_{th} = 0$ when the aggregate TCP throughputs are 500 [kbyte/s] and 800 [kbyte/s], respectively. When 10 [ms] of additional delay is acceptable, we can obtain roughly 0.5 and 0.6 reductions from the ratio for $D_{th} = 0$ when the aggregate TCP throughputs are 500 [kbyte/s] and 800 [kbyte/s], respectively. In other words, when the aggregate TCP throughput is 800 [kbyte/s], we need to accept 10 [ms] of additional delay to save roughly 60% of energy, whereas accepting only 3 [ms] of additional delay is sufficient to reduce roughly 40% of energy consumption.

From above results, we conclude that, because SCTP tunneling causes additional delay for each TCP packet, a value of m for sleeping with burst transmission must be selected such that the trade-off between energy efficiency and

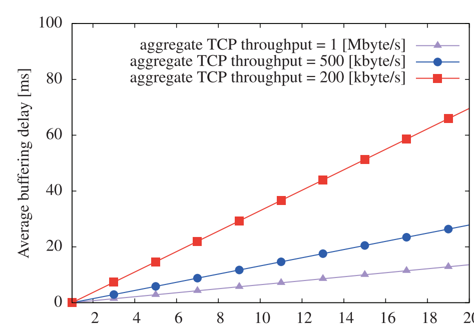


Fig. 8 Average buffering delay for various m values and $q = 0.1$

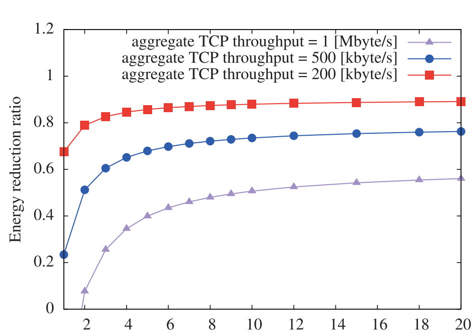


Fig. 7 Energy reduction ratio for various m values and $q = 0.1$.

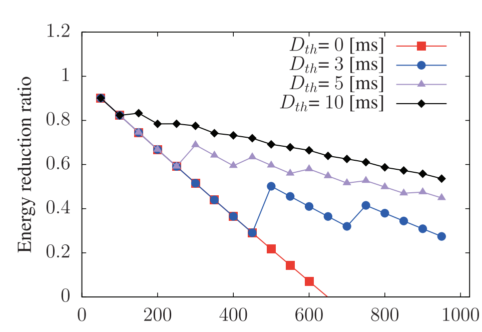


Fig. 9 Energy reduction ratio achieved subject to $D < D_{th}$ and $q = 0.1$.

the delay is at a level acceptable for users or applications.

6. Conclusion

We have proposed a transport-layer approach to reduce the energy consumed by TCP data transfer over a WLAN, termed SCTP tunneling. SCTP tunneling has two key features: flow aggregation and burst transmission at the transport-layer level. To assess the energy efficiency gained by SCTP tunneling, we formulated a power consumption model of SCTP tunneling based on the energy efficiency analysis of a single TCP flow in a WLAN. Numerical results of the model show that the power consumption of SCTP tunneling is predominantly determined by the aggregate throughput of TCP flows. Fortunately, burst transmission can considerably reduce power consumption with only a moderate increase in delay.

In the future, we plan to implement the SCTP tunneling on commercial WLAN APs and wireless clients with power saving mode such as PSM and APSD.

Acknowledgments

This work was supported in part by the "Strategic Information and Communications R&D Promotion Programme (SCOPE)" of the Ministry of Internal Affairs and Communications (MIC), Japan, and by a Grant-in-Aid for JSPS Fellows, 24-2395, from the Japanese Society for the Promotion of Science.

References

[1] Atheros Communications, "Power consumption and energy efficiency comparisons of wlan products," Atheros White Papers, May 2003.
[2] V. Raghunathan, T. Pering, R. Want, A. Nguyen, and P. Jensen, "Experience with a low power wireless mobile computing platform," Proc. ISLPED 2004, pp.363–368, 2004.
[3] Y. Agarwal, C. Schurgers, and R. Gupta, "Dynamic power management using on demand paging for networked embedded systems," Proc. ASP-DAC 2005, pp.755–759, Jan. 2005.
[4] T. Jin, G. Noubir, and B. Sheng, "WiZi-cloud: Application-transparent dual ZigBee-WiFi radios for low power internet access," Proc. INFOCOM 2011, pp.1593–1601, April 2011.
[5] IEEE 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. June 2007.
[6] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," Wireless Netw., vol.11, no.1-2, pp.135–148, Jan. 2005.
[7] IEEE 802.11e, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Medium Access Control (MAC) Quality of Service Enhancements. Oct. 2005.
[8] D. Camps Mur, X. Pérez-Costa, and S. Sallent Ribes, "An adaptive solution for wireless LAN distributed power saving modes," Computer Netw., vol.53, no.18, pp.3011–3030, Dec. 2009.
[9] J. Lee, S. Kwon, and D. Cho, "Adaptive beacon listening protocol for a TCP connection in slow-start phase in WLAN," IEEE Commun. Lett., vol.9, no.9, pp.853–855, Sept. 2005.
[10] Y. He and R. Yuan, "A novel scheduled power saving mechanism for 802.11 wireless LANs," IEEE Trans. Mobile Comput., vol.8, no.10, pp.1368–1383, Oct. 2009.
[11] K.C. Ting, H.C. Lee, H.H. Lee, and F. Lai, "An idle listening-aware energy efficient scheme for the DCF of 802.11n," IEEE Trans. Consum. Electron., vol.55, no.2, pp.447–454, May 2009.
[12] J. Liu and L. Zhong, "Micro power management of active 802.11 interfaces," Proc. MobiSys 2008, pp.146–159, 2008.
[13] H. Yan, S.A. Watterson, D.K. Lowenthal, K. Li, R. Krishnan, and L.L. Peterson, "Client-centered, energy-efficient wireless communication on IEEE 802.11b networks," IEEE Trans. Mobile Comput., vol.5, no.11, pp.1575–1590, Nov. 2006.
[14] V. Namboodiri and L. Gao, "Energy-efficient VoIP over wireless LANs," IEEE Trans. Mobile Comput., vol.9, no.4, pp.566–581, April 2010.
[15] F.R. Dogar, P. Steenkiste, and K. Papagiannaki, "Catnap: Exploiting high bandwidth wireless interfaces to save energy for mobile devices," Proc. MobiSys 2010, pp.107–122, June 2010.
[16] R. Stewart, "Stream control transmission protocol," Request for Comments 4960, Sept. 2007.
[17] M. Hashimoto, G. Hasegawa, and M. Murata, "Modeling and analysis of power consumption in TCP data transmission over a wireless LAN environment," Proc. GreenComm 2011, pp.1–6, June 2011.
[18] M. Hashimoto, G. Hasegawa, and M. Murata, "Energy efficiency analysis of TCP with burst transmission over a wireless LAN," Proc. ISCIT 2011, pp.292–297, Oct. 2011.
[19] S. Ladha and P.D. Amer, "Improving file transfers using SCTP multistreaming," Proc. IPCCC 2004, pp.513–522, April 2004.
[20] P. Natarajan, P.D. Amer, and R. Stewart, "Multistreamed web transport for developing regions," Proc. NSDR 2008, pp.43–48, Aug. 2008.
[21] R. Braden, "Requirements for internet hosts — Communication layers," Request for comments 1122, Oct. 1989.
[22] Wistron NeWeb Corp., "CM9: WLAN 802.11 a/b/g mini-PCI Module," available at http://site.microcom.us/CM9.pdf
[23] C. Andren, T. Bozych, B. Road, and D. Schultz, "PRISM power management modes: Application note AN9665," Feb. 1997.

Masafumi Hashimoto received the M.E. degree in Information Science and Technology from Osaka University, Japan, in 2010. He is now a D.E. candidate at Graduate School of Information Science and Technology, Osaka University. His research work is in the area of transport architecture for future high-speed networks.

Go Hasegawa received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1997 and 2000, respectively. From July 1997 to June 2000, he was a Research Assistant of Graduate School of Economics, Osaka University. He is now an Associate Professor of Cybermedia Center, Osaka University. His research work is in the area of transport architecture for future high-speed networks and overlay networks. He is a member of the IEEE.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than five hundred papers of international and domestic journals and conferences. His research interests include computer communication network architecture, performance modeling and evaluation. He is a member of IEEE and ACM. He is a chair of IEEE COMSOC Japan Chapter since 2009. Also, he is now partly working at NICT (National Institute of Information and Communications Technology) as Deputy of New-Generation Network R&D Strategic Headquarters.