ATLANTIS PRESS
*Advances in Computer Science Research, volume 65*
3rd International Conference on Communications, Information Management and Network Security (CIMNS 2018)

# Research and Exploration on a New Method of Finding the Blocking Probability

Fan Bu and Yi Sun*
School of Software Engineering, State Key Laboratory of Networking and Switching Technology Beijing University of Posts and Telecommunications
*Corresponding author

***Abstract*—This electronic document is the exploration of using the blocking probability without the wavelength conversion link to predict the blocking probability with the wavelength conversion**

***Keywords*—DWM; blocking probability; links; network**

## I. INTRODUCTION

Network blocking is a phenomenon when a portion of a communication subnet has too many packets. Make this part of the network have no time to deal with them, So that this part and even the entire network performance decline. [1]

Blocking probability is the probability that the signal will be blocked during transmission. At same time, it is a variable used to predict the probability of network blocking

Wavelength Conversion can effectively improve link utilization in network transmission, therefore, it is necessary to use Wavelength Conversion. However, it is limited by many factors, the number of wavelengths available is still limited, at the same time, the technology is not enough to support the application of a large number of nodes. [2] This requires us to use the Wavelength Conversion most efficiently, and so that leads to: How many wavelength converters are set? Where is the wavelength converter set? And so on.

These problems require a lot of theoretical calculation and experimental verification. In order to carry out the theoretical calculation smoothly, a stable and reliable method which is used to estimate network transmission efficiency is necessary. Therefore, the study of blocking probability becomes an inevitable trend.

## II. MODEL ANALYSIS

### A. Actual Analysis

In today's network communications, we often use WDM technology. WDM is a system in which one fiber carries multiple wavelengths. Convert 1 fiber to multiple "virtual" fibers. Each virtual fiber works independently at different wavelengths, this greatly increases the transmission capacity of the optical fiber. [3]

### B. Sorting Model

There are a series of nodes throughout the link network. Every two nodes is connected by a fixed number of links. Each of these links contains a certain number of road which has different wavelength. In one of the links, a wavelength has only one road. The arrival of data follows a poisson process with a velocity of μ. Service time obeys the poisson distribution of $\lambda_0$

## III. ANALYSIS AND PRELIMINARY SOLUTION

In the usual way of calculating, Direct extrapolation with wavelength converters is often used, So it will be a little bit more complicated, The formulas are naturally more complex. So in this paper, we first analyze the absence of wavelength converter. Then analyze the network with Wavelength Conversion.

To connect the two cases smoothly, this paper proposes a new method. By solving: The probability of k paths where the wavelength is a certain value (λ) and between two adjacent certain nodes (i, j) all being blocked (This variable are recorded as ( $C_k$) to solve the problem.

In the following section we will try to find $C_k$, use the method which combine the blocking probability get from the two different ways.

### A. Solution Method 1

Firstly, we use the method assuming that there is a feasible wavelength to consider the solution of transmission success probability among nodes:

Let's start with the analysis the probability of using a certain wavelength (λ) to smoothly pass the link between all nodes, denote as $P_\lambda$.

$$P_\lambda = (1 - C_k)^H \tag{1}$$

$H + 1$ denote: Number of nodes that need to pass through for transmission.

In this way, the probability that a signal cannot be transmitted smoothly over a certain wavelength which recorded as $C_0$ can be solved.

$$C_0 = 1 - P_\lambda = 1 - (1 - C_k)^H \tag{2}$$

So the probability that a signal cannot be transmitted smoothly over all wavelengths which recorded as $C_1$ can be solved. That's what we call the blocking probability

$$C_1 = (1 - P_\lambda)^W = (1 - (1 - C_k)^H)^W \tag{3}$$

Copyright © 2018, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

W denote: the number of wavelength categories contained in each link.

*B. Solution Method 2*

And then we'll do it other way, Look for a wavelength that can be used to transmit a signal across all nodes.

First we introduce a set of random variables$\{X_1, X_2, X_3, \ldots, X_H\}$ To record the number of free wavelengths for H-segment transmission process formed by dividing the H+1 points passing through the transmission .At the same time introduce a new variable $X_m$ to record the number of idle wavelengths in all processes. It's not hard to see what the blocking probability ($C_1$) is:

$$C_1 = P_r \cdot \{X_m = 0\} \quad (4)$$

Since our conclusion is relatively simple, it is difficult to think further. Therefore, Use the knowledge of probability theory [4] to expand it:

$$C_1 = P_r\{X_m = 0 \mid X_1 = m_1, X_2 = m_2, X_3 = m_3 \ldots, X_H = m_H\} \cdot P_r\{X_1=m_1, X_2=m_2, X_3=m_3, \ldots X_H=m_H\} \quad (5)$$

Let's consider one of the easier scenarios: $P_r\{X_m = n \mid X_1 = m_1, X_2=m_2\}$Know from the classical models of probability, [4] The value of the operation is as follows when there is only one link between each node:

$$P_r\{X_m = n \mid X_1=m_1, X_2=m_2\} = \binom{m_1}{n}\frac{\binom{c-m_2}{m_1-n}}{\binom{c}{m_1}} \quad (6)$$

And we go on to think about, In cases where there is only one link between each node, The probability that the number of wavelengths available for a link is less than n. Result are as follows:

$$P_r\{X_m < n \mid X_1=m_1, X_2=m_2\} = \sum_{k=0}^{n-1} P_r\{X_m = k \mid X_1 = m_1, X_2 = m_2\} = \sum_{k=0}^{n-1}\binom{m_1}{n}\frac{\binom{W-m_2}{m_1-n}}{\binom{W}{m_1}} \quad (7)$$

And then we can solve $P_{rk}\{X_m = n \mid X_1=m_1, X_2=m_2\}$, In the case of k links between each node

$$P_{rk}\{X_m = n \mid X_1 = m_1, X_2 = m_2\}$$

$$= \sum_{i=0}^{k-1}\binom{k}{i}\left(\binom{m_1}{n}\frac{\binom{W-m_2}{m_1-n}}{\binom{W}{m_1}}\right)^{k-i} \cdot \left(\sum_{k=0}^{n-1}\binom{m_1}{n}\frac{\binom{W-m_2}{m_1-n}}{\binom{W}{m_1}}\right)^{i} \quad (8)$$

$$= \left(\sum_{k=0}^{n}\binom{m_1}{n}\frac{\binom{W-m_2}{m_1-n}}{\binom{W}{m_1}}\right)^{k} - \left(\sum_{k=0}^{n-1}\binom{m_1}{n}\frac{\binom{W-m_2}{m_1-n}}{\binom{W}{m_1}}\right)^{k} \quad (9)$$

Transformation from (8) to (9) use knowledge of binomial theorem. [5]

W denote: the number of wavelength categories contained in each link.

k denote: the number of links between two nodes.

We find that the original formulas can be summed up as simple models. Result are as follows:

$$C_1(X_1=m_1, X_2=m_2, X_3=m_3, \ldots X_H=m_H) =$$

$$\sum_{k=0}^{m_H-1}\sum_{m_{H'}}^{W} C_1(X_1 = k, X_2 = m_H') * P_r\{X_m = k \mid X_1=m_1, X_2=m_2, X_3=m_3, \ldots X_{H-1}=m_{H-1}\} * P_r\{X_H=m_H'\} \quad (10)$$

But there is still a problem, the solutions of $P_r\{X_H=m_H'\}$ still don't know. Let's suppose that $\{X_l\}$ is independent. So we get it by Erlang formula [6] when there is only one link between each node.

$$P_r\{X_H=m_H'\} = \frac{\lambda_0^{W-m}}{W-m_{H'}}\left(\sum_{k=0}\frac{\lambda_0^{k}}{k!}\right)^{-1} \quad (11)$$

$$= \frac{\lambda_0^{W-m}}{c-m_{H'}}e^{\lambda_0} \quad (12)$$

Transformation from (11) to (12) use knowledge of series. [5]

And we go on to think about, In cases where there is only one link between each node, the value of $P_r\{X_H<m_H'\}$. Result are as follows:

$$P_r\{X_H<m_H'\} = \sum_{i=0}^{m_H'-1}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0} \quad (13)$$

And then we can solve $P_{rk}\{X_H=m_H'\}$, In the case of k links between each node

$$P_r\{X_H=m_H'\}$$

$$= \sum_{i=0}^{k-1}\binom{k}{i}\left(\frac{\lambda_0^{W-m}}{c-m_{H'}}e^{\lambda_0}\right)^{k-i}\left(\sum_{i=0}^{m_H'-1}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0}\right)^{i} \quad (14)$$

$$= \left(\left(\frac{\lambda_0^{W-m}}{W-m_{H'}}e^{\lambda_0}\right) + \left(\sum_{i=0}^{m_H'-1}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0}\right)\right)^{k} - \binom{k}{k}\left(\sum_{i=0}^{m_H'-1}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0}\right)^{k} \quad (15)$$

$$= \left(\sum_{i=0}^{m_H'}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0}\right)^{k} - \left(\sum_{i=0}^{m_H'-1}\frac{\lambda_0^{W-m}}{W-i}e^{\lambda_0}\right)^{k} \quad (16)$$

Transformation from (14) to (15 and (16) use knowledge of binomial theorem. [15]

W denote: the number of wavelength categories contained in each link.

m denote: the number of wavelength types remaining between all nodes

This is a recursive formula and we need to solve it programmatically. The solution implementation program is given as follows(C programming language for example)

(i) the data structure definition is implemented:

typdef Node{

int mark;

int wave_number;

};

Node list[H+1];

Initialize with {$X_1$, $X_2$, $X_3$,...….., $X_H$}

(ii)Define the solve function of equation (6)

int factorial(int n){

// This function is used to recursively solve the factorial

if(n==1||n==0)return(1);

else return(n*factorial(n-1));

}

int combinatorial_number(int n ,int m){

// A function used to solve combinations Numbers$\binom{n}{m}$

return (factorial(m)/ (factorial(n)* factorial(m-n)))

}

int p_rnm(int n,int m1,int m2){

// function that needs to be defined

return(combinatorial_number( $m_1$ ,n)* combinatorial_number( $c-m_2$ , $m_1 -$ n)/ combinatorial_number(c,$m_1$))

}

(iii)Define the solve function of equation (9)

/*Because function power() is used,we should include the headfile which is named math.h*/

double sum_p_rnm(int n,int m1,int m2,int k)// Define the sum function

{

double sum=0;

for(i=1,i>k,i++){

sum+=p_rnm(n,m1,m2);

}

return(sum)

}

double pr_knm(int n,int m1,int m2,int k)// function that needs to be defined

{ return(power( sum_p_rnm(n, m1, m2, k),k)-power( sum_p_rnm(n, m1, m2, k-1),k) );

}

(iv)Define the solve function of blocking probability

int c_1(Node* list ,int list_length, int n){

/*The first parameter fills in the array name,

The second parameter fills in the current array length, The third element fills in the final number of remaining wavelength species, The last parameter fills in the number of links between two adjacent nodes */

if(list_length==2)return(p_rnm(0,list[0].wave_number, list[0].wave_number));

else {

int m=0;

for(i=0,i<=list[list_length-1].wave_number,i++){

for(j=n,j<=c,j++){

m+= p_krnm(n, k,j,2)* c_1(list , list_length-1, k)* $P_r${ $X_H$= $m_H{}'$}

}

}

}

return(m);

}

*C. Simultaneous Exploration*

Combine the results of the two methods

$$(1- P_\lambda)^{w} = (1-(1-C_k)^H)^{w} = C_1 = Cr_1 \quad (17)$$

So we can figure out:

$$C_k = \sqrt[H]{1-\sqrt[w]{Cr_1}} \quad (18)$$

So we get expression of $C_k$

## IV. Exploration of a Link Network With Wavelength Converters.

*A. Model Assumes*

We assume that in all the D nodes there is $D'$ nodes with wavelength converter. The Wavelength Conversion can only carry out partial wavelength conversion, We assume that the Wavelength Conversion can only convert $W_1$ of W wavelength.

*B. Analyze the Blocking Probability*

Know from the classical models of probability, we can know the probability of H^’ nodes has Wavelength Conversion when there is H nodes in total:

$$P_{H+1}^{H'} = \frac{\binom{D-D'}{H+1-H'}\binom{D'}{H'}}{\binom{D}{H+1}} \quad (19)$$

Know from the classical models of probability, we can know the probability of the last node happens to have a Wavelength Conversion when $H'$ nodes has Wavelength Conversion when there is H nodes in total:

$$P_{H+1}^{H'\,'\,'}=\frac{\binom{H}{H'-1}A_{H+1-H'}^{H+1-H'}A_{H'-1}^{H'-1}}{\binom{H+1}{H'}A_{H+1-H'}^{H+1-H'}A_{H'}^{H'}} \tag{20}$$

Get rid of the same term

$$P_{H+1}^{H'\,'\,'}=\frac{\binom{H}{H'-1}A_{H'-1}^{H'-1}}{\binom{H+1}{H'}A_{H'}^{H'}} \tag{21}$$

Get rid of the same term after swap the number of permutations in

$$P_{H+1}^{H'\,'\,'}=\frac{1}{H+1} \tag{22}$$

We divided the block rate into two parts to think about:

1. The blocking probability of the wavelength served by the Wavelength Conversion ($C_a$)

2. The blocking probability of the wavelength without the Wavelength Conversion serving it($C_b$)

So the final blocking probability is going to be

$$C_2=P_{H+1}^{H'}C_aC_b \tag{23}$$

First, the link with the wavelength converter is analyzed,

$$C_a=(1-P_{H+1}^{H'\,'\,'})\{1-[1-(C_K)^{w_1}]^{H'}\}+P_{H+1}^{H'\,'\,'}\{1-[1-(C_K)^{w_1}]^{H'-1}\} \tag{24}$$

Substitute the (22) into the (24)

$$C_a=\frac{H}{H+1}\{1-[1-(C_K)^{w_1}]^{H'}\}+\frac{1}{H+1}\{1-[1-(C_K)^{w_1}]^{H'-1}\} \tag{25}$$

Subsequent analysis of the blocking probability of the wavelength without a Wavelength Conversion:

$$C_b=[1-（1-C_K)^{w-w_1}]^{H-H'} \tag{26}$$

combine (18), (25) and (26),can get blocking probability

## V. Inspection Of Results

### A. Use the Program to Solve the Blocking Probability

(i)Continue the initialization on the basis of the previous variable initialization:

int d=D;// Record the number of nodes

int d_=$D'$;// Record the number of nodes which have the Wavelength Conversion

int w=w;// Record the number of types of wavelengths

int w_=$w_1$;// Record the number of types of wavelengths which can change by the Wavelength Conversion

(ii)Define the solve function of (18)

/*Because function power() is used,we should include the headfile which is named math.h*/

```
doube c_k(int h,int k){
return(power(1-power(c_1(list,h,0,k),1/w),1/h));
}
```

(iii)Define the solve function of (19)

```
double p_h_h+1(int h,int h_){
return(combinatorial_number(d-d_,h+1-h_)* combinatorial_number(d_,h_)/ combinatorial_number(d,h+1))
}
```

(iv)Define the solve function of (25)

```
doube c_a(int h,int h_, int k){
/*h Number of nodes passing through minus one
h_ Number of nodes with Wavelength Conversion
k The number of links between adjacent nodes*/
s1=power( 1-power(c_k(w,h,k),1/w_),1/h_);
s2=power( 1-power(c_k(w,h,k),1/w_),1/(h_-1));
return((h/(h+1))*s1+(1/(h+1))*s2)
}
```

(v)Define the solve function of (26)

```
doube c_b(int h,int h_, int k){
/*h Number of nodes passing through minus one
h_ Number of nodes with Wavelength Conversion
k The number of links between adjacent nodes*/
return( power(1-(1-power(c_k(w, h, k),1/(w-w_)),1/(h-h_)));
}
```

(6) So the blocking probability is:

```
doube c_0(int h,int h_, int k){
return(c_k(h,k)* c_a(h, h_, k)* c_b(h, h_, k));
}
```

### B. Inspection

(i)Continue the initialization on the basis of the previous variable initialization:

int k=k;// Used to store the number of links between two adjacent nodes

(ii) the data structure definition is implemented:

```
typdef node_r{
int mark;
int h;
int h_;
```

double result;

};

node_r result[n];//the n which is in array should depends on the number of results

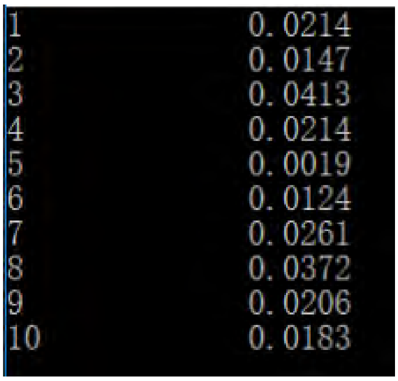
(iii) Define a function that recursively solves the variance

double variance(node_r* list,int n){

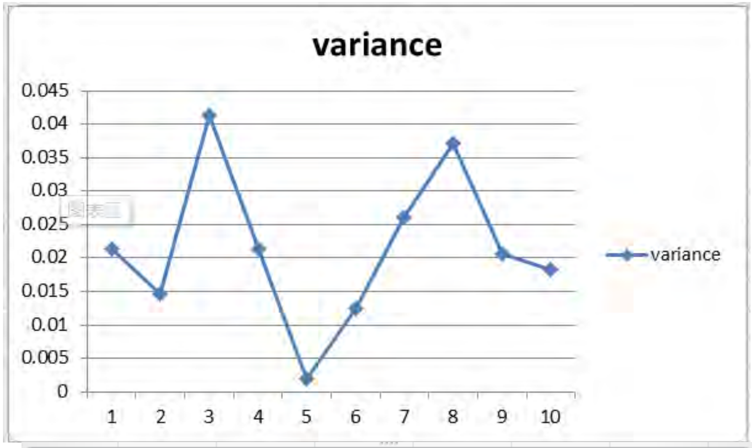
return (power(c_b(list[n-1].h, list[n-1].h_, k)-result, 2)+ variance(list, n-1));

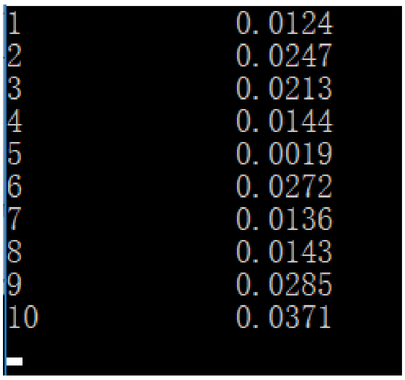
}

The test results are figure I:



| | |
|---|---|
| 1 | 0.0214 |
| 2 | 0.0147 |
| 3 | 0.0413 |
| 4 | 0.0214 |
| 5 | 0.0019 |
| 6 | 0.0124 |
| 7 | 0.0261 |
| 8 | 0.0372 |
| 9 | 0.0206 |
| 10 | 0.0183 |



| | |
|---|---|
| 1 | 0.0124 |
| 2 | 0.0247 |
| 3 | 0.0213 |
| 4 | 0.0144 |
| 5 | 0.0019 |
| 6 | 0.0272 |
| 7 | 0.0136 |
| 8 | 0.0143 |
| 9 | 0.0285 |
| 10 | 0.0371 |

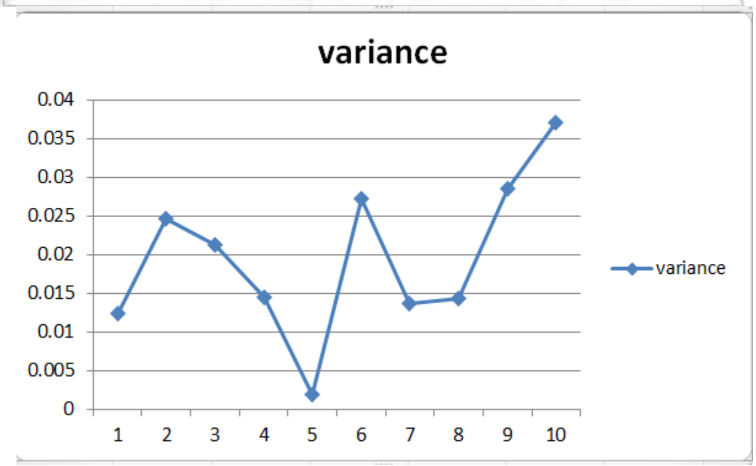



FIGURE I. THE RESULT OF TWO TESTING EXPERIMENTS .

## ACKNOWLEDGMENT

This research was financially supported by the 2018 Double-class notch talent competition Project.

## REFERENCES

[1] Xuanxi Ning , Blocking flow theory and its application (second edition), 9787030235985

[2] Gerd Keiser , Optical Fiber Communications, Fourth Edition , McGraw-Hill, 9787121161711

[3] WenWu , Optical wavelength division multiplexing system (revised), Publishing House of Electronics Industry 9787121104589

[4] ZhouSheng ShiqianXie ChengyiPan, Probability Theory and Mathematics Statistics, Higher Education Press

[5] Department of mathematics, tongji university, Advanced mathematics (7th edition),Higher Education Press

[6] JiashanTang, Queuing theory and its application, science press 9787030493897