2

Transmission Concept

A super block is one symbol, which contains a number of blocks. Each block contains a number of time slots. For each super block length in blocks, we calculate
1. Number of ways to organize the blocks in a super block

$$n!$$

2. Number of Bits / Symbol

$$log_2(n!)$$

3. Number of Bits / Photon

$$\frac{log_2(n!)}{n}$$

4. Number of Bits / Time Slot

$$\frac{log_2(n!)}{n} \times \frac{n}{T}$$

For example, we have [1,2,3,4], there are 4! = 24 permutation of ways to organize the blocks to generate different super blocks representing the corresponding symbols as follow:

$[1,2,3,4] \rightarrow A, [1,2,4,3] \rightarrow B, [1,3,2,4] \rightarrow C, [1,3,4,2] \rightarrow D, [1,4,2,3] \rightarrow E, [1,4,3,2] \rightarrow F$

$[2,1,3,4] \rightarrow G, [2,1,4,3] \rightarrow H, [2,3,1,4] \rightarrow I, [2,3,4,1] \rightarrow J, [2,4,1,3] \rightarrow K, [2,4,3,1] \rightarrow L$

$[3,1,2,4] \rightarrow M, [3,1,4,2] \rightarrow N, [3,2,1,4] \rightarrow O, [3,2,4,1] \rightarrow P, [3,4,1,2] \rightarrow Q, [3,4,2,1] \rightarrow R$

$[4,1,2,3] \rightarrow S, [4,1,3,2] \rightarrow T, [4,2,1,3] \rightarrow U, [4,2,3,1] \rightarrow V, [4,3,1,2] \rightarrow W, [4,3,2,1] \rightarrow X$

The information content of the super block is

$$log_2(4!) = 4.6 \quad \text{bits/symbol}$$

For each photon, it contains

$$1.15 \quad \text{bits/photon}$$

For each time slot, it has

$$0.33 bits/timeslot$$

3

Our Method

We define a block as an integral number of time bins (or time slots, or other encoding sources that are orthogonal, i.e., that can be perfectly discriminated).

Our method uses 4 photons in 14 time slots
It means that our method has:
1. $4! = 24$ ways to order them
2. 4.6 bits/symbol
3. 1.15 bits/photon
4. 0.33 bits / time slot

	Number of Photon	Number of Permutation	Number of Bits per Symbol	Number of Bits per Photon	Number of Bits per Time Slots
0	1.000000	1.000000	0.000000	0.000000	0.000000
1	2.000000	2.000000	1.000000	0.500000	0.000000
2	3.000000	6.000000	2.584963	0.861654	0.000000
3	4.000000	24.000000	4.584963	1.146241	0.000000
4	5.000000	120.000000	6.906891	1.381378	0.000000
5	6.000000	720.000000	9.491853	1.581976	0.000000
6	7.000000	5040.000000	12.299208	1.757030	0.000000
7	8.000000	40320.000000	15.299208	1.912401	0.000000
8	9.000000	362880.000000	18.469133	2.052126	0.000000
9	10.000000	3628800.000000	21.791061	2.179106	0.000000
10	11.000000	39916800.000000	25.250493	2.295499	0.000000
11	12.000000	479001600.000000	28.835455	2.402955	0.000000
12	13.000000	6227020800.000000	32.535895	2.502761	0.000000
13	14.000000	87178291200.000000	36.343250	2.595946	2.596000
14	15.000000	1307674368000.000000	40.250140	2.683343	2.683000
15	16.000000	20922789888000.000000	44.250140	2.765634	2.766000
16	17.000000	355687428096000.000000	48.337603	2.843388	2.843000
17	18.000000	6402373705728000.000000	52.507528	2.917085	2.917000
18	19.000000	121645100408832000.000000	56.755456	2.987129	2.987000
19	20.000000	2432902008176640000.000000	61.077384	3.053869	3.054000

Figure 3.1: Table

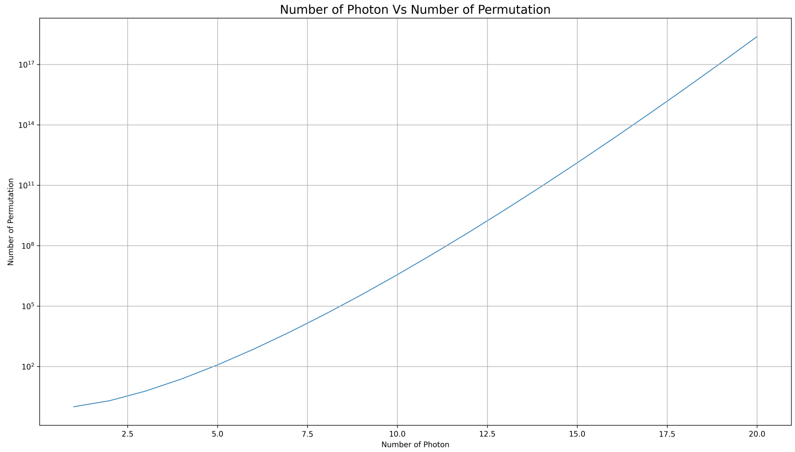

Figure 3.2: Number of Photons Vs Number of Permutation

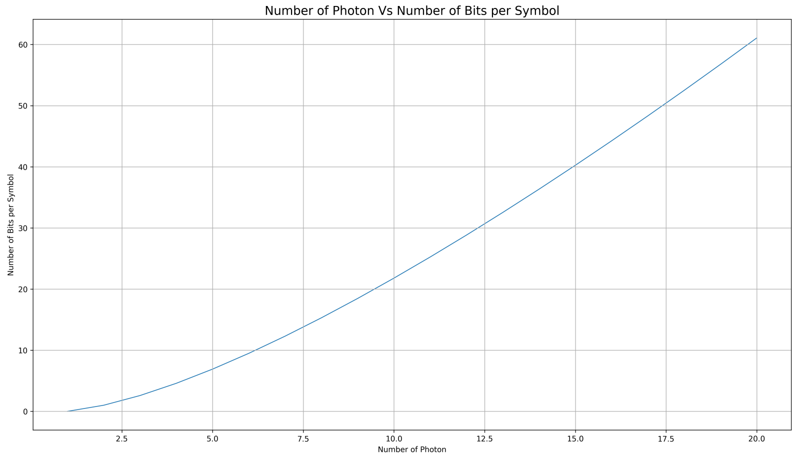

Figure 3.3: Number of Photons Vs Number of Bits per Symbol

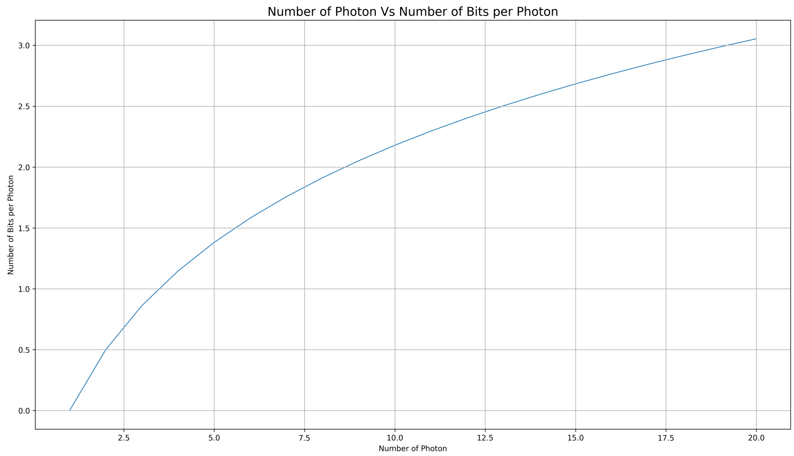

Figure 3.4: Number of Photons Vs Number of Bits per Photon

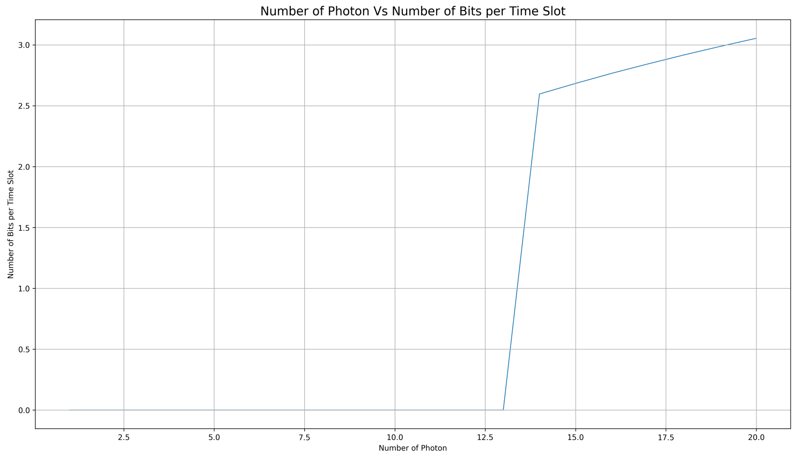

Figure 3.5: Number of Photons Vs Number of Bits per Time Slot

4

PPM

The representation of bits to symbol is as follow:
100000 $\rightarrow A$
010000 $\rightarrow B$
001000 $\rightarrow C$
000100 $\rightarrow D$
000010 $\rightarrow E$
000001 $\rightarrow F$

PPM uses 1 photon in 14 time slots
It means that PPM has:
1. 14 ways to order them
2. 3.8 bits/symbol
3. 3.8 bits/photon
4. 0.27 bits / time slot

[?]

	Number of Photon	Number of Permutation	Number of Bits per Symbol	Number of Bits per Photon	Number of Bits per Time Slots
0	1.000000	14.000000	3.807355	3.807355	0.000000
1	2.000000	14.000000	3.807355	1.903677	0.000000
2	3.000000	14.000000	3.807355	1.269118	0.000000
3	4.000000	14.000000	3.807355	0.951839	0.000000
4	5.000000	14.000000	3.807355	0.761471	0.000000
5	6.000000	14.000000	3.807355	0.634559	0.000000
6	7.000000	14.000000	3.807355	0.543908	0.000000
7	8.000000	14.000000	3.807355	0.475919	0.000000
8	9.000000	14.000000	3.807355	0.423039	0.000000
9	10.000000	14.000000	3.807355	0.380735	0.000000
10	11.000000	14.000000	3.807355	0.346123	0.000000
11	12.000000	14.000000	3.807355	0.317280	0.000000
12	13.000000	14.000000	3.807355	0.292873	0.000000
13	14.000000	14.000000	3.807355	0.271954	0.272000
14	15.000000	14.000000	3.807355	0.253824	0.254000
15	16.000000	14.000000	3.807355	0.237960	0.238000
16	17.000000	14.000000	3.807355	0.223962	0.224000
17	18.000000	14.000000	3.807355	0.211520	0.212000
18	19.000000	14.000000	3.807355	0.200387	0.200000
19	20.000000	14.000000	3.807355	0.190368	0.190000

Figure 4.1: Table

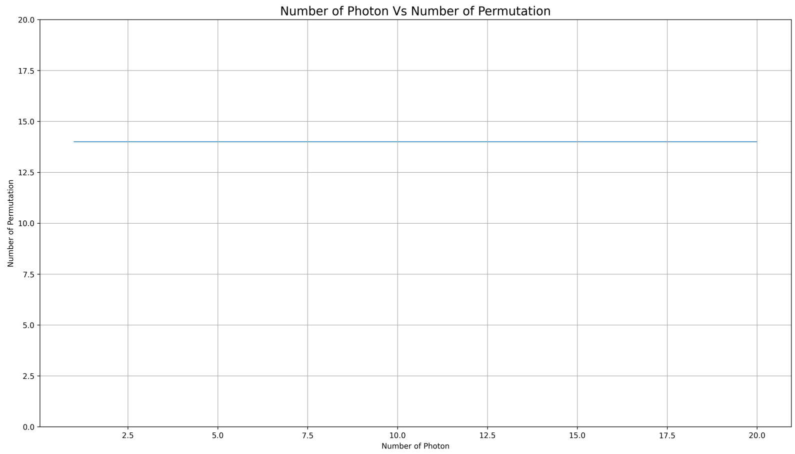

Figure 4.2: Number of Photons Vs Number of Permutation

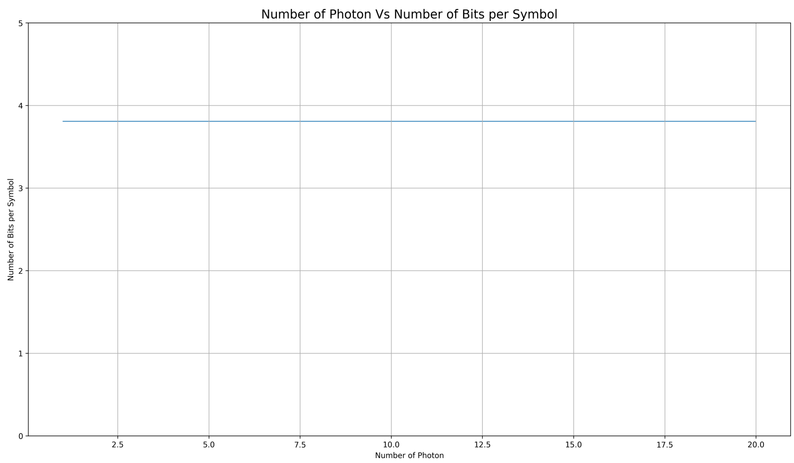

Figure 4.3: Number of Photons Vs Number of Bits per Symbol

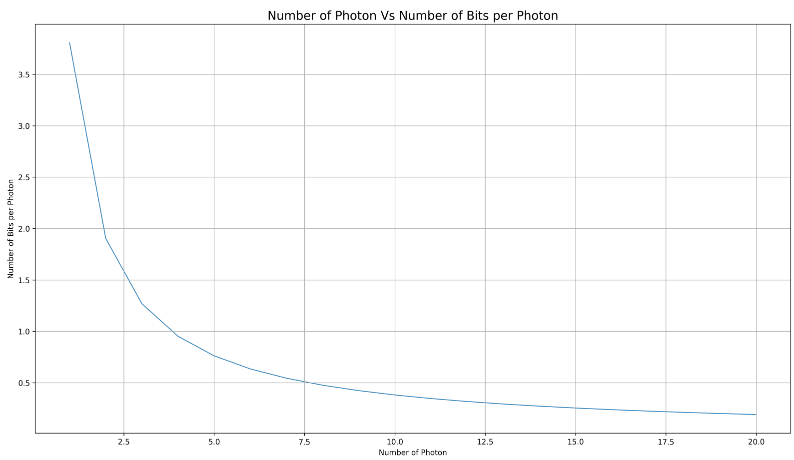

Figure 4.4: Number of Photons Vs Number of Bits per Photon

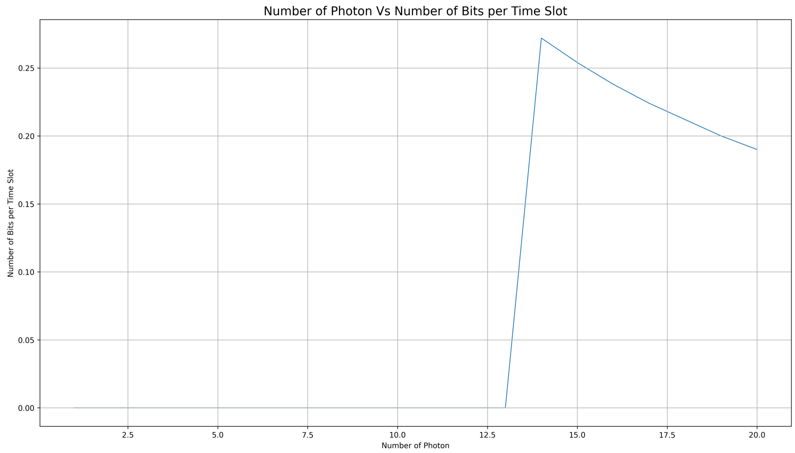

Figure 4.5: Number of Photons Vs Number of Bits per Time Slot

5

On-Off Key (OOK)

OOK uses 7 photons in average in 14 time slots
It means that PPM has:
1. $2^{14} = 16,384$ ways to order them
2. 14 bits/symbol
3. 2 bits/photon
4. 1 bits / time slot

	Number of Photon	Number of Permutation	Number of Bits per Symbol	Number of Bits per Photon	Number of Bits per Time Slots
0	1.000000	16384.000000	14.000000	14.000000	0.000000
1	2.000000	16384.000000	14.000000	7.000000	0.000000
2	3.000000	16384.000000	14.000000	4.666667	0.000000
3	4.000000	16384.000000	14.000000	3.500000	0.000000
4	5.000000	16384.000000	14.000000	2.800000	0.000000
5	6.000000	16384.000000	14.000000	2.333333	0.000000
6	7.000000	16384.000000	14.000000	2.000000	0.000000
7	8.000000	16384.000000	14.000000	1.750000	0.000000
8	9.000000	16384.000000	14.000000	1.555556	0.000000
9	10.000000	16384.000000	14.000000	1.400000	0.000000
10	11.000000	16384.000000	14.000000	1.272727	0.000000
11	12.000000	16384.000000	14.000000	1.166667	0.000000
12	13.000000	16384.000000	14.000000	1.076923	0.000000
13	14.000000	16384.000000	14.000000	1.000000	1.000000
14	15.000000	16384.000000	14.000000	0.933333	0.933000
15	16.000000	16384.000000	14.000000	0.875000	0.875000
16	17.000000	16384.000000	14.000000	0.823529	0.824000
17	18.000000	16384.000000	14.000000	0.777778	0.778000
18	19.000000	16384.000000	14.000000	0.736842	0.737000
19	20.000000	16384.000000	14.000000	0.700000	0.700000

Figure 5.1: Table

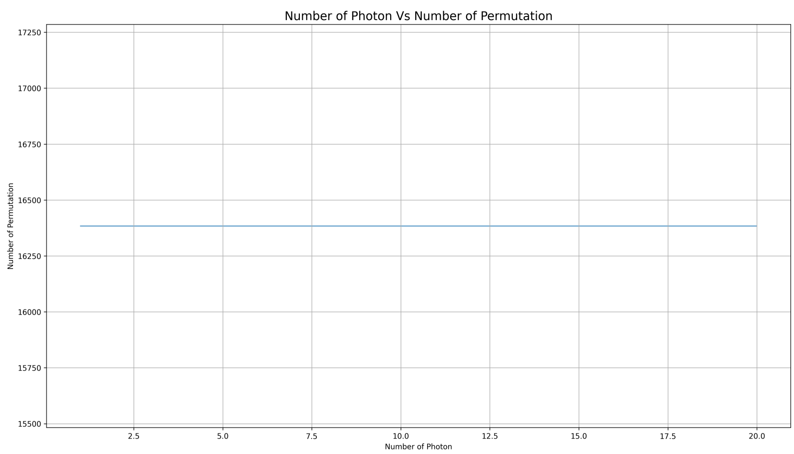

Figure 5.2: Number of Photons Vs Number of Permutation

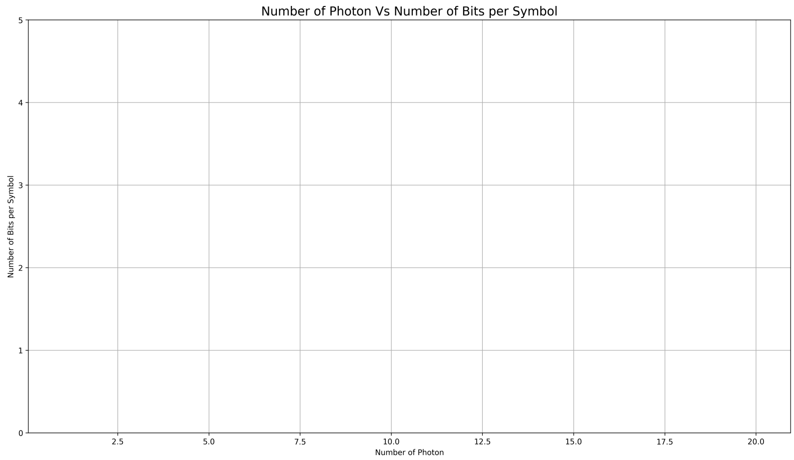

Figure 5.3: Number of Photons Vs Number of Bits per Symbol

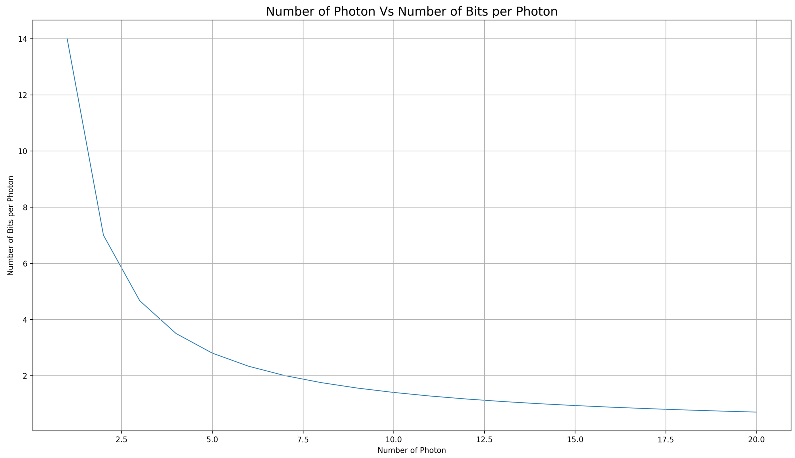

Figure 5.4: Number of Photons Vs Number of Bits per Photon

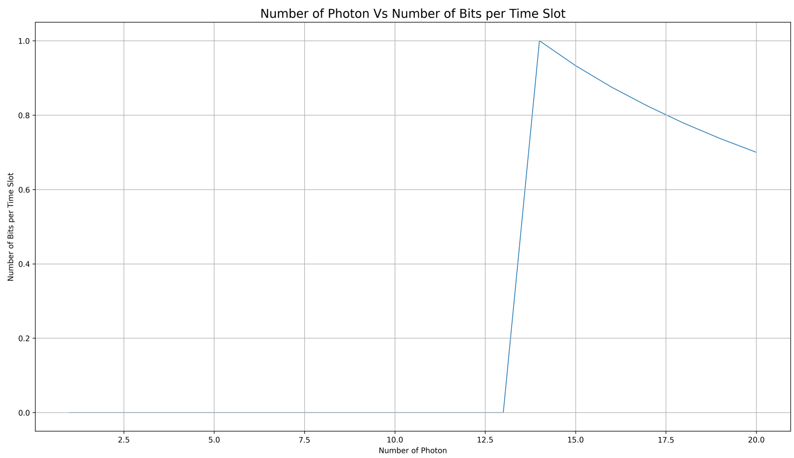

Figure 5.5: Number of Photons Vs Number of Bits per Time Slot

6
In General

In general, it takes 4 photons in 14 time slots
It means that it has:
1. 1,001 ways to order them
2. 10 bits/symbol
3. 2.5 bits/photon
4. 0.71 bits / time slot

7

Reed-Solomon Codes

The Reed-Solomon (RS) codes are the non-binary codes, they are important for the use in communication systems where errors appear in bursts rather than independent random errors.

RS codes were discovered by Reed and Solomon in 1960. The non-binary BCH block codes have $2^m(\{0, 1, 2, \ldots, 2^m - 1\})$ symbols with block length $n = 2^m - 1$, which can be extended to $n = 2^m$ or $m = 2^m + 1$. RS codes can correct up to e_0 errors within a block of n symbols by using $n - k = n - 2e_0 = 2^m - 1 - 2e_0$ parity symbols.

RS code can aciheve the maximum number of error correction by finding the largest possible $d_{min} = 2e_0 + 1$