Calculating approximate blocking probabilities for TDM wavelength optical networks with OTSIs

Yoong Cheah Hueia,∗, Pung Hung Keng a and Nikolai Krivulin b

a School of Computing, National University of Singapore, Singapore
b Faculty of Mathematics and Mechanics, St. Petersburg State University, St. Petersburg, Russia

Abstract. Previous works have proposed analytical models of TDM wavelength networks to evaluate the blocking performance, but they differ in their underlying assumptions and have varying computation complexities. In this paper, we propose an analytical model of TDM WDM optical networks with optical time-slot interchangers and without wavelength converter for minimum fixed hop routing. In order to make the analysis tractable, we propose a schema that work for any number of partition patterns regardless of numbers of links, wavelengths in each link, and time-slots in each wavelength. Our analytical model provides good accuracies in individual routes and average network blocking probability when compared with the simulation results.

Keywords: Schema, optical time-slot interchangers, average network blocking probability

1. Introduction

Wavelength division multiplexing (WDM) is a promising technology to accommodate the explosive growth in telecommunication and Internet traffic. Currently, this technology is widely employed in the backbone network.

Recent advances in WDM technology have generated multiple magnitudes of raw bandwidth. Wavelength bandwidth allocation problem has been investigated in [10,23]. In order to accommodate the demerit of wavelength routed networks in bandwidth wastage, the time division multiplexed (TDM) wavelength routed network is proposed [17]. In each TDM frame, the number of time-slots is fixed. Each routing node behaves like a traditional TDM circuit switching node. Thus, the time-slots are pre-assigned during connection set-up. The main function of the routing node is to connect the incoming data in each time-slot into the desired output port, and the data transmission is all-optical. Hence, the bandwidth of each wavelength is more efficiently utilized, and the bottleneck of electronic data processing at the time-slot level is avoided. Paper [20] studies the problem of routing, wavelength and time-slot-assignment in wavelength routed TDM WDM optical networks with the goal maximizing throughput in the network. Paper [13] studies the switch reconfiguration capability in TDM wavelength routing networks. Paper [4] intends to maximize the performance of optical TDM networks with a small number of optical buffers. Paper [7] proposes an optical architecture that is able to transmit data optically at the time-slot wavelength level without using optical time-slot interchangers (OTSIs). Design and simulation study on OTSIs producing encouraging results are found in [11]. An OTSI has been demonstrated to be feasible in [3].

The generalized reduced load approximation method for circuit-switched networks is given in [8] and further developed by [5]. Paper [2] extends the method to wavelength routing model for fixed routing and without wavelength converter (WC). The idea given in [2] is extended by [19] to derive an analytical expression to compute the

∗Corresponding author: Yoong Cheah Huei, School of Computing, National University of Singapore, 117543 Singapore. Tel.: +02 9237 6921; Fax: +65 6469 0490; E-mail: Ayoong1@gmail.com.
blocking probability of networks with limited-range wavelength conversion for fixed routing. Paper [14] studies the blocking probabilities of WDM networks with no wavelength conversion using inclusion–exclusion principle of combinatorics. Papers [1,9,24] analyze the blocking performances of networks with no wavelength conversion and full wavelength conversion at each node. Papers [12,16,22] present analytical models with limited wavelength conversion. Paper [6] focuses on success probability using WCs in a network. Paper [21] extends the model presented in [9] to multiwavelength TDM network with and without WC. The independent link load assumption is used in [21]. Paper [18] shows less accuracy in one time-slot method using partitions of 0 slots free, 1 slot free and ≥2 slots free. The reduced load model is not used in [18]. Paper [15] presents a generalized framework for analyzing time–space switched optical networks. This paper uses a \( z \)-link path model, where the first hop has \( z - l \) links, and the second hop consists of the last two links. The Markovian correlation is assumed because the number of trunks free on the last link depends on the number of trunks free on the previous link.

In this paper, we extend the proposal in [2] to TDM WDM optical networks with OTSIs and without WC. Our partition based method does not use the \( z \)-link path model and Markovian correlation assumption. The dependent link load assumption is used in our method. The reduced load algorithm is used in our paper.

The rest of the paper is organized as follows. Section 2 analyzes the proposed analytical model for TDM WDM optical networks with OTSIs and without WC. Section 3 calculates the average network blocking probability. Section 4 presents the comparisons of the simulated and analytical results. Section 5 concludes this paper.

2. Analytical model

2.1. Analysis of TDM WDM optical networks with OTSIs and without WC

The following assumptions are used in our proposed analytical model:

(i) Each connection is assumed to use an entire time-slot on a single wavelength on each link within its route. Each link is assumed to have the same number of wavelengths, and each wavelength contains the same number of time-slots. The capacity of each link denoted by \( C \) is the same for all the links in the network.
(ii) External calls arrive at each node according to the Poisson process with rate \( \lambda \).
(iii) Call holding time is exponentially distributed with unit mean.
(iv) Calls that cannot be routed in the network are blocked and never return.
(v) Random assignment of a wavelength and a time-slot in the selected wavelength at the first hop.
(vi) Existing calls cannot be assigned another time-slot to accommodate new call requests.

Let \( p_{1}^{l}(x_{1}, \ldots, x_{r}) \) be the probability not to have at least one empty time-slot at common wavelengths throughout the links \( 1, \ldots, r \) each having \( l \) wavelengths with \( s \) time-slots per wavelength, provided that there are \( x_{i} \) empty time-slots over all wavelengths at link \( i, i = 1, \ldots, r \).

Let \( N_{\text{case}} \) denote the total number of unique cases and \( N_{\text{xin}} \) denote the number of arrangements for each unique case in link \( x_{i} \). The total number of arrangements in link \( x_{i} \) for the number of empty time-slots in link \( x_{i} \) is given by \( T_{\text{arr}} = \sum_{x_{i}=1}^{N_{\text{arr}}} N_{\text{xin}} \) if the number of empty time-slots in a wavelength is bounded by \( s \). We shall use \( T_{\text{arr}} = \sum_{x_{i}=1}^{N_{\text{arr}}} N_{\text{xin}} \) for each link in the denominator.

\( T_{\text{arr}} \) can be determined by the following steps:

(1) Calculate the number of unique cases. For each unique case, the number of empty time-slots in all the wavelengths of a link compared with the rest of the cases must not be the same regardless of the arrangement position.

(2) For each unique case, determine the number of arrangements. The numerator is the factorial of number of wavelengths (\( l \)) in a link. The denominator is the product of the factorial of each unique number of empty time-slots in each wavelength. If the same number of empty time-slots in each wavelength of a link occurs only once, we have \( l! \). A \( 2! \) means that the same number of empty time-slots in each wavelength of a link occurs twice. Similar explanation is used for \( 3!, 4! \) and so on.
(3) Add the number of arrangements for each unique case, and we shall obtain the value for $T_{arr}$.

For example, let $l = 2$, $s = 2$ and $x_k$ – number of empty time-slots in link $k = 2$.

For step 1, we calculate the number of unique cases and their corresponding arrangements.

<table>
<thead>
<tr>
<th>Case</th>
<th>Number of arrangements</th>
</tr>
</thead>
<tbody>
<tr>
<td>$wav_1$</td>
<td>$wav_2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Finally, step 3, we add the number of arrangements for unique cases one and two. Thus, $T_{arr}$ is three – $2! = 2$ and $1! = 1$.

Another example, let $l = 2$, $s = 2$ and $x_i$ – number of empty time-slots in link $i = 3$.

For the numerator, we used appropriate partitions of a matrix-based scheme (partition pattern) like the form

$$
\begin{bmatrix}
  + & 0 \\
  0 & +
\end{bmatrix}
$$

and

$$
\begin{bmatrix}
  0 & + \\
  + & 0
\end{bmatrix}
$$

The related blocking probability is then represented as $2(\sum_{n=1}^{\text{N}_{\text{case}}} N_{x1n})^{-1}(\sum_{n=1}^{\text{N}_{\text{case}}} N_{x2n})^{-1}$. Finally, we have

$$
p^{(2)}_0(x_1, x_2) = \begin{cases} 
1 & \text{if } x_1 = 0 \text{ and/or } x_2 = 0, \\
2 & \sum_{n=1}^{N_{\text{case}}} N_{x1n} \sum_{n=1}^{N_{\text{case}}} N_{x2n} \\
0 & \text{otherwise.}
\end{cases}
$$

Now, let us consider the blocking probability $p^{(2)}_0(x_1, x_2, x_3)$ for three links, each with two wavelengths.

For the denominator, we shall use the formula $T_{arr} = \sum_{n=1}^{\text{N}_{\text{case}}} N_{xin}$ to determine the total number of arrangements for the number of empty time-slots in a link if the number of empty time-slots in a wavelength is bounded by $s$.

For the numerator, we calculate number of partitions which have no empty time-slots at common wavelengths using a matrix-based scheme (partition pattern) like the form

$$
\begin{bmatrix}
  + & 0 \\
  0 & +
\end{bmatrix}
$$

where zero at the place on row $i$ and column $j$ means $x_{ij} = 0$, whereas the plus sign implies that $x_{ij} > 0$. 

Suppose that $0 < x_1, x_2, x_3 \leq s$. The suitable partition patterns are represented as follows with the overall number of particular partitions put on the right of each group of related patterns:

\[
\begin{align*}
&\begin{bmatrix} + & 0 \\ + & 0 \end{bmatrix}, \begin{bmatrix} + & + \\ 0 & + \end{bmatrix}, \begin{bmatrix} 0 & + \\ 0 & + \end{bmatrix}, \left( x_1 + 1 \right) = (x_1 + 1); \\
&\begin{bmatrix} + & 0 \\ 0 & + \end{bmatrix}, \begin{bmatrix} + & + \\ 0 & + \end{bmatrix}, \begin{bmatrix} 0 & + \\ + & 0 \end{bmatrix}, \left( x_2 + 1 \right) = (x_2 + 1); \\
&\begin{bmatrix} 0 & + \\ + & 0 \end{bmatrix}, \begin{bmatrix} + & + \\ 0 & + \end{bmatrix}, \begin{bmatrix} 0 & + \\ 0 & + \end{bmatrix}, \left( x_3 + 1 \right) = (x_3 + 1).
\end{align*}
\]

Since the patterns in the first and third columns appear to be counted twice and each of the patterns has only one representative partition, the number of different partitions is

\[
2(x_1 + 1) + 2(x_2 + 1) + 2(x_3 + 1) - 6 = 2(x_1 + x_2 + x_3).
\]

The related blocking probability is equal to

\[
p_o^{(2)}(x_1, x_2, x_3) = \frac{2(x_1 + x_2 + x_3)}{(\sum_{n=1}^{N_{case}} N_{x1n})(\sum_{n=1}^{N_{case}} N_{x2n})(\sum_{n=1}^{N_{case}} N_{x3n})}.
\]

Consider the case when $s < x_1 \leq 2s$ and $0 < x_2, x_3 \leq s$. The partition patterns and number of partitions are as follows:

\[
\begin{bmatrix} + & + \\ + & 0 \end{bmatrix}, \begin{bmatrix} + & + \\ 0 & + \end{bmatrix} = 2(2s - x_1 + 1).
\]

For the case when $s < x_2 \leq 2s$ and $0 < x_1, x_3 \leq s$, the number of partitions is $2(2s - x_2 + 1)$. For the case when $s < x_3 \leq 2s$ and $0 < x_1, x_2 \leq s$, the number of partitions is $2(2s - x_3 + 1)$. Finally, we get (2). When any two of the variables $x_1, x_2, x_3$ are greater than $s$, there is always a common wavelength with empty time-slots.
\[ p_0^{(2)}(x_1, x_2, x_3) = \begin{cases} 
1 & \text{if } x_i = 0 \text{ for at least one } i = 1, 2, 3, \\
\frac{2(x_1 + x_2 + x_3)}{(\sum_{n=1}^{N_{x1n}} N_{x2n}) \sum_{n=1}^{N_{x3n}}} & \text{if } 0 < x_1, x_2, x_3 \leq s, \\
\frac{2(2s - x_1 + 1)}{(\sum_{n=1}^{N_{x1n}} N_{x2n}) \sum_{n=1}^{N_{x3n}}} & \text{if } 0 < x_1, x_2 \leq s \text{ and } s < x_3 \leq 2s, \\
\frac{2(2s - x_2 + 1)}{(\sum_{n=1}^{N_{x1n}} N_{x2n}) \sum_{n=1}^{N_{x3n}}} & \text{if } 0 < x_1, x_3 \leq s \text{ and } s < x_2 \leq 2s, \\
\frac{2(2s - x_1 + 1)}{(\sum_{n=1}^{N_{x1n}} N_{x2n}) \sum_{n=1}^{N_{x3n}}} & \text{if } 0 < x_2, x_3 \leq s \text{ and } s < x_1 \leq 2s, \\
0 & \text{otherwise.} 
\end{cases} \] (2)

Now let us consider three wavelengths in each of the two links. The blocking probability of \( p_0^{(3)}(x_1, x_2) = 1 \), if \( x_1 = 0 \) and/or \( x_2 = 0 \). The blocking probability of \( p_0^{(3)}(x_1, x_2) = 0 \), if \( x_1, x_2 > 0 \) and at least one of the variables \( x_1 \) and \( x_2 \) is greater than \( 2s \).

For the denominator, we shall use the formula \( T_{\text{arr}} = \sum_{n=1}^{N_{xin}} N_{xin} \) to determine the total number of arrangements for the number of empty time-slots in a link if the number of empty time-slots in a wavelength is bounded by \( s \).

For the numerator, we begin by assuming that \( 0 < x_1 \leq s \) and \( 0 < x_2 \leq s \). The suitable partition patterns are represented in the next page together with overall number of particular partitions put on the right of each group of related patterns.

\[
\begin{bmatrix}
+ & 0 & 0 \\
0 & + & 0 \\
0 & 0 & + \\
\end{bmatrix}, \quad \begin{bmatrix}
+ & 0 & 0 \\
0 & 0 & + \\
0 & 0 & + \\
\end{bmatrix}, \quad \begin{bmatrix}
+ & 0 & 0 \\
0 & + & 0 \\
0 & + & 0 \\
\end{bmatrix}, \quad \begin{bmatrix}
+ & 0 & + \\
0 & + & 0 \\
0 & + & 0 \\
\end{bmatrix}, \quad \begin{bmatrix}
+ & 0 & + \\
0 & + & 0 \\
0 & + & 0 \\
\end{bmatrix}, \quad \begin{bmatrix}
+ & 0 & + \\
0 & + & 0 \\
0 & + & 0 \\
\end{bmatrix}
\]

Note that each row in the above pattern table describes a general pattern type. The types are intended to determine the partition subsets which are mutually disjoint. For instance, three patterns in the first row define together a general partition pattern type “\( x_1 \) is fully assigned to \( x_{11} \), whereas \( x_2 \) can be freely distributed among \( x_{21} \) and \( x_{23} \)”. If \( x_2 = 1, x_3 \) cannot be divided into two positive parts. In this case, there are two partitions with one part equal to zero. Both of them are represented through the first and second patterns. It is clear that the third pattern is not inapplicable in this case.

Since \( x_1 \) has a fixed assignment, the actual number of partitions is determined by the ways of distributing \( x_2 \) among two destinations, including two ways when one of the destinations gets zero, that is, \( \left( \frac{x_2 + 1}{1} \right) = x_2 + 1 \).

Note also that one pattern in the fourth row means “\( x_1 \) can be freely distributed among \( x_{11} \) and \( x_{12} \) provided that it can be divided into two positive parts, whereas \( x_2 \) is fully assigned to \( x_{23} \)”. It is clear that this partition cannot be applied to the case when \( x_1 = 1 \).
Since the number of partitions is determined by the ways of distributing $x_1$ among two destinations provided that each destination does not get zero, the actual number of partitions is $x_1^{x_1-1}$ if $x_1 = 1$. If $x_1 = 2$, the expected result is zero. Finally, the partition with $x_1 = 1$ and $x_2$ assigned to $x_23$ has already be counted with the second partition patterns in the first and second rows.

The number of partitions when $0 < x_1, x_2 \leq s$ is equal to $3(\frac{x_1-1}{1}) + 3(\frac{x_2+1}{1}) = 3(x_1 + x_2)$ whereas the blocking probability takes the form $p_0^{(3)}(x_1, x_2) = \frac{3(x_1 + x_2)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}}$. The suitable partition patterns with the actual numbers of related partitions when $s < x_1 \leq 2s$ and $0 < x_2 \leq s$ can be represented as follows:

\[
\begin{bmatrix}
  + & + & 0 \\
  0 & 0 & + \\
\end{bmatrix}, \quad (2s - x_1 - 1);
\]

\[
\begin{bmatrix}
  + & 0 & + \\
  0 & + & + \\
\end{bmatrix}, \quad (2s - x_1 - 1);
\]

\[
\begin{bmatrix}
  0 & + & + \\
  + & 0 & 0 \\
\end{bmatrix}, (2s - x_1 - 1).
\]

The number of partitions is equal to $3(2s - x_1 + 1)$, whereas the blocking probability takes the form $p_0^{(3)}(x_1, x_2) = \frac{3(2s - x_1 + 1)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}}$. By obvious symmetry, the blocking probability when $0 < x_1 \leq s$ and $s < x_2 \leq 2s$ takes the form $p_0^{(3)}(x_1, x_2) = \frac{3(2s - x_2 + 1)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}}$.

Finally, it is easy to see that for $s < x_1, x_2 \leq 2s$ the blocking probability is equal to 0 since there is at least one common wavelength with empty time-slots. The summary of results is presented in (3).

\[
p_0^{(3)}(x_1, x_2) = \begin{cases} 
1 & \text{if } x_1 = 0 \text{ and/or } x_2 = 0, \\
\frac{3(x_1+x_2)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}} & \text{if } 0 < x_1 \leq s \text{ and } 0 < x_2 \leq s, \\
\frac{3(2s-x_1+1)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}} & \text{if } s < x_1 \leq 2s \text{ and } 0 < x_2 \leq s, \\
\frac{3(2s-x_2+1)}{\sum_{n=1}^{\text{Naw}} N_{z1n} + \sum_{n=1}^{\text{Naw}} N_{z2n}} & \text{if } s < x_1 \leq 2s \text{ and } 0 < x_2 \leq s, \\
0 & \text{otherwise.}
\end{cases}
\]

By continually applying the partitions principle and the matrix-based scheme, the mathematical expressions of $l = 4, 5, \ldots$ for two links can be obtained. Similarly, the mathematical expressions of $l = 3$ and number of links $>2$, and $l > 3$ and number of links $\geq 2$ can also be obtained. However, as the number of wavelengths and links increases, the natural reasoning method has difficulty to obtain the mathematical expressions because of the large number of groupings. Thus, we propose a schema that is able to calculate the $p_0^{(3)}(x_1, \ldots, x_r)$ for any number of links, any number of wavelengths in each link, and any number of time-slots in each wavelength. In the next section, the proposed schema is presented.

2.2. Schema

Generally, the number of nested while loops in the schema is determined by the number of links; for example, a two link route has two while loops (Section 2.2.1.1), and a three link route has three while loops (Section 2.2.2.1).
The pattern that is used to determine blocking is a one row matrix for each link like \([+ 0]\) for two wavelengths. A + in this case means there is at least one time-slot in wavelength one. A 0 in this case means that there is no time-slot empty in wavelength two. Blocking occurs if there is no + sign in the common wavelength of all the links. The minimum and maximum number of wavelength(s) required to store the number of empty time-slots is determined. The pattern changes as the number of wavelengths in a link is altered. For example, the pattern can be \([+ + 0]\) for three wavelengths in a link and \([+ + + 0]\) for four wavelengths in a link. Each condition such as \(s < x_2 \leq 2s\) can easily be tested using an if-then-else statement to obtain required patterns for a link. For example, the required patterns for number of \(x_1\) empty time-slots distributed in three wavelengths of link \(i\) with condition \(s < x_i \leq 2s\) are \([+ + 0],[0 + +],[+ 0 +] and [+ + +]. These patterns are used to determine the blocking patterns for the numerator of our formula like in Section 2.2.1.1, line 11, and Section 2.2.2.1, line 35. In the same example, the total number of patterns for each link in the denominator is calculated using the formula \(T_{arr} = \sum_{n=1}^{N_{case}} N\times n\). Let \(\alpha\) denote the total number of blocking patterns. Let \(\beta\) denote the total number of blocking and non-blocking patterns. The value of \(p_0^l(x_1, \ldots, x_r)\) is \(\frac{\alpha}{\beta}\). In order to illustrate the workings of the schema, two simple examples – one for two links and the other for three links – are provided.

2.2.1. Example of two links

The first example is that for two links. Each link has two wavelengths, and each wavelength has two time-slots. The value of \(s\) is two because each wavelength can contain two time-slots. The number of empty time-slots for each link is assumed to be two. The minimum number of wavelengths in each link that need to store the empty time-slots is one, and the maximum number of wavelengths in each link that need to store the empty time-slots is two. Hence, there are three different patterns in each link. They are \([+ 0]\) where the empty time-slots are all in wavelength one, \([0 +]\) where the empty time-slots are all in wavelength two, and \([+ +]\) where there is one empty time-slot in wavelength one and another empty time-slot in wavelength two. By applying formula (1), the blocking probability, \(p_0^2(x_1, x_2)\) is \(\frac{2}{(\sum_{n=1}^{N_{case}} N\times 1n)(\sum_{n=1}^{N_{case}} N\times 2n)}\) which is equal to \(\frac{2}{9}\), where the number of empty time-slots for each link is two and number of time-slots in a wavelength is two and number of wavelengths in a link is two. Thus, the value of \(x_1 = 2\) and \(x_2 = 2\) and \(s = 2\) and \(l = 2\). The value of the denominator is 9 because there are three possible patterns for each link which are \([1 1],[2 0]\) and \([0 2]\). Thus, the total number of possible patterns for both links are 9 (3 multiply by 3).

Now, let us explain the workings of the schema to get the same blocking probability value of \(\frac{2}{9}\). We use the formula from the schema pseudo-codes (Section 2.2.1.1, line 11),

\[
\frac{\sum_{n=1}^{N_{case}} N\times 1n}{\sum_{n=1}^{N_{case}} N\times 2n}.
\]

The denominator of this formula is the total number of blocking and non-blocking patterns for the two links. In this case, it is \(\sum_{n=1}^{N_{case}} N\times 1n)(\sum_{n=1}^{N_{case}} N\times 2n)\). The total number of arrangements for each link is calculated to be three. Thus, the total value of the denominator is nine (3 multiply by 3).

The numerator is the value of \(blk\), a floating point variable. The value of \(blk\) is initialized to 0.0. If the pattern in the first link is \([+ 0]\), blocking will only occur if the second link is \([0 +]\) (Section 2.2.1.1, line 6). Since the empty time-slots are stored in one wavelength for both links (Section 2.2.1.1, line 7), \(blk\) is incremented by one (Section 2.2.1.1, line 8). If the pattern in the first link is \([0 +]\) (Section 2.2.1.1, line 6). Since the empty time-slots are stored in one wavelength for both links (Section 2.2.1.1, line 7), \(blk\) is incremented by one (Section 2.2.1.1, line 8). If the pattern in the first link is \([+ +]\), no blocking will occur because there always exists a common wavelength in both links (Section 2.2.1.1, line 6). Thus, the final value of \(blk\) is two. In order to make the schema runs more efficiently, the pattern \([+ +]\) can be omitted because we know that blocking will not occur.

2.2.1.1. Schema pseudo-codes for two links and each link has two wavelengths and each wavelength has two time-slots

Input: Number of available time-slots in each link for \(r\) number of links (\(r\) is >1).

Output: \(p_0^l(x_1, \ldots, x_r)\) for a connection of \(r\) links.
Line

(1) Determine minimum and maximum number of wavelengths required for the empty of time-slots in link \( r - 1 \) and \( r \). /* Number of empty time-slots must be \( \geq 1 \). For two links, the value of \( r \) is 2. */

(2) while (pattern for matching is empty for link \( r - 1 \)) do { // Number of patterns is three for matching two wavelengths, where each wavelength has two time-slots on each link

(3) Get a pattern for link \( r - 1 \) // Available patterns to be matched are \([+ 0], [0 +], [+ +]\)

(4) while (pattern for matching is available for link \( r \)) do {

(5) Get a pattern for link \( r \) // Available patterns to be matched are \([+ 0], [0 +], [+ +]\)

(6) if (no common wavelength exists in both links) // There is a blocking pattern

(7) if (the available time-slots in links \( r - 1 \) and \( r \) are in only one wavelength)

(8) \( blk = blk + 1.0 \)

(9) } // end of second while loop beginning at line 4

(10) } // end of first while loop beginning at line 2

(11) Calculate the blocking probability of \( \frac{blk}{(\sum_{n=1}^{N_{\text{new}}} N_{-x1n})(\sum_{n=1}^{N_{\text{new}}} N_{-x2n})(\sum_{n=1}^{N_{\text{new}}} N_{-x3n})} \).

2.2.2. Example of three links

The second example consists of three links. Each link has two wavelengths, and each wavelength has two time-slots. Thus, the value of \( s \) is two because each wavelength can contain two time-slots. The number of empty time-slots for each link is assumed to be two. The minimum number of wavelengths in each link that need to store the empty time-slots is one and the maximum number of wavelengths in each link that need to store the empty time-slots is two. Hence, the number of patterns for each link is three. They are \([+ 0]\) where the empty time-slots are all in wavelength one, \([0 +]\) where the empty time-slots are all in wavelength two, and \([+ +]\) where there is one empty time-slot in wavelength one and another empty time-slot in wavelength two. By applying formula (2), the blocking probability, \( p_{0}^{(2)}(x_1, x_2, x_3) \), where \( 0 < x_1, x_2, x_3 \leq s \) is \( \frac{2(x_1 + x_2 + x_3)}{(\sum_{n=1}^{N_{\text{new}}} N_{-x1n})(\sum_{n=1}^{N_{\text{new}}} N_{-x2n})(\sum_{n=1}^{N_{\text{new}}} N_{-x3n})} \), which is equal to \( \frac{12}{27} \), where the number of empty time-slots for each link is two. Thus, the value of \( x_1 = 2, x_2 = 2 \) and \( x_3 = 2 \). Now, let us explain the workings of the schema to get the same blocking probability value of \( \frac{12}{27} \). We use the formula from the schema pseudo-codes (Section 2.2.2.1, line 35).

The denominator of our formula is the total number of blocking and non-blocking patterns for the three links. In this case it is \((\sum_{n=1}^{N_{\text{new}}} N_{-x1n})(\sum_{n=1}^{N_{\text{new}}} N_{-x2n})(\sum_{n=1}^{N_{\text{new}}} N_{-x3n})\). The total number of arrangements for each link is calculated to be three. Thus, the total value of the denominator is 27. The numerator is the value of \( blk \), a floating point variable. The value of \( blk \) is initialized to 0.0. The integer variables \( Arr_r, Arr_{r - 1} \) and \( Arr_{r - 2} \) are set to one (Section 2.2.2.1, line 12). Figure 1 shows the 12 suitable partition patterns that allow a connection to be blocked. Each row in a partition pattern represents the pattern for each link. The first column represents wavelength one. The second column represents wavelength two. The first row represents the first link, the second row represents the second link and so on.

![Fig. 1. Suitable partition patterns that causes blocking.](image-url)
Blocking will only occur for all the 12 cases in Fig. 1 (Section 2.2.2.1, line 8). Since the empty time-slots are stored in one wavelength (Section 2.2.2.1, line 9) for all the three links in the first six cases (first row of Fig. 1), \( \text{blk} \) is incremented by one for each case (Section 2.2.2.1, line 10). Thus the value of \( \text{blk} \) is six after evaluating the first six cases. For the next six cases (second row of Fig. 1), there is a pattern \([+ +] \), where one time-slot is stored in wavelength one, and another time-slot is stored in wavelength two. For the seventh \( ( \begin{bmatrix} + & 0 \\ 0 & + \end{bmatrix} ) \) case and eighth \( ( \begin{bmatrix} 0 & + \\ + & 0 \end{bmatrix} ) \) case, the empty time-slots in link one are stored in more than one wavelength (Section 2.2.2.1, line 13). The number of parts for \([+ +] \) is two (Section 2.2.2.1, line 14). Since the number of \( n \) compositions represents the number of empty time-slots, the value of \( n \) is two. As the number of parts is equal to number of \( n \) compositions (Section 2.2.2.1, line 15), the value of \( \text{Arr}_r - 2 \) remains one. For the second and third links, the empty time-slots are stored in only one wavelength so the values of \( \text{Arr}_r - 1 \) and \( \text{Arr}_r \) still remain as one. Because \( \text{Arr}_r, \text{Arr}_r - 1 \) and \( \text{Arr}_r - 2 \) have the value one (Section 2.2.2.1, line 28), \( \text{blk} \) is incremented by one for each of the two cases (Section 2.2.2.1, line 29). Now, the value of \( \text{blk} \) is eight. For the ninth \( ( \begin{bmatrix} 0 & + \\ + & 0 \end{bmatrix} ) \) case and tenth \( ( \begin{bmatrix} 0 & + \\ + & 0 \end{bmatrix} ) \) case, the empty time-slots in link two are stored in more than one wavelength (Section 2.2.2.1, line 18). The number of parts for \([+ +] \) is two (Section 2.2.2.1, line 19). Since the number of \( n \) compositions is the number of empty time-slots, the value of \( n \) is two. As the number of parts is equal to number of \( n \) compositions (Section 2.2.2.1, line 20), the value of \( \text{Arr}_r - 1 \) remains one. For the first and third links, the empty time-slots are stored in only one wavelength so the values of \( \text{Arr}_r - 2 \) and \( \text{Arr}_r \) still remain as one. Because \( \text{Arr}_r, \text{Arr}_r - 1 \) and \( \text{Arr}_r - 2 \) have the value one (Section 2.2.2.1, line 28), \( \text{blk} \) is incremented by one for each of the two cases (Section 2.2.2.1, line 29). Now, the value of \( \text{blk} \) is 10. Similar explanation applies for the last two cases. Finally, the final value of \( \text{blk} \) is 12.

Let a plus (+) represents a wavelength with at least one empty time-slot. In a different situation where the number of \( k \) parts is not equal to number of \( n \) compositions and \( k \) parts is greater than one, the number of patterns for that link can be computationally calculated by calling the function \( \text{Calculate\_patterns}(k, n) \). If a link has two wavelengths with each has at least one empty time-slot ([+ +]) and the total number of empty time-slots is three, the value of \( k \) is two \( (k = 2) \) and \( n \) is three \( (n = 3) \), respectively. Assume that each wavelength contains a maximum of two time-slots \( (s = 2) \). The minimum number of wavelengths that need to store the empty time-slots is two. The maximum number of wavelengths that need to store the empty time-slots is also two because each link has only two wavelengths. The total number of patterns calculated is 2. The two cases are \([2 \ 1]\), where there are two empty time-slots in wavelength one and one empty time-slot in wavelength two and \([1 \ 2]\), where there are one empty time-slot in wavelength one and two empty time-slots in wavelength two. The cases \([3 \ 0]\) and \([0 \ 3]\) are not included because the number of empty-time slots in each wavelength should not be more than two.

### 2.2.2.1. Schema pseudo-codes for three links and each link has two wavelengths and each wavelength has two time-slots

**Input:** Number of empty time-slots in each link for \( r \) number of links (\( r > 1 \)).

**Output:** \( p_{ij}^{(1)}(x_1, \ldots, x_r) \) for a connection of \( r \) links.

**Line**

(1) Determine minimum and maximum number of wavelengths required for the empty of time-slots in links \( r - 2, r - 1 \) and \( r \). Number of empty time-slots must be \( \geq 1 \). For three links, the value of \( r \) is 3.

(2) while (pattern for matching is available for link \( r - 2 \)) do {
   // Number of patterns is three for matching two wavelengths, where each wavelength has two time-slots on each link
   (3) Get a pattern for link \( r - 2 \) // Available patterns to be matched are [+ 0], [0 +], [+] +]
   (4) while (pattern for matching is available for link \( r - 1 \)) do {
Number of patterns is three for matching two wavelengths, where each wavelength has two time-slots on each link.

(5) Get a pattern for link \( r - 1 \) // Available patterns to be matched are \([+ 0], 0 +, ++\]

(6) while (pattern for matching is available for link \( r \)) do {
  // Number of patterns for matching is three for two wavelengths, where each wavelength has two time-slots on each link
  (7) Get a pattern for link \( r \) // Available patterns to be matched are \([+ 0], 0 +, ++\]
  (8) if (no common wavelength exists in the three links) // There is a blocking pattern
  (9) if (available time-slots in links \( r - 2, r - 1 \) and \( r \) are in only one wavelength)
  (10) \( \text{blk} = \text{blk} + 1.0 \)
  (11) else {
    (12) \( \text{Arr}_r = \text{Arr}_r - 1 = \text{Arr}_r - 2 = 1 \)
    (13) if (empty time-slots in link \( r - 2 \) is in more than one wavelength)
    (14) Determine the number of \( k \) parts
    /* Value of \( k \) is the number wavelengths that has at least one empty time-slot and \( n \) compositions is the number of empty time-slots in link \( r - 2 \)*/
    (15) if (\( k \) parts \(!= \) number of \( n \) compositions)
    (16) \( \text{Arr}_r = \text{Calculate_patterns}(k, n) \);
    (17) } // End of if statement beginning at line 13
    (18) if (empty time-slots in link \( r - 1 \) is in more than one wavelength)
    (19) Determine the number of \( k \) parts
    /* Value of \( k \) is the number wavelengths that has at least one empty time-slot and \( n \) compositions is the number of empty time-slots in link \( r - 1 \)*/
    (20) if (\( k \) parts \(!= \) number of \( n \) compositions)
    (21) \( \text{Arr}_r - 1 = \text{Calculate_patterns}(k, n) \);
    (22) } // End of if statement beginning at line 18
    (23) if (empty time-slots in link \( r \) is in more than one wavelength)
    (24) Determine the number of \( k \) parts
    /* Value of \( k \) is the number wavelengths that has at least one empty time-slot and \( n \) compositions is the number of empty time-slots in link \( r \)*/
    (25) if (\( k \) parts \(!= \) number of \( n \) compositions)
    (26) \( \text{Arr}_r - 2 = \text{Calculate_patterns}(k, n) \);
    (27) } // End of if statement beginning at line 23
    (28) if (\( \text{Arr}_r = \text{Arr}_r - 1 = \text{Arr}_r - 2 = 1 \))
    (29) \( \text{blk} = \text{blk} + 1.0 \)
    (30) else
    (31) } // End of if statement beginning at line 8
    (32) } // End of third while loop beginning at line 6
    (33) } // End of second while loop beginning at line 4
    (34) } // End of first while loop beginning at line 2
    (35) Calculate the blocking probability of
    \[
    \text{int Calculate_patterns}(k, n)\{ \\
    \frac{\sum_{n=1}^{N_{case}} N_{x1n}!}{\sum_{n=1}^{N_{case}} N_{x2n}! \sum_{n=1}^{N_{case}} N_{x3n}!}
    \]
    /* \( s \) is the maximum number of time-slots in each wavelength, \( k \) is the number of wavelengths in each link that has at least one empty time-slot, and \( n \) is the number of empty time-slots for that link. Let a plus (+) represents a wavelength that has at least one empty time-slot.*/
    
(1) Put a one in each wavelength that has a plus (+) since a plus (+) in a wavelength represents at least one empty time-slot. Assume that there are \( x \) such wavelengths. This means that \( k = x \).
(2) Calculate the number of unique cases for distributing \( n - x \) empty time-slots in \( k \) parts where each part is not more than \( s - 1 \) because we have already put a one in each wavelength that has a plus in it. For each unique case, the number of empty time-slots in \( x \) wavelengths of a link compared with the rest of the cases must not be the same regardless of the arrangement position.

(3) For each unique case, calculate the number of ways of distributing \( n - x \) empty time-slots in \( k \) parts. The numerator is the factorial of \( x \) wavelengths in a link. The denominator is the product of the factorial of each unique number of empty time-slots in \( x \) wavelengths. If the number of empty time-slots in \( x \) wavelengths of a link occurs only once, we have 1!. A 2! means that the number of empty time-slots in \( x \) wavelengths of a link occurs twice. Similar explanation is used for 3!, 4! and so on.

(4) Add the value for each unique case.

(5) Return the total value.

3. Calculating approximate blocking probabilities

3.1. Fixed time-slot wavelength routing

Let us consider a network of arbitrary topology with \( J \) links and \( C \) time-slots on each link. The total number of time-slots (\( C \)) of each link is equivalent to the number of time-slots of each wavelength multiplied by the number of wavelengths per link. A route \( R \) is a subset of links from \( \{1, \ldots, J\} \). Calls arrive for route \( R \) as a Poisson stream with rate \( a_R \). A call is accepted if it is assigned a time-slot in a wavelength \( w_i \) on all links in route \( R \). Let \( X_j \) be the random variable denoting the number of idle time-slots on link \( j \) in equilibrium. Let \( X = (X_1, \ldots, X_J), q_j(t) = \Pr[X_j = t] \); and \( t = 0, \ldots, C \) be the idle capacity distribution. The approximation is made where the random variables \( X_1, \ldots, X_J \) are mutually independent. Then \( q_j(t) = \prod_{j=1}^{J} q_j(t_j) \), where \( t = (t_1, t_2, \ldots, t_J) \). The next approximation is that there are \( t \) idle time-slots on link \( j \); the time until the next call is set up on link \( j \) is exponentially distributed with \( \alpha_j(t) \). This parameter is the call set-up rate on link \( j \) when \( t \) time-slots are free on link \( j \). It follows that the number of idle time-slots on link \( j \), and this can be viewed as a birth-and-death process as shown in Fig. 2. Each state \( m \) is the number of idle time-slots on link \( j \). The death rate is \( C - m \) at state \( m \). By solving the Markov chain, we have

\[
q_j(t) = \frac{C(C - 1) \cdots (C - t + 1)}{\alpha_j(1) \alpha_j(2) \cdots \alpha_j(t)} q_j(0),
\]

where \( t = 1, \ldots, C \) and

\[
q_j(0) = \left[ 1 + \sum_{t=1}^{C} \frac{C(C - 1) \cdots (C - t + 1)}{\alpha_j(1) \alpha_j(2) \cdots \alpha_j(t)} \right]^{-1}.
\]

Fig. 2. Birth-and-death process for idle time-slots distribution on link \( j \).
The call set-up rate on link $j$ when there are $t$ idle time-slots on link $j$, $\alpha_j(t)$, is obtained by combining the contributions from the request streams to routes of which link $j$ is a member:

$$\alpha_j(t) = \begin{cases} 0 & \text{if } t = 0 \\ \sum_{R: j \in R} a_R \Pr[X_R > 0 | X_j = t], & \text{if } t = 1, \ldots, C. \end{cases} \tag{6}$$

If the route consists of one link, the probability term $\Pr(\cdot)$ under the summation sign in (6) will be equal to 1. If the route consists of two links, let $R = \{i, j\}$. The term $\Pr(\cdot)$ can be further simplified by conditioning it on the set of disjoint events $\{X_i = l | l = 0, \ldots, C\}$.

$$\Pr[X_{\{i, j\}} > 0 | X_j = t] = \sum_{l=1}^C \Pr[X_i = l | X_j = t] \Pr[X_R > 0 | X_j = t, X_i = l]$$

$$= \sum_{l=1}^C q_i(l)(1 - p_0(t, l)), \tag{7}$$

where $p_0(t, l)$ may be given by (1) or (3) or the proposed schema.

Similarly, for a three-hop route $R = \{i, j, k\}$, and routes more than three hops, the probabilities can be obtained. For a three-hop route with two wavelengths per link, $p_0(l, t, n)$ is given by (2) or the proposed schema.

3.2. Blocking probability for route $R$

The blocking probability for calls to route $R$ is

$$L_R = \Pr[X_R = 0]$$

$$= q_i(0) \quad \text{if } R = \{i\}$$

$$= \sum_{l=0}^C \sum_{t=0}^C q_i(l)q_j(t)p_0(l, t) \quad \text{if } R = \{i, j\} \tag{8}$$

$$= \sum_{l=0}^C \sum_{t=0}^C \sum_{n=0}^C q_i(l)q_j(t)q_k(n)p_0(l, t, n) \quad \text{if } R = \{i, j, k\}$$

$$= \sum_{l=0}^C \sum_{t=0}^C \sum_{n=0}^C \sum_{m=0}^C q_i(l)q_j(t)q_k(n)q_r(m)p_0(l, t, n, m) \quad \text{if } R = \{i, j, k, r\}.$$
Step 2. Determine $q_j(\cdot)$ from (4) and (5). This is to determine the idle capacity distribution when link $j = 1, \ldots, J$.

Step 3. Obtain new values of $\alpha_j(\cdot), j = 1, \ldots, J$, using (6). (For two-hop routes, (7) is used instead of (6) and suitable generalizations are used for paths with more than two hops.)

Step 4. Determine $L_R$, for all routes $R$ using (8). If $\max_R |L_R - \overline{L_R}| < \varepsilon$ (where $\varepsilon$ is suitably small positive value like 0.00001), then terminate else let $\overline{L_R} = L_R$, and go to step 2.

Step 5. The average network blocking probability is then given by

$$\Pr_{\text{network}} = \frac{\sum_{i=1}^{R} a_i L_i}{a_{\text{network}}},$$

where $a_{\text{network}}$ is the network offered load.

4. Numerical results

The blocking performance simulation is done using OPNET for a network of seven nodes and nine links (Fig. 3) and the famous NSFNET (14 nodes, 21 links) topology (Fig. 4). At the first hop, a wavelength and a time-slot in the selected wavelength are randomly assigned. At subsequent hop(s), a time-slot in a wavelength is assigned based on the First-Fit heuristic since no wavelength converter is used. Simulated results are generated many times to obtain a 95% confidence interval. The reduced load algorithm is used in our method.

For the seven nodes and nine links network (Fig. 3), we consider all the possible 18 routes. Tables 1 and 2 show the individual route blocking probabilities. $R$ denotes the set of links. $L_R^{\text{sim}}$ (%) denotes the blocking probability obtained by simulation. $L_R$ (%) denotes the blocking probability calculated analytically. Simulation results are run 30 times to get a 95% confidence interval. Each link has eight wavelengths and each wavelength has eight time-slots. The total network offered load for Table 1 is 5 Erlangs. Since the load is uniformly distributed on all the

Fig. 3. Seven nodes and nine links network.

Fig. 4. NSFNET topology.
Table 1
Total offered load of 5 Erlangs

<table>
<thead>
<tr>
<th>No.</th>
<th>R</th>
<th>(L_{RL} (%))</th>
<th>(L_R (%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>(0.0062, 0.0101)</td>
<td>0.0087</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>(0.0171, 0.0310)</td>
<td>0.0191</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>(0.0172, 0.0304)</td>
<td>0.0190</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>(0.0049, 0.0060)</td>
<td>0.0055</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>(0.0030, 0.0045)</td>
<td>0.0043</td>
</tr>
<tr>
<td>6</td>
<td>{6}</td>
<td>(0.0000, 0.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>7</td>
<td>{7}</td>
<td>(0.0041, 0.0075)</td>
<td>0.0070</td>
</tr>
<tr>
<td>8</td>
<td>{8}</td>
<td>(0.0000, 0.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>9</td>
<td>{9}</td>
<td>(0.0000, 0.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>{1, 2}</td>
<td>(0.0347, 0.0501)</td>
<td>0.0325</td>
</tr>
<tr>
<td>11</td>
<td>{5, 7}</td>
<td>(0.0196, 0.0283)</td>
<td>0.0245</td>
</tr>
<tr>
<td>12</td>
<td>{2, 3}</td>
<td>(0.0225, 0.0321)</td>
<td>0.0245</td>
</tr>
<tr>
<td>13</td>
<td>{7, 8}</td>
<td>(0.0228, 0.0262)</td>
<td>0.0254</td>
</tr>
<tr>
<td>14</td>
<td>{3, 4}</td>
<td>(0.0221, 0.0332)</td>
<td>0.0323</td>
</tr>
<tr>
<td>15</td>
<td>{6, 7}</td>
<td>(0.0101, 0.0227)</td>
<td>0.0136</td>
</tr>
<tr>
<td>16</td>
<td>{1, 2, 3}</td>
<td>(0.0521, 0.0678)</td>
<td>0.0516</td>
</tr>
<tr>
<td>17</td>
<td>{4, 5, 7}</td>
<td>(0.0495, 0.0673)</td>
<td>0.0545</td>
</tr>
<tr>
<td>18</td>
<td>{1, 2, 3, 4}</td>
<td>(0.0637, 0.0887)</td>
<td>0.0801</td>
</tr>
</tbody>
</table>

Table 2
Total offered load of 10 Erlangs

<table>
<thead>
<tr>
<th>No.</th>
<th>R</th>
<th>(L_{RL} (%))</th>
<th>(L_R (%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{1}</td>
<td>(0.0118, 0.0204)</td>
<td>0.0120</td>
</tr>
<tr>
<td>2</td>
<td>{2}</td>
<td>(0.0313, 0.0607)</td>
<td>0.0352</td>
</tr>
<tr>
<td>3</td>
<td>{3}</td>
<td>(0.0330, 0.0612)</td>
<td>0.0352</td>
</tr>
<tr>
<td>4</td>
<td>{4}</td>
<td>(0.0064, 0.0092)</td>
<td>0.0090</td>
</tr>
<tr>
<td>5</td>
<td>{5}</td>
<td>(0.0058, 0.0075)</td>
<td>0.0071</td>
</tr>
<tr>
<td>6</td>
<td>{6}</td>
<td>(0.0021, 0.0042)</td>
<td>0.0028</td>
</tr>
<tr>
<td>7</td>
<td>{7}</td>
<td>(0.0095, 0.0145)</td>
<td>0.0122</td>
</tr>
<tr>
<td>8</td>
<td>{8}</td>
<td>(0.0000, 0.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>9</td>
<td>{9}</td>
<td>(0.0000, 0.0000)</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>{1, 2}</td>
<td>(0.0702, 0.0881)</td>
<td>0.0685</td>
</tr>
<tr>
<td>11</td>
<td>{5, 7}</td>
<td>(0.0383, 0.0534)</td>
<td>0.0449</td>
</tr>
<tr>
<td>12</td>
<td>{2, 3}</td>
<td>(0.0402, 0.0597)</td>
<td>0.0450</td>
</tr>
<tr>
<td>13</td>
<td>{7, 8}</td>
<td>(0.0441, 0.0512)</td>
<td>0.0487</td>
</tr>
<tr>
<td>14</td>
<td>{3, 4}</td>
<td>(0.0424, 0.0653)</td>
<td>0.0630</td>
</tr>
<tr>
<td>15</td>
<td>{6, 7}</td>
<td>(0.0221, 0.0342)</td>
<td>0.0291</td>
</tr>
<tr>
<td>16</td>
<td>{1, 2, 3}</td>
<td>(0.1017, 0.1388)</td>
<td>0.1045</td>
</tr>
<tr>
<td>17</td>
<td>{4, 5, 7}</td>
<td>(0.1036, 0.1337)</td>
<td>0.1070</td>
</tr>
<tr>
<td>18</td>
<td>{1, 2, 3, 4}</td>
<td>(0.1263, 0.1753)</td>
<td>0.1284</td>
</tr>
</tbody>
</table>

routes, each route has a load of 0.277 Erlangs. The total network offered load for Table 2 is 10 Erlangs. The load is uniformly distributed on all the routes. Thus, each route has a load 0.555 Erlangs. We observe that there are good accuracies in the analytical results for a total network load of 5 Erlangs and a total network load of 10 Erlangs.

For Figs 5–8, routes are constructed using minimum fixed hop routing. The loads to all routes are assumed to
be the same. Routes are considered from different parts of the network. If multiple routes are present between two node pairs, at most three routes are randomly selected. The simulated and analytical results of Figs 5–8 are for NSFNET topology. For Figs 5 and 6, we consider 42 one-hop, 28 two-hop and 28 three-hop routes. For Fig. 5, similar results are obtained using the derived mathematical expressions and the proposed schema. Figure 5 shows the simulation and analytical results of (i) two wavelengths per link, and each wavelength has eight time-slots ($W = 2, T = 8$), and (ii) two wavelengths per link, and each wavelength each wavelength has 10 time-slots ($W = 2, T = 10$). We plot the average network probability for these routes versus the total offered load to the network. The analytical results of the proposed model are in agreement with the simulated results. For the purpose of analysis, we compare the simulated and analytical results of three wavelengths. This is shown in Fig. 6. The figure shows the simulated and analytical results of three wavelengths per link, and each wavelength has eight time-slots ($W = 3, T = 8$). We plot the average network blocking probability for these routes versus the total
offered load to the network. As the offered load (Erlangs) increases, the simulation results matches well with the analytical results.

For Figs 7 and 8, we consider 42 one-hop, 28 two-hop, 28 three-hop and four-hop routes. Figure 7 shows the simulated and analytical results of eight wavelengths per link, and each wavelength has eight time-slots ($W = 8, T = 8$). Figure 8 shows the simulated and analytical results of 16 wavelengths per link, and each wavelength has four time-slots ($W = 16, T = 4$). For Figs 7 and 8, overall good accuracies are observed when the simulated and analytical results are compared.

5. Conclusion

In this paper, an analytical model to calculate the individual routes and average network blocking probabilities for TDM WDM optical networks with OTSIs and without WC is considered. To make the analysis tractable, we proposed a schema that can work for any number of partition patterns regardless of numbers of links, wavelengths in each link, and time-slots in each wavelength. We use the reduced load approximation method for state dependent routing. When compared with the simulated results, our analytical model provides good accuracies in average network blocking probability for minimum fixed hop routing. Since the results of our model are shown to be close to the simulated results, network designers can use our model to predict the blocking probabilities of similar networks.

References


Enhancement of packet reordering in a mobile stream control transmission protocol for a heterogeneous wireless network vertical handover

Bashar J. Hamza *, Chee Kyun Ng, N.K. Noordin, M.F.A. Rasid, A. Ismail and Yaseen H. Tahir

Department of Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia, Selangor, Malaysia
E-mails: mrym_bashar@yahoo.com, {mpnck, nknordin, fadlee, alyani}@eng.upm.edu.my, yahasan@yahoo.com

Abstract. Future wireless access networks will be heterogeneous wireless network (HWN) environment which consists of various wireless technologies including universal mobile telecommunications system (UMTS) networks and wireless local area networks (WLAN). They are used together through vertical handover (VHO) to ensure global mobility and service continuity. The mobile stream control transmission protocol (mSCTP) layer supports dynamic association reconfiguration. This protocol allows mSCTP endpoints to dynamically add, change and delete IP addresses when the mobile node (MN) is switched between HWNs. During a mSCTP handover, the endpoints of the mSCTP are required to change the primary link from an old link to a UMTS into a new link to a WLAN. However, due to the disparity between UMTS/WLAN bandwidth, a packet reordering problem will occur when the MN of the mSCTP leaves to a new network. This packet reordering problem can then cause additional drawbacks such as impossibility of growing an mSCTP congestion window, unnecessary fast retransmissions, actual packet losses, and reduced efficiency of the receiving mSCTP. In this paper, we propose a packet reordering model (PRM) that is inserted inside the MN, and works as a special buffer of a large capacity with one input and one output port to receive all transmission sequence numbers (TSNs). It then forwards all incoming data chunks to the MN/WLAN networks after the VHO. The performance of the system is simulated and analyzed using NS-2 simulation tool. The simulation results show that the suggested model enhances the performance throughput and the congestion window of the conventional mSCTP through VHO by handling the packet reordering problem. In other words, the average performance throughput of the proposed PRM scheme is 181.48 Kbps or 16% increment compared to conventional mSCTP at 165.54 Kbps.

Keywords: Vertical handover, 3G, WLAN and UMTS, mSCTP, PRM

1. Introduction

Integration of third-generation (3G) cellular networks such as the universal mobile telecommunications system (UMTS) and wireless local area networks (WLAN) has been intensively investigated in recent years because of their complementary characteristics. IEEE 802.11 WLAN has been widely deployed in offices, homes, campus, airports and hotels given its low communication cost, high data rate (11 Mbits/s), and ease of deployment. However, a serious disadvantage of 802.11 is the small coverage area (up to 300 m) and low mobility [6,17]. The most known standards belong to the IEEE 802.11 WLAN family, which includes the popular 802.11b, the 802.11a and the 802.11g as shown in Table 1.

International Telecommunication Union (ITU) defines 3G as devices that can transfer data up to 384 Kbps. As comparison, the global system for mobile communications (GSM) bandwidth is up to 14.4 Kbps and general packet
Table 1
IEEE 802.11 WLAN family

<table>
<thead>
<tr>
<th>802.11</th>
<th>Date</th>
<th>Freq. Band and Mod.</th>
<th>Throughput (typical) (Mbit/s)</th>
<th>Net bit rate (Mbit/s)</th>
<th>Range (indoor) (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11b</td>
<td>October 99</td>
<td>2.4 GHz/DSSS</td>
<td>∼5</td>
<td>11</td>
<td>∼38</td>
</tr>
<tr>
<td>802.11a</td>
<td>October 99</td>
<td>5 GHz/OFDM</td>
<td>27</td>
<td>54</td>
<td>∼35</td>
</tr>
<tr>
<td>802.11g</td>
<td>June 03</td>
<td>2.4 GHz/DSSS or OFDM</td>
<td>∼22</td>
<td>54</td>
<td>∼up to 100</td>
</tr>
</tbody>
</table>

Fig. 1. UMTS architecture. (The colors are visible in the online version of the article.)

radio service (GPRS) bandwidth is around 53.6 Kbps. Both are used in 2G and 2.5G, respectively [6]. UMTS is a 3G wireless protocol that is part of the ITU. UMTS is expected to deliver low-cost, high-capacity mobile communications, offering data rates of about 1 Mbps. The wireless radio access network for UMTS contains user equipment (UE) and UMTS terrestrial radio access network (UTRAN), which includes the node-B and radio network controller (RNC) [6,17,21]. The packet domain core network includes the serving GPRS support node (SGSN) and the gateway GPRS support node (GGSN) as shown in Fig. 1.

The complementary characteristics of UMTS and WLAN suggest the possibility of combining these two wireless technologies. WLANs offer low mobility with much higher data rates compared to mobile nodes and a low communication cost over a geographically small area, while UMTS networks provide relatively low data rates with high connectivity and high mobility to mobile nodes but the communication cost is relatively high [6,17,21]. To achieve an ongoing continuous connection for a mobile node (MN) that leaves across the heterogeneous wireless network (HWN) regions a vertical handover (VHO) is required. This is contrary to horizontal handover (HHO) that is occurred between homogeneous networks of a wireless access system.

The type of handover between UMTS and WLAN networks is known as a VHO [15]. This VHO provides seamless internet access for the MN that switches from one wireless network to another as shown in Fig. 2. Towards this end, some new mechanism for smooth mobility management is needed to reduce the long VHO latency between UMTS/WLAN networks. The radio resources must be professionally treated to guarantee the delivery of data between two MNs between UMTS/WLAN networks. Thus, VHOs are executed across various wireless networks, which differ in several aspects such as communication cost, bandwidth, frequency of operation and data rate. To support the ongoing VHO research into UMTS/WLAN networks, the stream control transmission protocol (SCTP) was previously proposed to minimize transition time during VHO over HWNs [2,5]. In particular, SCTP has a highly reliable transport layer protocol which is placed at the side of the user datagram protocol (UDP) and transmission control protocol (TCP) [11,12,16,18]. It provides ordered delivery and stability of data between two endpoints similar to UDP and TCP. Unlike UDP and TCP, the SCTP offers such advantages as multihoming and multistreaming capabilities.

A multihoming host can use more than one network layer address to communicate. For a MN, it can use multiple network interfaces simultaneously. If the primary path fails, the protocol will send traffic over the alternate path. Multistreaming can be used to deliver multiple objects (webpage, audio, video and text) that belong to the same association independently. Each stream is given a stream number that is encoded inside SCTP packets flowing through the association. Hence, SCTP enables its endpoints to be installed across multiple wireless interfaces that are recognized by many IP addresses [4]. SCTP usually transmits data chunks to a destination IP address appointed by the main address. It can redirect the data chunks to an alternative IP address if the main IP address becomes
unreachable. The link between two MNs that employ the main IP address is known as the main link and the link between two MNs using an alternative IP address is known as the secondary link. Note that two MNs in SCTP can have only one main link, but more than one secondary link. This kind of data session is explained in SCTP [4,11,12,18,19].

In order to support ongoing VHO research into UMTS/WLAN networks, a mobile SCTP (mSCTP) has recently been proposed by internet engineering task force (IETF) [11,12]. The mSCTP VHO supports dynamic address reconfiguration (ASCONF), which allows the MN to dynamically add, change and delete IP addresses when MN travels between HWNs. However, in such different wireless network technologies, the radio interface may have asymmetric features that lead to the degradation of reception of the data chunks between the MNs. This results in packet reordering problem. In mSCTP VHO, the packet reordering problem can arise from the disparity between the propagation delay and the bandwidth between UMTS/WLAN networks when MN moves across them. Moreover, this packet reordering problem leads to additional negative side effects such as the impossibility of growing an mSCTP congestion window, actual packet losses, unnecessary fast retransmissions and reduced efficiency of the receiving mSCTP [9,14].

There are some recent approaches aimed to overcome the mSCTP handover drawbacks in HWNs. Ma et al. in [14] presented a new method to make possible the VHO between wide region cellular data networks such as UMTS and WLANs employing SCTP. The dynamic address configuration extension and multihoming capability of SCTP has been used in the overlay architecture of UMTS/WLAN to reduce the VHO delay and to improve throughput performance. Experimental results show that the suggested scheme can avoid the problem of a long latency during VHO, particularly when using the dual-homing SCTP configuration.

Keun et al. in [9] present an SCTP Efficient Flow Control (SCTP-EFC) mechanism through a VHO by enabling a mobile client to freely switch between IP addresses acquired in different networks. The SCTP-EFC reduces the change of traffic data rates and improves the throughput performance significantly during a VHO. The result shows that SCTP-EFC adjusts to a network wireless environment after VHO and gives a throughput enhancement for a few seconds after a VHO.

Seok et al. in [3] dealt with the packet reordering problem which occurs during the mSCTP handover. However, the researchers only focused on the unnecessary retransmission at the endpoints of the mSCTP. They solved this
problem by using a new model in the corresponding nodes (CN) to resend the outstanding data chunks before sending the newly generated data chunks across the new link. Huang et al. in [8] proposed a solution to avoid the packet reordering problem using a mobile multipath SCTP (m²-SCTP) that allows a MN that desires to send data to use UMTS/WLAN multiple paths. The result shows that m²-SCTP achieves a throughput enhancement during a VHO.

In this paper, we propose a packet reordering model (PRM) to solve the packet reordering problem that is inherent in the mSCTP VHO between MNs by increasing the congestion window (CWND) of the conventional mSCTP in a HWN. This new PRM scheme is used to hold all the outstanding data chunks. They are then forwarded to the MN if it becomes free during the round-trip time (RTT) interval. After that the CWND will be increased when mSCTP MN moves across UMTS/WLAN networks. The simulation results show that the proposed PRM scheme has a higher performance throughout in the mSCTP VHO and has a larger CWND than the conventional mSCTP. In other words, the proposed PRM model has an advantage in some aspects; increasing the throughput by about 16% compared to the conventional scheme, reducing the VHO delay and achievement of seamless VHO.

The rest of this paper is organized as follows: Section 2 discusses the mSCTP handover while Section 3 introduces the packet reordering problem during a vertical handover. Section 4 covers the packet reordering model work to enhance the vertical handover between UMTS/WLAN networks. Simulation of the proposed approach is presented in Section 5. Finally, we conclude the paper in Section 6.

2. Mobile stream control transmission protocol (mSCTP) handover

The mSCTP is an order of data packet delivery between two endpoints. It has a highly reliable transport layer protocol that provides stability (similar to TCP) and also preserves the boundaries of the data message (similar to UDP). However, unlike TCP and UDP, mSCTP offers advantages such as the capabilities of multihoming and multistreaming, which both enhance reliability and availability. The mSCTP with dynamic association reconfiguration (DAR) allows IP addresses to be added and removed from an SCTP association, meaning that data packets can then be transmitted to the new destination. Through the VHO, the endpoints of the mSCTP are needed to change the main link from the old link of cellular UMTS network to a new link of the UMTS network as shown in Fig. 3.

Figure 3 shows the mSCTP VHO between UMTS/WLAN networks, where the MN first travels into the WLAN and then leaves away again from it. We consider the MN that initiates an mSCTP data session with any CN. The following steps are executed when the MN travels into a WLAN network [19]:

Step 1: Obtain an IP address for a new location from the new access router of the WLAN by transmitting an mSCTP address configuration change ASCONF data chunk. The MN may receive a replying ASCONF-ACK data chunk from the CN.

Step 2: Change the main IP address by transmitting an ASCONF data chunk with a set main IP address. In response, the CN replies with an ASCONF-ACK data chunk.

Step 3: Delete the old IP address after the MN moves away from the coverage of the WLAN network by transmitting an ASCONF data chunk with a delete IP address. The MN receives the ASCONF-ACK data chunk reply from the CN [4,11,12,14,19].

3. Packet reordering problem during a VHO

When an mSCTP MN moves across UMTS/WLAN networks, the MN should change its main link to a new link with the WLAN with a higher data packet rate and/or lesser transmission delay. Unexpected packet reordering may occur due to the disparity of propagation delay and bandwidth between UMTS/WLAN networks [3,8]. Packet reordering in a UMTS/WLAN network VHO is caused by the data chunks that are transmitted using the cellular UMTS link may arrive at the WLAN before VHO. It is also possible that these data chunks may arrive later than the data chunks that are transmitted using the link of the WLAN after VHO. This scenario is shown in Fig. 4.
The packet reordering problem then causes additional negative side effects such as the impossibility of growing an mSCTP CWND, unnecessary fast retransmissions, packet losses and reduced efficiency of the receiving mSCTP. Note that the packet reordering problem is occurred only when an mSCTP MN moves from UMTS network to WLAN network. This is because the data rate of UMTS network is smaller than WLAN data rate. For example, the four data chunks of transmission sequence number (TSN) such as TSN1, TSN2, TSN3 and TSN4 are sent from the CN to a MN/UMTS during which the link become congested before the VHO as shown in Fig. 4. We assume that the MN moves from a cellular UMTS to a WLAN network where an IP1 address was used in the cellular network of the UMTS and an IP2 address is used in the WLAN network. Hence, these four data chunks (TSN1, TSN2, TSN3 and TSN4) are transmitted before the main link is changed from the UMTS-IP1 to the WLAN-IP2 address [10,13]. After the main link is changed, the CN transmits two new data chunks (TSN5 and TSN6) to the new destination of the MN/WLAN, which has much better conditions than the old link of the cellular UMTS. Since the traffic data rate of the WLAN is much better than that of the cellular UMTS, therefore, the two data chunks (TSN5 and TSN6) may reach at the WLAN sooner than the four data chunks (TSN1, TSN2, TSN3 and TSN4), which will cause a packet reordering problem. Moreover, the problem of packet reordering may reduce the CWND for the WLAN-IP2 address and degrade the utilization of the mSCTP VHO [20].

4. Proposed packet reordering model (PRM)

To solve the packet reordering problem, we propose a PRM in which the MN can enhance the VHO performance and the CWND of a conventional mSCTP in HWNs. This new scheme of PRM is used to hold all the outstanding
data chunks. They are then forwarded to the MN if it becomes free during an RTT interval. After that the CWND will be increased when the mSCTP MN moves across UMTS/WLAN networks. The PRM works as a special buffer that has a large capacity with one input and one output port to receive and hold all TSNs during the VHO. It will then forward all incoming data chunks to the MN/WLAN-IP2 after the VHO.

Figure 5 shows the algorithm used in a PRM to solve the packet reordering problem and avoid reduction of CWND when VHO occurs. After VHO occurs the PRM has information about all the outstanding data chunks. If these outstanding data chunks arrive when the retransmission timer expires, the PRM holds the outstanding data chunks and forwards them to the MN if it becomes free during an RTT in order to avoid a reduction of the CWND. Otherwise, the outstanding data chunks will prompt the PRM to initialize two jobs. First, it sends an acknowledgment to the CN to retransmit the outstanding data chunks, and it also increases the CWND by an amount greater than the slow start/avoidance congestion mode by the number of the outstanding data chunks. The second job is to avoid the reduction of the CWND where the PRM repeats the previous hold/forward procedure.

Slow start and avoidance congestion are modes that mSCTP uses to control congestion inside the network. When a MN begins to send data it is required to use the slow start algorithm at the beginning of the transfer. The congestion is controlled by increasing the mSCTP congestion window for each acknowledgment that is received. The value of CWND is increased by one TSN. It continues until either the CWND reaches the maximum window size or packet loss is detected. In contrary, the avoidance congestion mode deals with packet loss which occurs due to congestion of a network. When the maximum window size is reached, the avoidance congestion mode will increase the CWND by one TSN for each RTT.

In part A of Fig. 5, the CN labels all the outstanding TSNs by employing the added variable “out_ TSNn” added by the proposed scheme when it receives the data chunks of the ASCONF. Part B shows an “out_counter”
(A) Corresponding Node (CN) Side Behaviour
To determine the outstanding $TSN_n$ when vertical handover occurs
On receipt of vertical handover $ASCONF_ACK$ data chunk
   for each outstanding $TSN_n$ do
      $out_{TSN_n} = \text{TRUE}$
   end

(B) Packet Reordering (PRM) Model Side Behaviour
To calculate the amount of outstanding $TSN_n$ that are sent before vertical handover occurs
   reset $out_{counter} = 0$
   for each outstanding $TSN_n$
      increment $out_{counter}$ by 1
   end
To handling the outstanding $TSN_n$ that arrived before expiration of retransmission timer occurs
   for each outstanding $TSN_n$
      if outstanding $TSN_n$ receipt before expiration of retransmission timer occurs
         add $ack$ of outstanding $TSN_n$ into next SACK
         decrease $out_{counter}$ by 1
         hold outstanding $TSN_n$ then forward it to MN if it becomes free
      end
   end

(C) CN Side Behaviour
On receipt the current $SACK$
   Increment congestion window (CWND) by slow start/congestion avoidance mode + $out_{counter}$
   Sent the newly data chunks and retransmission the outstanding data chunks

(D) PRM Side Behaviour
   for each $TSN_n$
      if $out_{TSN_n} = \text{TRUE}$
         hold $TSN_n$ then forward it to MN if it becomes free
      end
   end

Fig. 5. Algorithm of the proposed PRM scheme.

variable is employed to compute the amount of outstanding $TSN_n$ that are sent before VHO occurs. This part is used to handle the outstanding $TSN_n$ that arrived before the expiration of the retransmission time occurs. Part C explains the growth of the CWND by the slow start/avoidance congestion mode plus the “out_counter” variable when the CN receives the current selective acknowledgment (SACK). SACKs have three functions; acknowledge data received, track the RTT of a path which is used calculate the retransmission timeout (RTO), and monitor the status of each path. In this part, the CN will transmit the new created data chunks and retransmit all the outstanding $TSN_n$ to the MN/WLAN-IP2 address when expiration of retransmission time occurs. In part D, the PRM checks whether the $TSN_n$ are new or outstanding data chunks in order to forward or hold/forward respectively when the MN becomes free. In this way, unnecessary CWND reductions and possibly unnecessary fast retransmissions can thus be avoided. To help illustrate the proposed PRM scheme, the procedure can be summarized as shown in Fig. 6.

5. Performance of PRM over mSCTP VHO

This section presents performance analyses of the proposed PRM scheme which are simulated using the NS-2 network simulator version 2.29 [7]. The SCTP module for our simulation is taken from the University of
Table 2

<table>
<thead>
<tr>
<th>Network parameters</th>
<th>UMTS</th>
<th>WLAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth</td>
<td>384 Kbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Transmission delay</td>
<td>25 m/s</td>
<td>15 m/s</td>
</tr>
<tr>
<td>MN’s speed</td>
<td>25 m/s</td>
<td>25 m/s</td>
</tr>
<tr>
<td>Wireless network coverage</td>
<td>radius 2000 m</td>
<td>radius 100 m</td>
</tr>
<tr>
<td>Fixed links between CN</td>
<td>10 Mbps</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>RSS</td>
<td>−50 dBm</td>
<td>−70 dBm</td>
</tr>
</tbody>
</table>

Delaware [1]. In addition, the following parameters listed in Table 2 are configured in our proposed scheme for the VHO in HWNs. Figure 7 shows the throughput performance for a VHO when a MN moves into a WLAN area and then adds the new IP address from the WLAN where the CN changes the WLAN link to a primary transmission link. This figure compares the throughput of a conventional mSCTP and the proposed PRM scheme with the same operational conditions, and with a packet delivery time interval of between 20 and 40 s. It can be observed that in all scenarios after VHO in the UMTS/WLAN networks, the PRM provides a clear throughput improvement from 21 to 30 s compared to the conventional mSCTP. It indicates that in a WLAN area the PRM scheme has a higher performance throughout in the mSCTP VHO and a larger CWND than the conventional mSCTP. The average performance throughout the proposed PRM scheme is 181.48 Kbps compared to the conventional mSCTP which is 165.54 Kbps. In other words, our proposed model outperforms the conventional mSCTP by about 16%. The proposed PRM scheme achieves the same performance as the conventional mSCTP scheme with a time of 30 s as the PRM is handling all outstanding data chunks at that time.
We also compare the performance of our proposed PRM scheme with the conventional mSCTP in terms of the CWND which is shown in Fig. 8. The abscissa (x-axis) denotes the packet delivery time and the ordinate (y-axis) denotes the CWND. When the MN sends data chunks using the UMTS link at the beginning, the CWND of the UMTS link is increased by nearly 65,000 bytes before VHO occurs. At time of 21 s, the MN performs a handover to the WLAN link where the mSCTP or the proposed PRM scheme association switches its main link to the WLAN link. In this condition, the ongoing conventional mSCTP association is changed to a slow-start phase because the congestion window and slow-start threshold will be set to half in the WLAN link. Thus, this transmission rate degradation is caused by the VHO as a result, while the proposed PRM scheme shows an enhancement in CWND even after VHO because it does not suffer the initial slow-start phase. On the other hand, the proposed PRM scheme skips the slow-start phase by using the hold/forward procedure. As shown in Fig. 8, it can be observed that our proposed model provides a clear enhancement in CWND compared to the conventional mSCTP during VHO between 21 and 30 s.

In general, the proposed PRM scheme can solve the packet reordering problem during VHO. Through the simulation results, we found that our proposed PRM scheme has the following features:

- It introduces about 16 percent enhancement in performance throughput over the conventional mSCTP scheme.
- It handles all outstanding data chunks at time 30 s.
- It skips the slow-start phase at time 21 to 30 s.

6. Conclusions

During an mSCTP VHO in HWNs, the endpoints of the mSCTP will change from the old link of using UMTS to a new link using a WLAN. Due to the disparity between UMTS/WLAN bandwidth, the packet reordering problem
will occur when the MN of an mSCTP moves to new network. To overcome this packet reordering problem a new model of PRM in the MN is proposed in this paper. The proposed PRM scheme will hold all the outstanding data chunks and then forward it to the MN if it becomes free during an RTT interval. The CWND is increased to more than slow start/avoidance congestion mode by the number of the outstanding data chunks. Simulations results show that the proposed PRM scheme gives the following advantages: increases the throughput by about 16% to outperform the conventional mSCTP scheme, reduces the VHO delay and achieves a seamless VHO. It also shows that the CWND is enhanced by several seconds because it skips the slow-start phase after VHO.

References


An optimized energy saving mechanism in IEEE 802.16e Mobile WiMAX systems

Alaa M. Baker\textsuperscript{a,\*}, Chee Kyun Ng\textsuperscript{a}, Nor Kamariah Noordin\textsuperscript{a}, Ahmed Mustafa\textsuperscript{b} and Ayyoub Akbari\textsuperscript{a}

\textsuperscript{a}Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Selangor, Malaysia
E-mails: alaa_samaka@yahoo.com, {mpnck, nknordin}@eng.upm.edu.my, ayyoub@ieee.org

\textsuperscript{b}School of Computer Technology, Sunway University College, Selangor, Malaysia
E-mail: ahmedm@sunway.edu.my

Abstract. The IEEE 802.16e standard defines a sleep mode operation for conserving power to support the battery life of mobile broadband wireless access (BWA) devices. The system saves energy when it goes through a sleep period with some delay in packet arrival response time. The relationship between energy consumption and the delay is studied to ensure best performance for mobile devices. This relationship has been analyzed by using a mathematical model. A new scheduling method is proposed to adjust the sleep cycle periods by adding a small increase to the next sleep cycle compared with the previous cycle instead of just simply doubling the previous cycle. The simulated results were obtained after adjusting the length of the first sleep cycle period ($T_{\text{min}}$). Adjusting $T_{\text{min}}$ provides a result of 54\% reduction in the time needed for every frame to get a response especially in a lower traffic region. In a high traffic region, a reduction of 21.5\% has been obtained in energy consumption for each sleep mode operation. Therefore, the proposed idea confirms a faster frame response time at lower energy consumption.

Keywords: IEEE 802.16e, WiMAX, sleep mode, energy consumption, frame response time

1. Introduction

The growing demand for internet and wireless multimedia applications has motivated the development of broadband wireless access (BWA) technologies in recent years [1]. In June 2004, the Institute of Electrical and Electronic Engineers (IEEE) standard of 802.16 (IEEE 802.16), or more generally known as Worldwide Interoperability for Microwave Access (WiMAX) has changed the principle of telecommunications as it eliminates the need for resources that service providers had suffered in the last century [3].

WiMAX is an emerging wireless communication system that is expected to provide high data rate communications in metropolitan area networks (MAN) [11]. Since the beginning of the telephone system, service providers have to establish a very high capital investment to deploy a telephone network. The incredibly high cost of deploying copper wire, building switches and connecting those switches made it very difficult to establish a high level system like this.

IEEE released the first WiMAX standard in 2002 which was named as IEEE 802.16 [3], and later on, many modifications and enhancements were added to this standard, such as IEEE 802.16a and IEEE 802.16d. Starting from 802.16 till 802.16d, these standards did not support mobile nodes; they only supported fixed node subscribers stations. The demand grew to include mobile nodes in the WiMAX system, and in 2006, the IEEE released the standard 802.16e [4], which supports mobile nodes and includes all aspects of mobile node parameters, such as handover and power saving mechanisms. With the release of IEEE 802.16e, the nodes are no more permanently

\*Corresponding author: Alaa M. Baker, Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia. E-mail: alaa_samaka@yahoo.com.
connected to a power source, and in this case, the Subscriber Station (SS) battery life has to be taken into consideration. The matter of energy consumption must be reduced to a minimum to sustain the longevity of the battery. To achieve this goal, a sleep mode scheduler mechanism is necessary in order to set the SS into repetitive sleep cycles to save energy.

In this paper, we focus on the sleep cycle time with regard to the packet arriving ratio, and introduce a new scheduling mechanism with two related parameters. The first parameter is the energy consumption within the SS. The second parameter is the frame response time, calculated as the delay between each frame arriving at the SS until it starts to get response. We introduce a new mechanism via mathematical formulas. We obtain the new mechanism performance results and compare them with the previous work. The results show that the new scheduling mechanism is faster in respect of the frame response time, with a decrease in power consumption.

The rest of the paper is organized as follows: Section 2 explains the sleep mode concept in detail, while Section 3 reviews the related works about the sleep mode mechanism. Section 4 describes the mathematical model. The new method is explained in Section 5 including the new proposed equation and all the important aspects. Section 6 discusses the performance results of the new method, and this paper is concluded in Section 7.

2. Sleep mode operation in IEEE 802.16e Mobile WiMAX

An 802.16e SS can be in one of the following two operational modes, wakeup mode and sleep mode. The wakeup mode will send and receive data according to the BS’s scheduling. The sleep mode involves two operational windows known as the sleep window and the listening window. Each sleep window represents a period of time where the SS has no obligation to listen to downlink traffic, and may power down one or more physical operational components in order to reduce power consumption. During the sleep window, the SS cannot receive downlink protocol data units (PDUs), or put simply, packets [5].

The sleep mode operation technique in IEEE 802.16e is shown in Fig. 1, where A is an active state in wakeup mode during which the SS will communicate with the BS by sending and receiving data packets. The I is an idle state in wakeup mode, where there is no data traffic between the BS and the SS. When the SS periodically receives idle states, it will be ready to start the sleep mode. Initially, it sends a mobile sleep request (MOB-SLP-REQ) message to the BS asking for permission to start the sleep mode and waits to get the mobile sleep response (MOB-SLP-RES) message from the BS to allow the SS to start the sleep mode. Thus, \( T \) is the time of the sleep windows, where \( T_0 \) is the time for the first sleep window and is equal to the \( T_{\min} \). \( T_1 \) is the time for the second sleep window and so on. The listening window, \( L \) follows each sleep window to check for any traffic or any data packets that have been queued at the BS. The MOB-SLP-RES message includes the duration of the minimum sleep window, \( T_{\min} \), the maximum sleep window, \( T_{\max} \) and the listening window \( L \) [15].

![Fig. 1. The traditional sleep mode and wakeup mode in IEEE 802.16e.](image-url)
Each node will go through many sleep cycles till it is waken up by an indicator of an arriving packet. Starting from the first sleep cycle after $T_{\text{min}}$, sleep cycles will increase the sleep period by doubling the sleep time of the previous sleep cycle period, till it reaches the maximum period of time, $T_{\text{max}}$. The values of $T_{\text{min}}$ and $T_{\text{max}}$ are already specified and fixed by the BS. $T_{\text{min}}$ is the main factor to manage the sleep time for each sleep cycle; an increase in $T_{\text{min}}$ will lead to the sleep cycle being for a longer period while reducing $T_{\text{min}}$ will lead to reduced sleep periods within each sleep cycle. The first sleep cycle period $T_{\text{min}}$ indicates the behaviour of the system, while the next sleep cycle period will be based on $T_{\text{min}}$. A longer $T_{\text{min}}$ will lead to longer sleep cycles that will save some energy consumption but will increase the frame response time [12].

During a listening window, the SS shall synchronize with the serving BS downlink and listen for the mobile traffic indication (MOB-TRF-IND) message to decide whether to change its status to wakeup mode or go back to the sleep mode [5]. Its value is fixed and it tends to be as short as possible to perform its function. A longer listening window will increase the power consumption because each listening window is considered as an active window. The same reason applies when the number of listening windows is increased in the system. There are two types of MOB-TRF-IND message, positive and negative MOB-TRF-INDs according to the format (FMT) field which is one bit in size. If the MOB-TRF-IND is a negative message which means there is 0 in the FMT field, it indicates that the SS can go for another sleep cycle as there are no data packets queued in the BS buffer, and there is no traffic specified for this unit. The next sleep cycle should be doubled than the previous one as mentioned in the IEEE 802.16e standard [4].

If the MOB-TRF-IND is a positive message with its FMT field set to 1, the sleep cycle will be terminated and it starts the active mode. The system is able to maintain the flow of data traffic in the system again, as it considers the SS to be at the end of the sleep mode. The same procedure will take place after receiving a few idle frames in the wakeup mode when no data packets are sent or received by the SS. The SS will be in an active state when it can send and receive data packets as shown in Fig. 1.

3. Related work

The sleep mode method is an important element to reduce the frame response time and due to this the overall delay will be reduced. Researchers who have worked in this area found a trade-off between frame response time and power consumption, reducing one will increase the other. Due to that, many researchers have conducted research into this matter and have tried to solve it from different aspects to find the optimum value for both. In [10] a general survey of energy-efficient network protocols for wireless networks is provided.

In [6], the authors’ main study was investigating the queuing behaviour of the sleep mode operation in IEEE 802.16e for conserving the power of an SS in terms of the dropping probability and the mean waiting times of packets in the queue of the BS. The results of this method will not be practically accurate because dealing with the queue at the BS leads them to ignore the power consumption parameter at the SS. It is assumed constant regardless whether SS being in sleep mode or in wakeup mode.

In [9], a study of queuing models with multiple vacations to analyze the power consumption and the delay which consists of the queuing delay and serving time. They proposed a theoretical Phase-Type (PH) based Markov chain model. In particular, the service process is designed as a discrete PH model; they derived the closed form expression of the mean delay time of packet arrival and power consumption. They proposed a simple utility function to quantify the efficiency of the sleep mode operation. This function allows the SS to decide when to enable sleep mode operation for power saving. Although the queuing models can properly cover all details of the sleep mode operation, they bring costly computation complexity.

In [2] the authors investigated a method to predict the packet arrival rate, which makes use of semi-Markov chain, but they did not obtain any enhancement to the system. Other researchers in [15] also tried the Markov chain model concept to analyze the traffic arrival rates and also proposed a new algorithm to optimize the system. They proposed to increase $T_{\text{min}}$ exponentially till it reaches the $T_{\text{max}}$. The next sleeping cycle after $T_{\text{max}}$ will not be equal to $T_{\text{max}}$, but it will be a customized period depending on the packet arrival rate during the start of the
sleep mode. The results were not satisfying for the environment where it was found that the delay increase for a high traffic rate range. The consumed energy was almost the same as the traditional method especially at middle and high traffic rate ranges. The researchers in [16] also used the Markov chain model to analyze the packet arrival rate, managed to tune the $T_{\text{min}}$ dynamically to get a proper trade-off between the power consumption and the mean delay according to the traffic load. However, despite the improve power consumption they faced a small increase in the mean delay in the system.

Due to the importance of avoiding increasing power consumption in the SS without adding some delay into the system, the researchers in [8] suggested a new power saving mechanism called the Probabilistic Sleep Interval Decision (PSID) that included the packet response delay. The PSID algorithm determined each sleeping interval by considering the fixed and variable delays. The idea behind the PSID algorithm was to make the SS wake up when the probability that a response packet will arrive at the BS is high. The delay that they assumed was based on past research that was based on a wired environment, because neither technical nor theoretical ways can assume a specific delay in a wireless environment.

In a subsequent paper [14], a distribution method called hyper-Erlang distribution was introduced. The obtained results show no significant improvement. In [13], the authors adapted the Poisson distribution method to analyze the packet arrival rate. The analytical sleep mode model, using both Uplink (UL) and Downlink (DL) data, was investigated by using the Poisson distribution method. Since then, many studies have followed [13] to use the Poisson distribution. In [5] the authors proposed different values for sleep mode parameters and investigated the results obtained. They found out that the best results may be obtained only after setting the initial sleep window size and the final sleep window size as queue empty time.

The applications of the Poisson distribution lead other researchers to come up with a new algorithm. The authors in [7] proposed that the next sleep cycle after $T_{\text{max}}$ will be equal to the average interval between $T_{\text{min}}$ and $T_{\text{max}}$. The period of the sleep cycle will be increased exponentially till it reaches the value of $T_{\text{max}}$ again and subsequently keeps repeating the same cycle. The results were shown as positive but with only a low percentage of improvement.

In general, most of the abovementioned researchers did not provide the common parameters for the sleep mode operation. Some of them did not give the optimized value between those parameters that trade-off between frame response time and energy consumption factors. In the next section, a new sleeping cycles mechanism will be introduced to enhance the trade-off between frame response time and energy consumption factors.

### 4. Analytical model

In this paper, $L$ will not be taken into consideration, due to its very small value. Its value is fixed within all the sleep cycles. It will not contribute a large effect in the equations, although it is part of the physical wakeup in the practical system. The traffic for data packet arrival and its distribution are not predictable. This leads to the use of the probability algorithm to measure the packet arrival rate at the SS. According to past researchers, the best probability algorithm that is compatible with wireless communication traffic is the Poisson distribution algorithm. This research also uses the Poisson distribution to predict the data packet arrival. The equations from (3) to (10) use the Poisson distribution with a rate $\lambda$.

We assume that the arrival frame rate at the SS follows a Poisson distribution with a rate $\lambda$ (frames per unit time). Then the inter-frame arrival time follows an exponential distribution with mean $1/\lambda$ (unit time). We define the sleep time, $D$ as the time period in the sleep mode including one or more sleep cycles. Each sleep cycle includes one sleep window and one listening window. Figure 1 shows the system that consists of the $j$th sleep cycle as the length of each $j$th cycle is $T_j + L$. Let $n$ denotes the number of sleep cycles before the SS gets to the wakeup mode. Let $E[\cdot]$ denotes the mean/average function. Let $e_j$ denotes the event that there is at least one frame arrival during $j$th sleep cycle that could be indicated at the listening window.

Assuming that each sleep cycle is double the previous sleep cycle, the time will be stated for each $j$th cycle as

$$T_j = \begin{cases} 2^{j-1}T_{\text{min}} & \text{if } 2^{j-1}T_{\text{min}} < T_{\text{max}}, \\ T_{\text{max}} & \text{if } 2^{j-1}T_{\text{min}} \geq T_{\text{max}}. \end{cases} \quad (1)$$
By taking the arrival frame rate as a Poisson distribution with a rate $\lambda$, the probability equations, $\Pr(\cdot)$ can be derived as follows

$$\sum \Pr(e_j = True) \Pr(e_j = False) = 1,$$

$$\Pr(e_j = True) = 1 - \Pr(e_j = False),$$

(2)

where

$$\Pr(e_j = False) = e^{-\lambda(T_j+L)}$$

(3)

is the Poisson’s distribution of the sleep mode parameters for no packet arriving at the SS. Substitute (3) into (2), the probability equations, $\Pr(\cdot)$ will become

$$\Pr(e_j = True) = 1 - e^{-\lambda(T_j+L)}.$$  

(4)

The probability of getting at least one downloaded packet at the SS in the first sleep cycle is determined as

$$\Pr(n = 1) = \Pr(e_1 = True) = 1 - e^{-\lambda(T_1+L)}.$$  

(5)

Therefore, the probability of getting at least one downloaded packet at the SS in the second sleep cycle and above can be derived as

$$\Pr(n = j) = \Pr(e_1 = False, e_2 = False, e_3 = False, \ldots, e_{j-1} = False, e_j = True)$$

$$= \prod_{i=1}^{j-1} \Pr(e_i = False) \Pr(e_j = True)$$

$$= \prod_{i=1}^{j-1} (e^{-\lambda(T_i+L)})(1 - e^{-\lambda(T_j+L)})$$

$$= e^{-\lambda \sum_{i=1}^{j-1}(T_i+L)}(1 - e^{-\lambda(T_j+L)}).$$  

(6)

The mean/average number of sleep cycles within one sleep mode operation, $E[n]$ can be obtained from adding the probability of arriving packets for all the sleep cycles as

$$E[n] = \sum_{j=1}^{\infty} j \Pr(n = j)$$

$$= \sum_{j=1}^{\infty} j (e^{-\lambda \sum_{i=1}^{j-1}(T_i+L)}(1 - e^{-\lambda(T_j+L)}))$$

$$= \sum_{j=1}^{\infty} j e^{-\lambda \sum_{i=1}^{j-1}(T_i+L)} + \sum_{j=1}^{\infty} j e^{-\lambda \sum_{i=1}^{j}(T_i+L)}.$$  

(7)

The mean/average time of sleeping cycles in one sleep mode operation, $E[D]$ can be derived by adding the probability of arriving packets for all the sleep cycles while multiplying the time needed for every sleep cycle by the
probability of that sleep cycle, as follow

\[ E[D] = \sum_{j=1}^{\infty} \Pr(n = j) \sum_{k=1}^{j} (T_k + L) \]

\[ = \sum_{j=1}^{\infty} \left( e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L) \right) \left( 1 - e^{-\lambda(T_j + L)} \right) \sum_{j=1}^{j} (T_k + L) \]

\[ = \sum_{j=1}^{\infty} e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L) \sum_{k=1}^{j} (T_k + L) - \sum_{j=1}^{\infty} e^{-\lambda} \sum_{i=1}^{j} (T_i + L) \sum_{k=1}^{j} (T_k + L). \] (8)

The energy consumption, \( E_C \) equation can be determined by adding the \( E_S \) and \( E_L \) to each sleep cycle time, where \( E_S \) and \( E_L \) denote the energy consumption units per unit of time in the sleep cycle and the listening cycle, respectively. Currently there is no information available on the values of \( E_S \) and \( E_L \). All prior research conducted on sleeping mode algorithm use the power consumption values from 802.11 WLAN devices. This research assumes the consumption in the wake up mode, \( E_L \) as 500 mW and the consumption in the sleep mode, \( E_S \) as 50 mW, so the ratio would be 10 to 1. Depending on the actual power consumption values, the exact performance will vary, but the general trends should remain the same. Hence, \( E_C \) can be derived as

\[ E_C = \sum_{j=1}^{\infty} \Pr(n = j) \sum_{k=1}^{j} (T_k E_S + L E_L) \]

\[ = \sum_{j=1}^{\infty} \left( e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L) \right) \left( 1 - e^{-\lambda(T_j + L)} \right) \sum_{j=1}^{j} (T_k E_S + L E_L) \]

\[ = \sum_{j=1}^{\infty} e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L) \sum_{k=1}^{j} (T_k E_S + L E_L) - \sum_{j=1}^{\infty} e^{-\lambda} \sum_{i=1}^{j} (T_i + L) \sum_{k=1}^{j} (T_k E_S + L E_L). \] (9)

Since the frame arrival follows a Poisson distribution, the arrival events are random observers to the sleep intervals. Therefore, the frame response time, \( E[R] \) can be derived as

\[ E[R] = \sum_{j=1}^{\infty} \frac{1}{2} \Pr(n = j)(T_j + L) \]

\[ = \sum_{j=1}^{\infty} \frac{1}{2} \left( e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L) \left( 1 - e^{-\lambda(T_j + L)} \right) \right)(T_j + L) \]

\[ = \sum_{j=1}^{\infty} \left( \frac{1}{2} e^{-\lambda} \sum_{i=1}^{j-1} (T_i + L)(T_j + L) - \frac{1}{2} e^{-\lambda} \sum_{i=1}^{j} (T_i + L)(T_j + L) \right). \] (10)

5. Optimized sleep mode management scheme for energy consumption and delay

The proposed optimized sleep mode management scheme concentrates on the number of sleep cycles during one sleep mode operation until the start of the wake up mode. The proposed new scheme focuses on both parameters
evenly, the frame response time and the power consumption. By increasing the number of sleep cycles per each sleep mode operation with a shorter period, the frame response time of the system can be decreased by a significant value. The energy consumption of the system will decrease as well. As $T_{\text{min}}$ and $T_{\text{max}}$ are fixed by the BS in the MOB-SLP-RES so that it will increase the duration of the sleep cycles between $T_{\text{min}}$ and $T_{\text{max}}$ until it reaches $T_{\text{max}}$. After that, the sleep cycle will be fixed at $T_{\text{max}}$ for that sleep mode operation.

Equation (1) showed that the next sleep cycle is double the previous one. Nevertheless, the new proposed scheme can increase the number of sleep cycles within the system with a shorter period as follows

$$
T_j = \begin{cases} 
2(j-1)T_{\text{min}} & \text{if } 2(j-1)T_{\text{min}} < T_{\text{max}}, \\
T_{\text{max}} & \text{if } 2(j-1)T_{\text{min}} \geq T_{\text{max}}.
\end{cases}
$$

The next sleep cycle will only increase a little bit from the previous cycle. The frame response time will be reduced due to shorter sleep cycles as shown in Fig. 2.

As the $T_{\text{min}}$ period value is fixed by the BS, the optimized algorithm adjusted the value of subsequent sleeping cycles after $T_{\text{min}}$ till reaching $T_{\text{max}}$. Sleeping cycles after $T_{\text{min}}$ have a shorter time as part of the optimized algorithm relative to the traditional algorithm. Each arriving frame will suffer a shorter time delay to get a response in the optimized algorithm compared to the traditional algorithm. However, the power consumption will increase because of the increase in the number of sleep cycles and each sleep cycle followed by a listening cycle, but the increase is negligibly small. The listening window is typically a wake up state that takes a small amount of energy every time it wakes up. Figure 2 shows that the data packets arriving at $T_n$ will get a response in a shorter time compared to Fig. 1, due to reducing the length of the sleep cycle period.

Suppose the increase in sleep cycles such as the sequence of 1, 2, 3, 4, 5, ..., is applied, the sleep period for each sleep cycle will be very short. This will have a beneficial impact on the frame arrival rate. However, this sequence indeed will consume so much more energy because every sleep window has to have one listening window. This listening window is considered as an active window and hence the energy consumed in the sequence of 1, 2, 3, 4, 5, ..., will put the concept of the sleep mode mechanism at the risk of not being an effective mechanism.

The results of the performance are obtained by determining the ratio between two parallel points on the graph. Hence, the ratio is between randomly selected pick up points from the optimized algorithm with the equivalent points from the traditional algorithm. The final ratio is the average of the previously calculated points.

6. Performance evaluation results

The following simulation metrics: $E[n]$, $E[D]$, $E[C]$ and $E[R]$ are considered in the performance evaluation. These parameter metrics are represented by (7)–(10), respectively. They are simulated mathematically using visual C++ under a packet arrival rate of between 0 and 0.2 packet frames per unit time. These simulations are obtained over Best Effort (BE) and Non-Real Time Variable Rate (NRT-VR) environments with a Variable BitRate (VBR).
data type. The simulations are completed with $T_{\text{min}} = 1, 4, 16$ ms over $\lambda$. These simulated results are using the following parameters; $L = 1$ ms, $T_{\text{max}} = 1024$, $E_S = 1$ mW and $E_L = 10$ mW.

The traditional algorithm results for $E[n]$, $E[D]$, $E_C$ and $E[R]$ are collected after substituting (1) into (7–10), respectively, while the optimized algorithm results are generated by substituting (11) into these same equations. We first conducted simulations for $E[n]$ from (7) which represents the mean/average number of sleep cycles within one sleep mode operation. Figure 3 shows both the curves of the traditional and optimized algorithm results for $E[n]$ at $T_{\text{min}} = 1$. It shows that the average number of sleep cycles for the proposed algorithm is more than the traditional algorithm. Figures 4 and 5 are the curves of the $E[n]$ results for $T_{\text{min}} = 4$ and 16, respectively. These figures also show the same improvement of the proposed algorithm over the traditional algorithm.

Figures 6–8 show the traditional and the optimized curves for $E[D]$ which represents the average duration time for all sleeping cycles within one sleep mode operation with $T_{\text{min}} = 1, 4, 16$, respectively. These figures show that the average sleep time for all sleep cycles is lower for the optimized algorithm comparing to the traditional algorithm. We can conclude that the energy consumption for the proposed algorithm is higher compared to the traditional algorithm.

Equation (9) evaluates the energy consumption for the optimized algorithm and the traditional algorithm. Using $T_{\text{min}} = 1$, the curves in Fig. 9 show the optimized algorithm uses more energy than the traditional algorithm by an insignificant amount. The average performance shows only 1.7% more consumption of energy power for the proposed algorithm compared to the convention algorithm. Figure 10 shows how the energy consumption decreases by an average of 2.2% for the proposed algorithm compared to the traditional algorithm for $T_{\text{min}} = 4$. It is observed that increasing $T_{\text{min}}$ will reduce the power consumption. Figure 11 shows a drop in power energy consumption with an average ratio of 21.5% compared to the traditional algorithm for $T_{\text{min}} = 16$. It shows about 19.3% improvement in the power consumption compared to $T_{\text{min}} = 4$.

For a low data packet arrival rate the system will go through many sleep cycles till it is woken up by a positive indicator. Furthermore, sleep cycles are considered to have a very low energy consumption compared to active
Fig. 4. The traditional and optimized values of $E[n]$ for $T_{\text{min}} = 4$.

Fig. 5. The traditional and optimized values of $E[n]$ for $T_{\text{min}} = 16$. 
Fig. 6. The traditional and optimized values of $E[D]$ for $T_{\text{min}} = 1$.

Fig. 7. The traditional and optimized values of $E[D]$ for $T_{\text{min}} = 4$. 
Fig. 8. The traditional and optimized values of $E[D]$ for $T_{\text{min}} = 16$.

Fig. 9. The traditional and optimized values of $E_C$ for $T_{\text{min}} = 1$. 
Fig. 10. The traditional and optimized values of $E_C$ for $T_{\text{min}} = 4$.

Fig. 11. The traditional and optimized values of $E_C$ for $T_{\text{min}} = 16$. 
cycles. For this reason the energy consumption is very low with a low data packet arrival rate in both the optimized and the traditional algorithms. Therefore, the energy cost is considered small.

The optimized algorithm tends to increase the number of sleep cycles within the sleep mode operation and shortens each sleep cycle period. The time needed for each packet frame to get a response is shorter, so the optimized algorithm can manage to reduce the delay in frame response. As an example, by using the traditional algorithm as referred to in (1), the seventh sleep cycle period is 128 time frames at $T_{\text{min}} = 1$ while the same cycle is only 14 time frames by using the optimized algorithm which is referred to in (11). The tenth sleep cycle period is 1024 time frames for the traditional algorithm, while the same cycle is only 20 time frames for the optimized algorithm. It is obvious that the packet frame will get a response sooner by using the optimized algorithm. The generated curves of $E[R]$ which represent the frame response time for the optimized and traditional algorithms are shown in Figs 12–14 for $T_{\text{min}} = 1, 4$ and 16, respectively. Figure 12 indicates that the optimized algorithm performs with a very low frame response time compared to the traditional algorithm with an average of 54%. Unlike the energy power consumption performance, the frame response time has a negative trend towards increasing $T_{\text{min}}$ as shown in Fig. 13 for $T_{\text{min}} = 4$, which indicates that the optimized algorithm shows a performance ratio of a 5.5% reduction in the frame response time. Furthermore, Fig. 14 shows an increase in the frame response time with a ratio of 2% compared to the traditional algorithm when $T_{\text{min}} = 16$ is used.

By increasing $T_{\text{min}}$, the mean/average number of sleep cycles will be reduced for the same $T_{\text{max}}$ due to the increase in the sleep cycle period. Applying $T_{\text{min}} = 1$ is much better in terms of a shorter frame response time compared to $T_{\text{min}} = 4$ and $T_{\text{min}} = 16$. Moreover, increasing the period of each sleep cycle tends to decrease the energy consumption as shown in Fig. 11, especially at a high data packet arrival rate ratio. Also, the frame response time will be decreased by increasing the energy consumption. For future work, the optimum value of $T_{\text{min}}$ should be obtained for an acceptable delay and energy consumption.

![Figure 12. The traditional and optimized values of $E[R]$ for $T_{\text{min}} = 1$.](image)
Fig. 13. The traditional and optimized values of $E[R]$ for $T_{\text{min}} = 4$.

Fig. 14. The traditional and optimized values of $E[R]$ for $T_{\text{min}} = 16$. 
7. Conclusion

In this paper, a new proposed algorithm for optimizing the energy saving mechanism in IEEE 802.16e WiMAX has been introduced. The sleep mode operational concept in WiMAX IEEE 802.16e has been discussed. The optimized results increase the mean/average number of sleep cycles per one sleep mode operation while decreasing the mean/average time for all sleep cycles within one sleep mode operation. The frame response time is 54% less than the traditional algorithm especially in a lower traffic region, while 21.5% of the energy has been reduced compared to the traditional algorithm especially in a higher traffic region. Therefore, our proposed system reacts faster to arriving packets with an insignificant increase in power consumption. On the power consumption side, the power used is reduced by increasing $T_{\text{min}}$ compared to the traditional algorithm. The frame response time is decreased by the increase of $T_{\text{min}}$. Overall our proposed algorithm would appear to be superior to the conventional algorithm because of the significant decrease in the packet arrival delay and also a decrease in power consumption for each sleep mode mechanism.

References