Performance Analysis of IEEE 802.16m Sleep Mode for Heterogeneous Traffic

Sunggeun Jin, Member, IEEE, Munhwan Choi, Student Member, IEEE, and Sunghyun Choi, Senior Member, IEEE

Abstract—We numerically analyze the performance of the emerging 802.16m’s sleep mode operation in order to gain a new insight regarding its power consumption and traffic transmission delay when a Mobile Station (MS) in the sleep mode is served with both non-realtime and realtime traffic simultaneously. We validate the analysis via the comparison with simulation results.

Index Terms—IEEE 802.16, IEEE 802.16m, sleep mode, power saving.

I. INTRODUCTION

In IEEE 802.16 Wireless Metropolitan Area Networks (WMANs), Mobile Stations (MSs) can save their energy consumption with sleep mode when they are served with lightly loaded and/or realtime traffic [1]. For the purpose, three types of Power Saving Classes (PSCs) are specified depending on traffic types. Accordingly, an MS in the sleep mode manages properly chosen PSC(s) for its connections according to their corresponding traffic types. When a PSC is activated, sleep windows interleaved with listening windows of a fixed duration repeat over time for the PSC. A listening window is a time duration during which traffic can be exchanged between the MS and the Base Station (BS) while a sleep window is used to power down MS’s transceiver for power saving.

However, the existing 802.16 sleep mode has a drawback. The listening window is not adjustable once its size is determined. For this reason, a BS cannot transmit traffic when a listening window expires even in the case that the BS has more traffic destined for an MS in the sleep mode. In such a case, the MS will experience an extended traffic reception delay. Additionally, in the 802.16 standard, an MS with multiple connections is allowed to manage multiple PSCs independently for its connections while multiple connections can be also mapped onto a single PSC. In the former case, a sleep window of a PSC might overlap with the listening windows of other PSCs. Note that the MS cannot power down its transceiver in such overlapped periods so that the energy cannot be saved.

In order to overcome these shortcomings, the emerging 802.16m, which inherits most sleep mode features from the 802.16, adopts the following new strategies: (1) an MS is constrained to have only a single PSC in the sleep mode; and (2) the listening window is adjustable depending on BS’s buffer status and/or Hybrid Automatic Repeat reQuest (HARQ) retransmission state [2]. In this letter, we investigate the impacts of these new strategies. Since many studies have been already made for the 802.16 sleep mode [4]–[9], we focus only on the new strategies of the 802.16m sleep mode by excluding the common parts of the 802.16m and the 802.16 sleep modes.

The letter is organized as follows: in Section II, we briefly explain the 802.16m sleep mode operation. In Section III, we present a numerical analysis based on a queuing model, and then, obtain performance evaluation results in Section IV. Finally, the letter concludes in Section V.

II. IEEE 802.16M SLEEP MODE

The 802.16m PSCs can be classified into two types depending on traffic types, namely, PSC type-I and PSC type-II, both of which are inherited from the 802.16. Note that we employ the terms of PSC type-I and PSC type-II for consistent explanation although those terms are obsolete in the 802.16m standard. The PSC type-I is designed for Best Effort (BE) and Non-RealTime (NRT) traffic while the PSC type-II is for RealTime (RT) traffic. Unlike the 802.16, the 802.16m sleep mode adopts sleep cycle comprising listening window and sleep window. Fig. 1 shows an exemplary operation of the PSC type-I. In the PSC type-I, the sleep cycle doubles until it reaches its maximum value with or without traffic exchange.
during a listening window. Fig. 2 shows that the sleep cycle remains constant for the 802.16m PSC type-II. In the figures, \( T_C \) represents a time duration for a sleep cycle. Besides the explained basic operations for the PSC type-I and II, the 802.16m sleep mode has the following new features: (1) only a single PSC is allowed for an MS no matter how many connections it has. This policy aims at preventing sleep windows from overlapping with listening windows when an MS in the sleep mode has multiple PSCs; (2) the MS reduces a listening window size by receiving BS’s request message when the BS’s buffer is empty; (3) a listening window is extended temporarily with a predefined time duration provided that traffic arrives within a short time duration before the listening window’s expiration. It can be also possibly extended in order to guarantee a successful HARQ retransmission. The extended listening window can grow continuously until a sleep cycle is completely filled with the listening window; and (4) the MS and the BS can renegotiate the PSC parameters, e.g., listening and sleep window sizes, and the sleep cycle increment policy, even during an ongoing sleep mode operation, if necessary.

III. ANALYSIS

In the previous work [4]–[9], the basic 802.16 sleep mode operations for the PSC type-I and the PSC type-II are independently analyzed enough to understand each. Recently, the authors of [3] study the efficiency of the 802.16m sleep mode. However, they deal with only non-realtime traffic. On the contrary, we analyze the new features of the 802.16m sleep mode under the scenario that an MS is serviced by both realtime and non-realtime traffic. For this scenario, the PSC type-II is naturally adopted since the delay bound requirement of realtime traffic can be satisfied only with this type. Then, non-realtime traffic transmission is scheduled after the realtime traffic transmission in each listening window.

However, despite that the periodic listening windows should accommodate both realtime and non-realtime traffic transmissions, the listening window size, which is determined at the beginning of the PSC type-II, usually does not match exactly the requirement. This is mainly caused by the different traffic transmission time in each listening window due to time-varying wireless channel characteristics. Additionally, it is also incurred by aperiodic non-realtime traffic arrivals. Nevertheless, as explained above, the listening window size is adjustable to meet the exact requirement when necessary in each sleep cycle.

Fig. 3 shows how to manage listening and sleep windows during the run-time by considering downlink transmissions. In each listening window, the BS transmits realtime traffic, and then, it continues to transmit non-realtime traffic (if any) until its transmission buffer becomes empty. After the BS finishes its transmissions, a sleep window starts. During the sleep window, the BS buffers newly-arriving non-realtime traffic for the subsequent listening window. Note that, during a listening window, the MS in the sleep mode stays awake to receive downlink traffic. In this figure, \( T_C, T_L \), and \( T_S \) indicate the time durations for sleep cycle, listening window, and sleep window. \( T_C, T_L \), and \( T_S \) are the corresponding expectations.

For a proper queuing analysis, we make the following two assumptions: (1) non-realtime packets arrive according to a Poisson process with rate \( \lambda \), while a realtime packet arrives at the beginning of each sleeping window; (2) the packet transmission times for the realtime and the non-realtime traffic are exponentially distributed with expectations of \( 1/\mu_r \) and \( 1/\mu \), respectively.

In the steady state, when the BS begins transmitting realtime traffic at the beginning of a listening window, it already contains non-realtime traffic buffered during the preceding sleep window. While the BS is transmitting the buffered packets, new packets may arrive at the buffer. Therefore, we discriminate the packet transmissions with index \( K \) indexing a group of packet transmissions for the packets buffered during the time required to complete the packet transmissions for the \((K-1)\)st group. Initially, \( K = 0 \) indexes the group of packet transmissions for the buffered packets during the previous sleep window as well as a packet for the realtime traffic. The random variable \( t(K=0) \) represents the time required to transmit the whole packets belonging to the \( K \)th group. For further derivations, we simply denote \( t(K=k) \) by \( t(k) \).

We derive the expectation of \( t^{(0)} \) by \( E[t^{(0)}] = 1/\mu_r + \lambda T_S/\mu \) since \( \lambda T_S \) packets are buffered in the proceeding sleep window. During \( t^{(0)} \), additional traffic might arrive in the buffer, thus requiring more transmission time, namely, \( t^{(1)} \). This rule is applied iteratively until the buffer becomes empty. From this rule, we have the relationship \( E[t(k)] = E[t(k-1)] + t^{(1)} \) for \( t(K=0) = t^{(0)} \). The probability \( \Pr[J=j] = t^{(1)}(j-1) = t^{(1)} \) representing \( j \) packet arrivals during \( t^{(1)} \) is given by \( (\lambda t^{(1)})^j e^{-\lambda t^{(1)}}/j! \).

Therefore, a listening window size \( T_L \) in a sleep cycle should be the total time required for transmissions until emptying the BS’s transmission buffer. Accordingly, the expected listening window size \( E[T_L] = E[T_S] \) is derived by \( T_L = \sum_{k=0}^{\infty} E[t(k)] = \sum_{k=0}^{\infty} E[t^{(0)}] \rho^k/\rho = \rho T_C + 1/\mu_r \), where utilization \( \rho = \lambda/\mu \). From this equation, we obtain the average power consumption \( P \) by:

\[
P = \frac{P_L T_L + P_S T_S}{T_C} = (P_L - P_S) \left( \frac{1}{\mu_r T_C} \right) + P_S, \tag{1}
\]
Fig. 4. Analysis and simulation results, where \( P_L \) and \( P_S \) are the amount of power consumptions in listening window and sleep window, respectively.

Prior to the derivation of the packet transmission delay, we observe the number of packets in the BS’s buffer vary in the steady state. The BS’s buffer contains \( 
\lambda T_S \) non-realtime packets at the beginning of the listening window while it becomes empty at the end of the listening window. During \( 1/\mu_r \) at the beginning of the listening window, one realtime packet is served with the highest priority. Then, non-realtime packets are transmitted with expected non-realtime packet transmission time (= \( 1/\mu \)). On the other hand, \( \lambda T_S \) non-realtime packets are buffered during the sleep window, which begins with the empty buffer. Therefore, we derive the expected number of packets in the buffer (= \( L \)) by \( L = \frac{1}{T_C} \int_0^{T_C} \left( (1 + \rho T_S + \lambda t) - \mu_r t \right) dt + \frac{1}{T_C} \int_0^{T_C} \left( (1 + \lambda T_S + \lambda \rho T_C) - \mu_r t \right) dt + \frac{1}{T_C} \int_0^{T_C} (\lambda t) dt = \frac{1}{T_C} \lambda T_C (1 - \rho) + \frac{1}{T_C} \mu_t + \frac{1}{T_C} \lambda T_C. \) From this equation and the Little’s Law, the the non-realtime packet transmission delay (= \( D \)) is derived by:

\[
D = \frac{1}{\lambda} \left( L - \frac{1}{T_C} \frac{1}{\mu_r} \right). \tag{2}
\]

IV. EVALUATION

For our evaluation, we assume that both \( 1/\mu_r \) and \( 1/\mu \) are set to 0.1 ms. \( 1/\lambda \) is determined depending on the value of the utilization \( \rho \), which ranges between 0.1 and 0.4 since we only have an interest in the MS with lightly loaded and realtime traffic for the sleep mode. Additionally, let \( P_L \) and \( P_S \) be 750 mW and 50 mW, respectively [7].

Fig. 4 shows both analysis and simulation results. In this figure, Analysis and Simulation show the numerical analysis and simulation results, respectively. The well-match results prove the equations are correctly derived. Fig. 4(a) depicts that the higher \( \rho \) incurs the more power consumption since the power consumption is directly proportional to \( \rho \). This figure shows that \( T_C \) gives minor impact to the power consumption due to the fact that the realtime packet transmission time is negligible compared with \( T_C \). Fig. 4(b) shows that the higher \( \rho \) causes the shorter packet transmission delay. \( T_S \), in which newly arrived packets are accumulated without transmission, decreases with higher \( \rho \). It implies that the average number of packets in the buffer decreases, thus resulting in the shorter packet transmission delay as \( \rho \) increases. In this figure, we observe that \( T_C \) influences the non-realtime packet transmission delay significantly. With smaller \( T_C \), the non-realtime traffic can be more frequently served.

From the observations, we learn the following two aspects: (1) there is a tradeoff relationship between the power consumption and the packet transmission delay as the utilization \( \rho \) varies; (2) \( T_C \) can be configured to set to a divisor of the realtime traffic arrival period in order to achieve short non-realtime packet transmission delay. For example, an MS is served with realtime traffic, of which arrival period is 100 ms. Then, either 50 ms or 20 ms may be applicable to the value of \( T_C \) for faster non-realtime traffic transmission.

V. CONCLUSION

In this letter, we derive the power consumption and the non-realtime packet transmission delay for the 802.16m sleep mode when an MS is served with both realtime and non-realtime traffic. The derivations are validated through simulations. We find out that there is a tradeoff relationship between the power consumption and the transmission delay depending on the utilization. Our analysis can be used to find an optimal point satisfying tradeoff relationship under a particular condition since the utilization can be under control with transmission rate adjustment.

REFERENCES