A Novel Encoding Scheme for Improving the Bandwidth Efficiency of DPPM

Md Shafiqul Islam, Mohamed Younis, Muntasir Mahmud, Jaeed Bin Saif
Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, Maryland, USA
mdislam1, younis, mmahmud1, jaeedb1@umbc.edu

Abstract*—The differential pulse position modulation (DPPM) is one of the popular power-efficient schemes for supporting visible light communication in underwater environments and across the air-water interface. Despite such an advantage, DPPM does not efficiently utilize the available channel capacity. This paper aspires to tackle such shortcomings by striking a better balance between power and bandwidth efficiency. Particularly, L-DPPM is considered where a block of M input data bits is mapped into one of the L distinct waveforms containing only one 'on' chip. A novel encoding algorithm and frame structure are proposed in order to shorten the time between consecutive symbols and consequently improve the bit rate of L-DPPM. The idea is based on avoiding bit patterns that contribute the most to bandwidth inefficiency. The proposed algorithm explores a number of bit patterns remapping through simple complement and shifting operations. A detailed frame structure with all necessary control bits is provided. Overall, boosting the bandwidth efficacy comes at the expense of a slight increase in control bit count and transmission power. The simulation results demonstrate the effectiveness of the proposed encoding algorithm and provide guidelines for determining M for best performance.

Keywords: Differential pulse position modulation; Visible light communication; Underwater optical networks; Free space optics.

I. INTRODUCTION

Wireless optical communication is very popular because of its potential for higher bandwidth efficiency. Traditionally it has been very effective for indoor communication. Recently, it has become more popular for outdoor communication which is known as free-space optical communication (FSO) or visible light communication (VLC). Among outdoor communication, underwater communication is gaining increased attention from the research community due to its vast applications such as oceanic studies, search and rescue, sea floor observation, and security surveillance, etc. Usually, an acoustic signal is used for underwater communication due to its capability of long-distance communication. However, the achievable bit rate using acoustic signal is very low compared to the optical signals. Therefore, VLC is a good alternate for underwater communication, where a higher bit rate is required. Another effective application of VLC is for enabling communication through the air-water interface. Neither RF nor acoustic signals can be used for such cross-medium communication because the air-water interface acts as a high impedance for these signals. In [1] we have shown that we can achieve better coverage area and signal strength to communicate through the air-water interface using visible light. Nonetheless, VLC still lacks signal strength for this kind of application, especially for long-distance communication. Therefore, a power efficient modulation scheme is required for such an application.

Numerous modulation techniques have been developed for optical communication over the years. For simplicity, underwater optical communication usually uses intensity-based modulation with direct detection technique (IM / DD). The most common modulation technique is on-off keying (OOK) with NRZ or RZ encoding [2][3][4]. Although the bandwidth efficiency and bit rate are very high using OOK-NRZ or OOK-RZ, power efficiency is not good. Since the optical signal needs to penetrate the air-water interface and to propagate in underwater environments, power efficiency is very crucial for optical communication due to the absorption and scattering loss of the optical signal. Among the various options, the pulse position modulation (PPM) is one of the most popular power efficient techniques for optical communication [5][6]. In PPM, each M bits are sent over a symbol $L = 2^M$ time chips and only one pulse is sent in L for the chip position, corresponding to the value of the M bits.

PPM requires very accurate clock synchronization between the transmitter and receiver, which is quite challenging in underwater environments. Moreover, PPM is not bandwidth efficient. In order to achieve better bandwidth efficiency, a number of modified versions of PPM have been proposed, such as overlapping PPM (OPPM) [7][8], multiple PPM (MPPM), differential PPM (DPPM) [9][10], pulse-interval modulation (DPIM) [11][12], and dual-header pulse-interval modulation (DH–PIMα) [13]. Among these PPM variants, DPPM is the most popular. DPPM starts the next symbol after sending the pulse, i.e., before the elapse of the remaining time chips of the symbol L. Thus, in DPPM the transmitter and receiver do not need to have tightly synchronized clocks.

Only a few studies could be found in the literature for improving the performance of DPPM. In [14], it has been shown that the energy efficiency of DPPM can be increased as high as 45.2 % in comparison to OOK by choosing optimal data word length. Soft decision decoding is not possible for DPPM because of its variable symbol size. A modified version of DPPM, called IDDPM, is derived from DPPM by adding an extra zero before the DPPM symbol [15]. This modification helps the receiver for soft-decision decoding. Although DPPM has a higher bandwidth efficiency than PPM, its bandwidth efficiency is still significantly lower than OOK. We found very

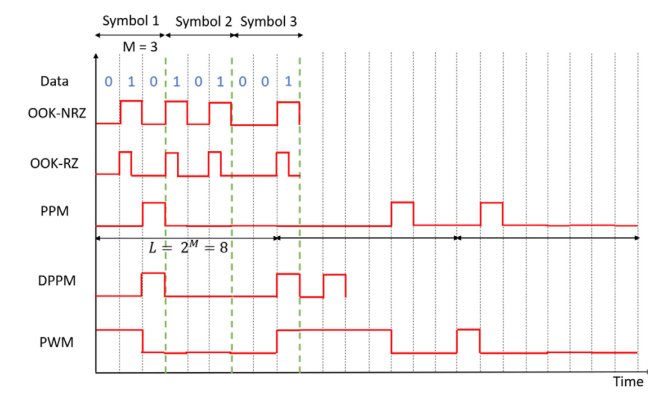

Fig. 1. Different modulation techniques for VLC communication

few published studies on boosting the bandwidth efficiency of DPPM. In [16], DPPM is combined with Pulse-width modulation (PWM) to improve bandwidth efficiency. However, the realization of such a scheme requires reducing the pulse duration which increases the packet error rate significantly. In our previous, work we have shown how we can improve power efficiency by varying modulation index, M of DPPM [17]. In this paper, our objective is to increase the bandwidth efficiency and assess the corresponding inverse effect on power efficiency. We propose a novel algorithm and frame structure which helps to shorten the DPPM frame size.

The paper is organized as follows. In section II, theoretical analysis is discussed. Section III presents different methods for bandwidth efficiency. Section IV describes the frame design approach. Section V presents the validation results. The paper is concluded in Section VI.

II. THEORETICAL ANALYSIS

In PPM, information is encoded in the pulse position; the pulse position depends on the value represented by the corresponding M input data. In L-PPM, a block of $M = log_2 L$ input data is mapped into one of the L distinct waveforms containing only one 'on' chip and the remaining L-1 chips are off, where $M > 0$. A pulse $p(t)$ is transmitted on that 'on' chip (time slot). Figure 1 explains the PPM with an example along with other modulation techniques. In this example, the actual data is 9 bits long, and M is to set to 3, which means $L = 2^M = 8$. The input data is thus partitioned into groups of 3 bits, with decimal values of 2, 5, and 1, respectively. The pulse positions will be the third, sixth, and second, within the first, second, and third L time slots, respectively.

PPM is power efficient because we are sending fewer 'on' pulses than other modulation techniques like OOK-NRZ, OOK-RZ, and PWM. This is a key advantage for the energy constrained applications like those involving underwater wireless optical communication. However, the bandwidth efficiency of PPM is not as good as OOK because the symbol is longer, and thus more time is needed to transmit the same data than OOK. Another disadvantage of PPM is the need for very tight clock synchronization between the transmitter and receiver since accurate pulse positioning is crucial for successful reception in PPM. These two issues are addressed in DPPM, which is a modified version of PPM. DPPM improves power efficiency as well as bandwidth efficiency by removing the extra zeros after the pulse position. Figure 1 also shows the DPPM waveform, where the extra zeros after the pulse have been omitted from the PPM waveform. Thus, the maximum and minimum DPPM frame size are:

$$F_{max} = 2^M \cdot \left\lceil \frac{D}{M} \right\rceil \text{ and } F_{min} = \left\lceil \frac{D}{M} \right\rceil \quad (1)$$

Where, D is the data size. Hence the average the frame size is:

$$F_{avg} = \frac{D(2^M+1)}{2M} \quad (2)$$

From Eq. (2) we can clearly see, the average frame size is bigger than actual data size, D and it grows rapidly with the increase of M. It is worth mentioning that among all the modulation technique OOK requires a smaller number of bits to modulate the actual data, and the modulated data size exactly matches the data length, D. To capture the relation between actual and modulated data sizes, we define protocol efficiency, η as follows:

$$\eta = \frac{D}{F_{avg}} \times 100\% = \frac{2M}{2^M+1} \times 100\% \quad (3)$$

The above equation indicates that the protocol efficiency for OOK is 100% and less than 100% for DPPM modulation for any value of M. Though the protocol efficiency of DPPM is less than OOK, it is very power efficient because it requires fewer number of 'on' chips than OOK. If D and M are fixed, then the DPPM frame always contains $\left\lceil \frac{D}{M} \right\rceil$ 'on' chips for any data bit pattern. For example, from Figure 1 we can see that the DPPM frame contains 3 'on' chips. If the value of data is changed, then the DPPM frame will again contain 3 'on' chips but in different positions. On the other hand, in OOK, the number of 'on' chips depends on the decimal value of data. If the probabilities of having 'on' and 'off' chips (meaning having 1 or 0 for a bit) in the data are the same, then a message of size D contains an average of $\frac{D}{2}$ 'on' chips. Thus, we can define the power efficiency of DPPM with respect to OOK as follows:

$$P_{dppm/ook} = \left(1 + \frac{\left\lceil \frac{D}{2} \right\rceil - \left\lceil \frac{D}{M} \right\rceil}{\left\lceil \frac{D}{2} \right\rceil}\right) \times 100\% = (1 + \frac{M-2}{M}) \times 100\% \quad (4)$$

For example, if $M = 4$, then, $P_{dppm/ook} = 150\%$, which means we can improve power efficiency by 50% compared to OOK. If M increases, the power efficiency grows; however, from Eq. (3) we know protocol efficiency diminishes for large M, meaning that the bandwidth efficiency decreases as well. In the next section we will discuss some methods by which we can improve the protocol efficiency.

III. IMPROVING BANDWIDTH EFFICIENCY

The DPPM frame size depends on the data length D. Technically from Figure 1, we can observe that the DPPM frame size depends on the decimal value of each group of M bits in the data pattern. Recall that in DPPM, a message is divided into $\frac{D}{M}$ groups. If we can map the message bit pattern to another where the sum of the decimal values of each group is less than that of the original message, then we can reduce the size of the DPPM frame, which eventually improves the bit rate and consequently the protocol efficiency. In order to do such mapping, it will be necessary to add some overhead, i.e., control bits, to the DPPM frame to enable successful message decoding at the receiver. The following discusses our proposed transformation for improving the DPPM bit rate.

1's complement method: For certain messages, we can improve the bandwidth efficiency by simply sending the 1's complement version of input data instead of sending the original one. For example, if input data, $D = '11110111'\ and\ M = 4$, after DPPM modulation it becomes '0000000000000000100000001'. The 1's complement of D is '00001000' and after modulation it becomes '1000000001'. Thus, in this case, clearly sending the original data pattern requires more bits than sending its 1's complement. Since there could be another scenario where sending the original data pattern requires fewer bits, an additional control bit has to be included in the DPPM frame to indicate whether the original or 1's complement version has been used to encode the data.

Shifting method: By circular shifting of message bit patterns, we can also minimize the sum of decimal values of each group. For example, if M = 4 and D = '0001100000000001' the decimal values of each 4 bits are (1, 8, 0, 1). Now if we right shift D three times circularly, D becomes '0010001100000000' which provides a much smaller sum of decimal values (2, 3, 0, 0). To support shifting, we need control flags to specify the shift amount. Since a message of length D can be shifted (D-1) times, the overhead to represent the number of shift amount is,

$$O_s = \lceil Log_2 D \rceil \quad (5)$$

The above equation indicates that a larger value of D, i.e., longer message, would increase the overhead.

Reducing extra 0's from DPPM frame: We have already mentioned that a higher value of M increases the DPPM frame size and that frame mostly contains a long string of 0's. Our next method is to find a way to reduce extra 0's from that DPPM modulated data. The idea is to indicate that only 50% of the zeros are indeed included in the symbol modulated bit pattern and the rest are omitted. For example, for "0000001" we include "0001" and provide a control bit to reflect that only 50% of the zeros are included. Generally, if the DPPM frame contains a count of n zeros before an 'on' chip, we can reduce that number by either $\frac{n}{2}$ or $(\frac{n-1}{2} + 1)$ based on whether n is even or odd. For instance, assume that '00000000010000010000101' is a DPPM modulated message, which contains (9, 5, 4, 1) 0's before every 1's. Now we can reduce those 0's to (5, 3, 2, 1) based on the above methods. A control bit will be set, if zeros are omitted. In order to indicate whether the reduction comes from an odd or even number of 0's, we need to add an additional controls bit. For this example, for the four symbols we could add '1101' where a one indicates that the actual number of 0's was odd and a zero implies an even number of 0's. The number of control bits, in this case, depends on how many groups in the message and can be determined by:

$$O_z = \left\lceil \frac{D}{M} \right\rceil \quad (6)$$

Figure 2 shows how much we can reduce the DPPM frame size by these three methods individually for a message size, D =32 bits for different values of M. From this figure we can see, a small value of M like 2, actually increases rather than decreases the frame size. However, for higher values of M, any method can reduce the frame size than original DPPM frame. In the next section, we develop an algorithm that leverages these three methods to minimize the DPPM frame size.

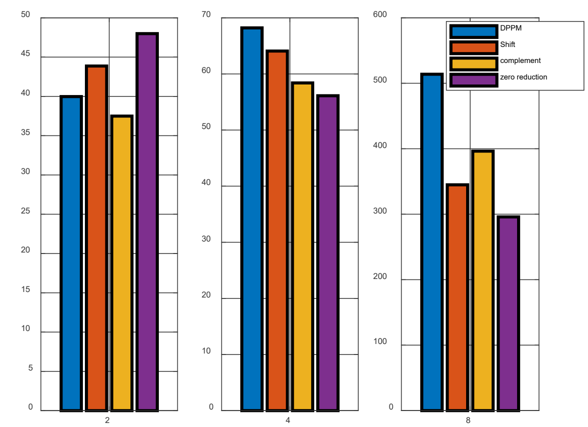

Fig. 2. DPPM frame sizes vs M for different methods (D =32 bits)

1's complement of individual groups: Applying the 1's complement to the whole message doesn't always reduce the DPPM frame size. For example, if message = '00001111' and M = 4, the 1's complement becomes '1111000' which means DPPM data size will stay the same. Alternatively, we explore applying 1's complement to the individual groups and adopt the version which has the lowest equivalent decimal values. In the above example, the first 4 bits of data is '0000', and thus will include it as is. The next 4 bits are '1111', and applying 1's complement is beneficial. Thus, our modified message will be '00000000' and corresponding the DPPM frame will be '1101', where two extra control bits are appended to indicate whether we have used the original or complement bit pattern of each group. Therefore, in this method, the added control bits, O_{cg}, equal to the number of groups.

$$O_{cg} = \left\lceil \frac{D}{M} \right\rceil \quad (7)$$

The frame design under this method is referred to as Complement DPPM frame (CDPPM).

IV. FRAME DESIGN

A. EDPPM frame

As discussed in the previous section, our proposed three methods work well in different message data patterns. In this section, we will combine those methods to create a frame structure which provides the minimum number of bits to represent a DPPM modulated data. We name our approach as Enhanced DPPM (EDPPM). If O_c, O_s, O_z represent the number of control bits required to represent complement, shifting, and reduction of 0's, then the total overhead in bits is:

$$O = O_c + O_s + O_z \quad (8)$$

Figure 3(a) shows the frame structure of EDPPM, where payload is the modulated data. Using Eq. (5) and (6), we can update Eq. (8) as follows:

(a) (b)

Fig. 3. EDPPM (a) and CDPPM (b) frame structures

$$O = 1 + \lceil Log_2 D \rceil + \left\lceil \frac{D}{M} \right\rceil \tag{9}$$

From Eq. (1) we know that the minimum original DPPM data size is $\frac{D}{M}$. For EDPPM we need to add the overhead, hence the minimum EDPPM frame size is:

$$F'_{min} = \left\lceil \frac{D}{M} \right\rceil + O = 1 + 2\left\lceil \frac{D}{M} \right\rceil + \lceil Log_2 D \rceil \tag{10}$$

Finding the maximum frame size is tricky since we are applying all three methods together. We need to find a pattern of the input data for which the modulated version doesn't get reduced by any of the above methods. For example, if M = 4 and D = '00001111', we cannot reduce the DPPM data. If we apply one's complement, the sum of the decimal value of each group doesn't change. Moreover, if we apply any number of right shifts, the sum of the decimal value of each group remains the same. This example tells us that if half of the groups have '0000' pattern and another half have '1111' pattern for M = 4, complement and shifting do not help to reduce the DPPM data size. Thus, we can represent the maximum frame is as follows:

$$F'_{max} = \frac{1}{2}\left\lceil \frac{D}{M} \right\rceil + 2^{M-1}\left\lceil \frac{D}{M} \right\rceil + O$$
$$= 1 + \left\lceil \frac{D}{M} \right\rceil \left(\frac{3}{2} + 2^{M-1}\right) + \lceil Log_2 D \rceil \tag{11}$$

By knowing F'_{min} and F'_{max}, we can calculate the average frame size.

$$F'_{avg} = 1 + \left\lceil \frac{D}{M} \right\rceil \left(\frac{7}{4} + 2^{M-2}\right) + \lceil Log_2 D \rceil \tag{12}$$

Using Eq. (3), we can again calculate the protocol efficiency of EDPPM as follows:

$$\eta' = \frac{D}{1 + \frac{D}{M}\left(\frac{7}{4} + 2^{M-2}\right) + \lceil Log_2 D \rceil} \times 100\% \tag{13}$$

Now, we redefine the power efficiency of our proposed frame structure in a similar way like Eq. (4). In the case of EDPPM, we need to keep in mind that the extra overhead also carries 'on' chips. If the probabilities of having an 'on' and 'off' chip are similar, the overhead will contain on average $\frac{O}{2}$ 'on' chips. Thus, the power efficiency of our proposed method relative to OOK will be

$$P_{edppm/ook} = \frac{\frac{D}{2} - \left(\left\lceil \frac{D}{M} \right\rceil + \frac{O}{2}\right)}{\frac{D}{2}} \times 100\% \tag{14}$$

Substituting the value of O from Eq. (8) into Eq. (14) we get,

$$P_{edppm/ook} = 1 - \frac{3}{M} - \frac{1}{D}\left(1 + \lceil Log_2 D \rceil\right) \tag{15}$$

This new protocol efficiency and power efficiency will be explained elaborately through simulation in Section V.

A pseudo-code summary of the steps for creating an EDPPM frame is shown in Algorithm 1. We will explain such an algorithm using the example shown in Figure 4.

Step 1: The key parameters are to be determined, specifically, the input data, *P*, the data size, *D*, and the modulation index, *M*. In the example, *P* = '1110011111111110', $D = 16$ bits and $M = 4$. Depending on the value of *M* and *P, the* size of O_s and O_z are determined using Eq. (5) and (6), which are 4 for both cases in this example.

Step 2: Apply 1's complement to *P*.

Steps 3-8: Perform circular right shift to both original and complement versions of *P* for *D* times. Each time we calculate the sum of decimal values of each group. We then identify the lowest sum achieved for both original and complemented versions. In this example, we get the lowest sum after shifting 1 time for original input data and 3 times for the complement version and the lowest sums are 40 and 5 for original and 1's complement version respectively. We adopt the smallest value and adjust the bit pattern accordingly. In the example, '0010001100000000' is picked. Thus, instead of sending the original bit pattern for P, this value will be considered for DPPM modulation. Since the 3rd right shift of the 1's complement version has been taken, control bit $O_c = 1$ and $O_s = 0011$.

Step 9: Since, M = 4, after DPPM modulation to '0010001100000000' yields '001000111'.

Input: Message data payload (P), D, M
Output: EDPPM frame

---

1. *initialize: M, D, and size of O_c, O_s and O_z*
2. *get 1's complement of P, P'*
3. ***for** i=1 to D=size of(P)*
4. *circular right shift P & P'*
5. *calculate sum of decimal value of each group*
6. ***end** for*
7. *get minimum sum of decimal values of each group*
8. *update P, O_c and O_s*
9. *modulate P using DPPM*
10. *shortening the string of leading 0's from each group of DPPM frame to either $\frac{n}{2}$ or $(\frac{n-1}{2} + 1)$*
11. *update O_z*
12. *EDPPM frame = DPPM + O_c + O_s + O_z*

Algorithm 1. Steps for generating an EDPPM frame

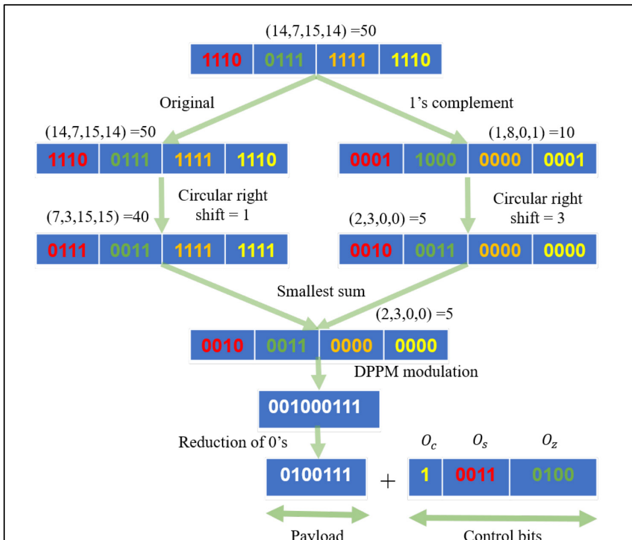

Fig. 4. Illustration of creating EDPPM frame

Steps 10-11: Removing some of the leading 0's from the modulated data, as discussed earlier. In the example, '001000111' becomes '0100111' which is the data payload of the EDPPM frame. Accordingly, O_z is updated to be '0100'.

Step 12: Creating the EDPPM frame by adding payload and overhead, which is '0100111100110100' in the example.

B. *CDPPM frame*

Creating a CDPPM frame is quite simple; Algorithm 2 describes the steps. In such a frame structure, we have only payload and one overhead field which has shown in Figure 3(b). The size of overhead O_{cg} can be calculated using Eq. (7). Figure 5 explains algorithm 2 with an example.

Steps 1-2: Again, the key parameters are *P*, *D*, and, *M*. In the example, *P* = '1011111100000111', $D = 16$ bits, and $M = 4$. Depending on the value of *M* and *P,* size of O_{cg} is determined using Eq. (7), which is 4 in this example.

Steps 3-12: Find the 1's complement of each group of *M* data bits and compare the decimal value to that of the original bit pattern. If the complement version has a smaller decimal value, it will be adopted. For instance, in the example in Figure 5, the decimal value of the first 4 bits of data is 11 and its complemented value is 4; hence, the latter is to be used and the corresponding O_{cg} field will be 1.

Step 13: Combine the new bit pattern for the data with O_{cg} to create the CDPPM frame. In this case, it is '01000000000001111100'

V. Validation Results

Algorithms 1 and 2 have been implemented using MATLAB to analyze the protocol and power efficiency of our proposed EDPPM and CDPPM frame structures. Figure 6 captures the effect of *M* on protocol efficiency for different data sizes of the EDPPM frame. We also show protocol efficiency for original DPPM data. From Eq. (3), it is clear that protocol efficiency doesn't change with the data size, *D,* which is obvious in this figure. For *M* = *2*, the protocol efficiency is actually worse than

Input: Message data payload (P), *D*, *M*
Output: CDPPM frame

---

1. *initialize: M, D and size of* O_{cg}
2. *group, G =* $\left\lceil \frac{D}{M} \right\rceil$
3. ***for*** *i =1 to G*
4. *calculate 1's complement of each group*
5. ***if*** *(1's complement of G(i) < original G(i)*
6. *new G(i) = 1's complement of G(i)*
7. $O_{cg}(i) = 1$
8. ***else***
9. *new G(i) = original G(i)*
10. $O_{cg}(i) = 0$
11. ***end if***
12. ***end for***
13. *CDPPM frame = G +* O_{cg}

Algorithm 2. Steps for generating the CDPPM frame

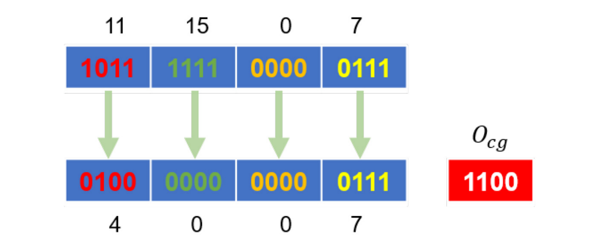

Fig. 5. Illustration of creating CDPPM frame

the original DPPM, which makes sense because in this case, the overhead is very high relative to the reduction of bit count achieved by our method. However, for higher values of *M*, we can see protocol efficiency increasing significantly and surpassing DPPM. For instance, we can increase protocol efficiency by 45% for *M* = *4*. One of the key problems of DPPM is that its protocol efficiency decreases a lot with the increase of *M*. Here, we can see, when *M* = *8*, the protocol efficiency drops to around 5%. Yet, using our method we can improve it a lot, especially for smaller data sizes.

The improved protocol efficiency comes in exchange for a slight reduction in the power efficiency, as shown in Figure 7, which has been plotted using Eq. (4) and Eq. (15). From this figure, we can observe that for a low value of *M*, the reduction of power efficiency becomes major. For example, if *D* = *64*, *M*

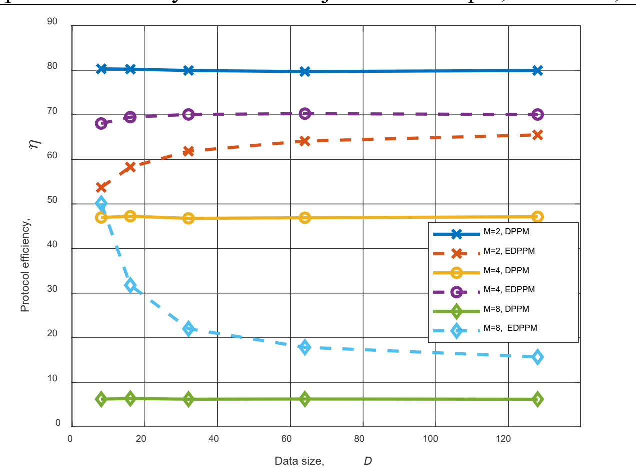

Fig. 6. Protocol efficiency of EDPPM frame for different data size

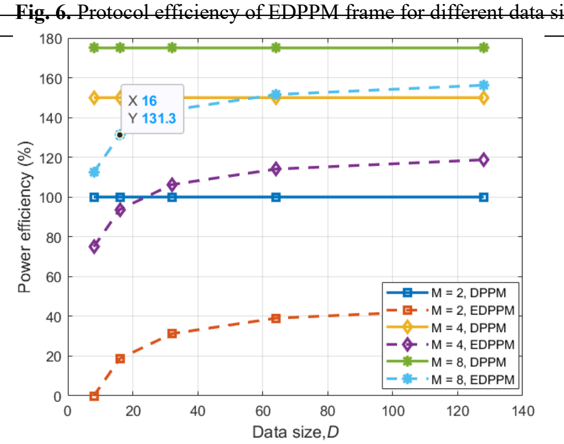

Fig. 7. Power efficiency of EDPPM frame relative to OOK frame for different data size

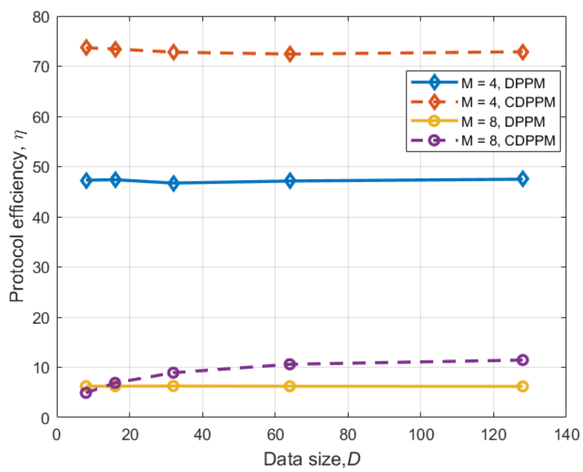

Fig. 8. Protocol efficiency of CDPPM frame for different data size

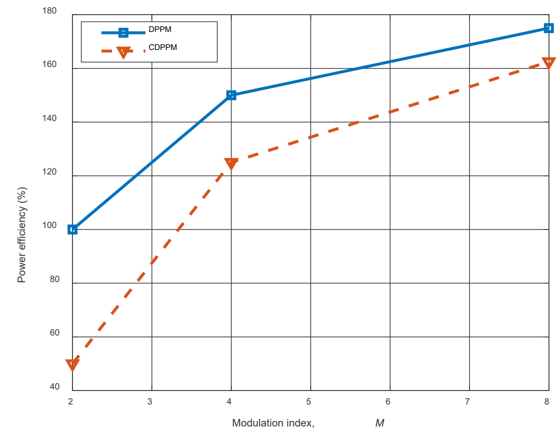

Fig. 9. Power efficiency of CDPPM frame relative to OOK frame for different data size

$=2$, the power efficiency using our method is 40%, down from 100% for the original DPPM. Yet if $D = 64$, $M = 8$, the power efficiency drops from 175% to 150% using our method, which is relatively small in this case. From the above discussion, we can conclude that for lower M values, EDPPM doesn't improve the protocol efficiency, yet for higher values of M, it increases protocol efficiency significantly.

We can improve protocol efficiency more while hurting less on the power efficiency using a CDPPM frame. The results are shown in Figures 8 and 9. From Figure 8, we can see protocol efficiency increasing from 48% to 74% for $M = 4$. Nonetheless, for higher M values of like $M = 8$, the increment is little. Thus, the CDPPM frame structure is good for relatively lower values of M. From Figure 9, we can also see the reduction in power efficiency is less than the EDPPM frame (Figure 7) for $M = 4$.

VI. Conclusion

This paper has presented options for improving bandwidth efficiency for DPPM modulation schemes. Two novel frame structures, namely, EDPPM and CDPPM, have been designed. EDPPM provides better results for a relatively higher value of M, while CDPPM provides better results for the lower value of M. Neither of them can increase bandwidth efficiency for the lower value of M like $M = 2$. Generally, the case of $M = 2$, is not used for DPPM, because it doesn't improve power efficiency which is the main advantage of DPPM. We have studied the frame overhead and provided analytical estimates. We have validated the advantages of EDPPM and CDPPM through simulation. The simulation results have confirmed that our designed frame structure outperforms the conventional DPPM method.

Acknowledgment: This work is supported by the National Science Foundation, USA, Contract #0000010465.

Reference

[1] M. S. Islam and M. F. Younis, "Analyzing visible light communication through air–water interface," *IEEE Access*, 7, pp. 123830-123845, 2019.

[2] J. Wang, C. Lu, S. Li, and Z. Xu, "100 m/500 Mbps underwater optical wireless communication using an NRZ-OOK modulated 520 nm laser diode," *Optics Express*, Vol. 27, pp. 12171-12181, 2019.

[3] X. Liu, S. Yi, R. Liu, L. Zheng and P. Tian, "34.5 m underwater optical wireless communication with 2.70 Gbps data rate based on a green laser with NRZ-OOK modulation," *Proc. of the 14th China Int'l Forum on Solid State Lighting,* China, Beijing, pp. 60-61, 2017.

[4] D. Zhang, Y. Zhu, and Y. Zhang, "Multi-LED phase- shifted OOK modulation based visible light communication systems," *IEEE Photonics Technology Letters*, Vol. 25, December 2013.

[5] J. Liu, B. Zheng, L. Zhao and Z. Gong, "A design of underwater wireless laser communication system based on PPM modulating method," *Proc of OCEANS 2015 - MTS/IEEE Washington*, Washington, DC, pp. 1-6, 2015.

[6] S. Meihong, Y. Xinsheng and Z. Zhangguo, "The modified PPM modulation for underwater wireless optical communication," *Proc. of the Conf. on Comm. Software & Nets*, Macau, China, pp. 173-177, Feb. 2009.

[7] B. Bai, Z. Xu and Y. Fan, "Joint LED dimming and high capacity visible light communication by overlapping PPM," *Proc. 19th Annual Wireless and Optical Comm. Conf. (WOCC 2010)*, Shanghai, China, May 2010.

[8] J. E. Gancarz, H. Elgala, and T. D.C. Little, "Overlapping PPM for band-limited visible light communication and dimming," *Journal of Solid State Lighting*, vol. 2, no.1, May 2015.

[9] E. Elsayed, B. Yousif and M. Alzalabani, "Performance enhancement of the power penalty in DWDM FSO communication using DPPM and OOK modulation," Optical and Quantum Electronics, vol. 50, June 2018.

[10] D. Shiu and J. M. Kahn, "Differential pulse-position modulation for power-efficient optical communication," *IEEE Transactions on Communications*, vol. 47, no. 8, pp. 1201-1210, Aug. 1999.

[11] G. A. Mahdiraji and E. Zahedi, "Comparison of selected digital modulation schemes (OOK, PPM and DPIM) for wireless optical communications," *Proc. of 4th Student Conference on Research and Development*, Selangor, Malaysia, pp. 5-10, June 2006.

[12] J. Ma, Y. Jiang, S. Yu, L. Tan and W. Du, "Packet error rate analysis of OOK, DPIM and PPM modulation schemes for ground-to-satellite optical communications," *Optics Comm.*, vol. 283, pp. 237-242, January 2010.

[13] N. M. Aldibbiat, Z. Ghassemlooy and R. McLaughlin, "Dual header pulse interval modulation for dispersive indoor optical wireless communication systems," *IEE Proceedings - Circuits, Devices and Systems*, vol. 149, no. 3, pp. 187-192, June 2002.

[14] M. Pulkkinen, T. Haapala, J. Salomaa and K. Halonen, "45.2% Energy efficiency improvement of UWB IR Tx by use of differential PPM in 180nm CMOS," *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, Montreal, QC, pp. 193-196, 2016

[15] J. Wang, Z. Xu, and W. Hu, "Improved DPPM modulation for optical wireless communications," *Proc. SPIE 5281, Optical Transmission, Switching, and Subsystems,* vol. 5281, May 2004.

[16] H. Ai-ping, F. Yang-Yu, L. Yuan-Kui, J. Meng, B. Bo and T. Qing-Gui, "A differential pulse position width modulation for optical wireless communication," *Proc. of the 4th IEEE Conference on Industrial Electronics and Applications, Xi'an*, China, pp. 1773-1776, 2009.

[17] M. S. Islam, and M. Younis, "An Adaptive DPPM for Efficient and Robust Visible Light Communication across the Air-Water Interface" *Proc. of for 29th Wireless and Optical Communications Conference (WOCC), Newark, NJ, May 2020.*