Dynamic Bandwidth Allocation in GMPLS Optical Networks using Minimum Execution Time Technique

Srivatsan T

IITH

April 30, 2021

GMPLS - Generalised MPLS

MPLS-Multi Protocol Label Switching

1. Traditional IP methods used routing tables and destination IP's to route incoming packets.
2. MPLS uses LSP (Label Switched paths) and reduces reliance on routing tables.
3. MPLS is protocol independent

MPLS in action

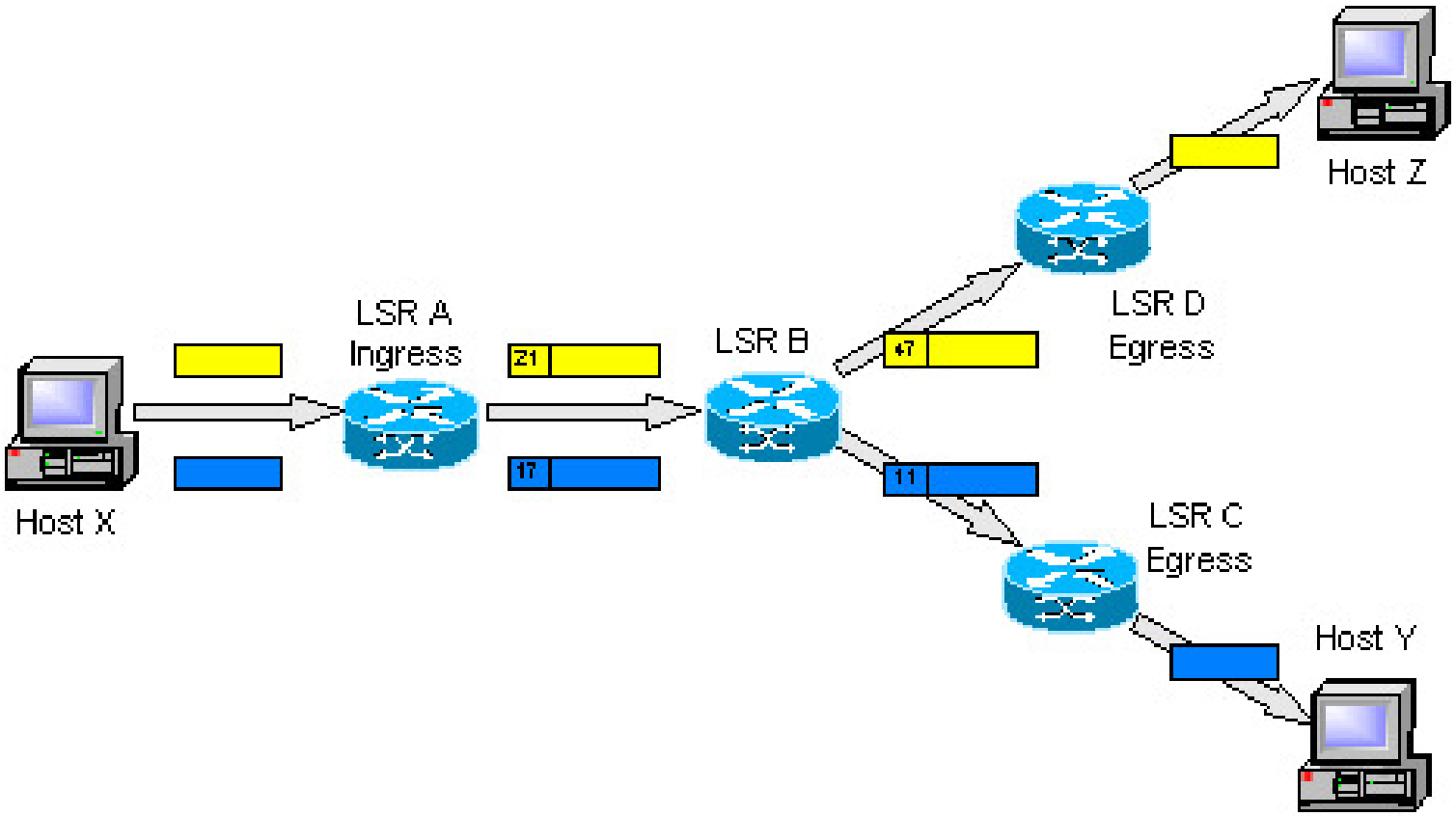


Figure: Label switching in MPLS connections

GMPLS

GMPLS uses implicit labels generated from the incoming packets itself

1. Timeslot to identify the LSP, on a Time Division Multiplexed (TDM) link
2. Wavelength to identify the LSP, on a Wavelength Division Multiplexed (WDM) link
3. Fiber or port on which a packet is received.

LSP traffic is switched based on a continuous, constant property of the data stream – making GMPLS a highly suitable protocol to run in high bandwidth networks.

Blocking Probability

Blocking-Definition

Blocking in telecommunication systems is when a circuit group is fully occupied and unable to accept further calls.

This can be handled in 2 ways

1. Unconnected users can be queued and connected again when servers are free
2. Unconnected user details can be cleared which leads to the call never being connected

Blocking Probability

Blocking probability-Definition

The probability that the call being made is not successfully connected immedietly. Alternatively it is the probability that the call being made by the user is either queued or cleared.

To improve quality of service (QoS) it is essential to minimize Blocking probability as much as we can.

Calculation of Blocking Probability

Let us first define some variables

Variables

1. N - Total number of servers available
2. λ - mean call arraival rate
3. Let the call timings be exponentially distributed with $\frac{1}{\mu}$ as its mean
4. μ - mean calls completion rate
5. q - size of erlang queue

Poisson's Distribution

Number of calls arrived and completed is modelled as a poisson random variable

Call Arrival

$$\text{P(m calls generated in time T)} = \frac{(\lambda T)^m}{m!}\ e^{-\lambda T}$$

Call completion

$$\text{P(m calls completed in time T)} = \frac{(\mu T)^m}{m!}\ e^{-\mu T}$$

Concept of Traffic

Definition

Offered Traffic is a measure of the number of concurrent calls that could be carried by the network if there were no restrictions on the number of servers.It is expressed in Erlangs (dimension less)

It is calculated by finding the busiest hour of the day and multiplying the average call time in that hour (in hours) and the number of calls that arrived in that hour.

Formula

Offered Traffic A $= \lambda \times \frac{1}{\mu}$ Erlangs

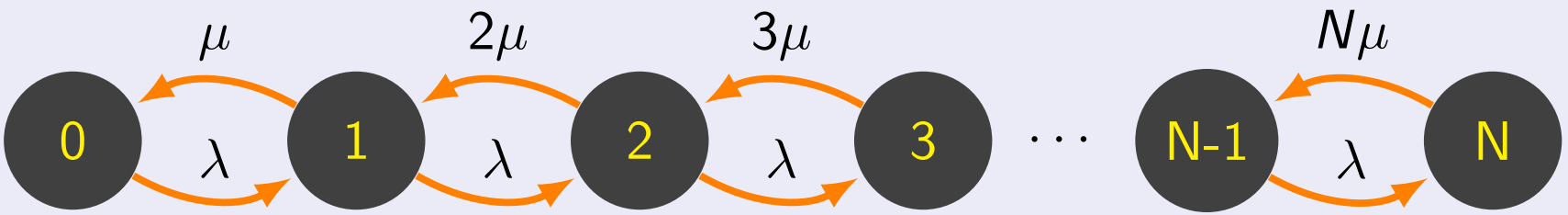
Erlang B Queuing model

In the Erlang B queueing model, customers arrive at a queueing system having n servers but having 0 waiting positions. We assume Poisson arrivals and exponentially distributed service times. Since there are no waiting positions, it is understood that if a customer finds upon arrival that no servers are available, then the customer goes away and is lost.

Birth Death Process

Definition

It is a kind of process in which there are only 2 possibilities for a system. It can either have a Birth which increments the state variable by 1 or it can have a Death which decrements the state variable by 1.



Let P_k denotes the probability that there are k callers in the system.

Birth-Death Contd.

$$P_1\mu = P_0\lambda$$

This can be generalised as follows

$$P_k \;\; k\mu = P_{k-1} \;\; \lambda \quad 1 \leq k \leq N$$

Which gives,

$$P_k = \frac{\lambda}{k\mu}P_{k-1} = \frac{\lambda^2}{k(k-1)\mu}P_{k-2} = \cdots = \frac{1}{k!}\left(\frac{\lambda}{\mu}\right)^k P_0 \tag{1}$$

We can calculate P_0 using the normalisation condition $\sum_{k=0}^{N} P_k = 1$
Replacing $\frac{\lambda}{\mu}$ with offered traffic A, We get the Blocking Probability as follows

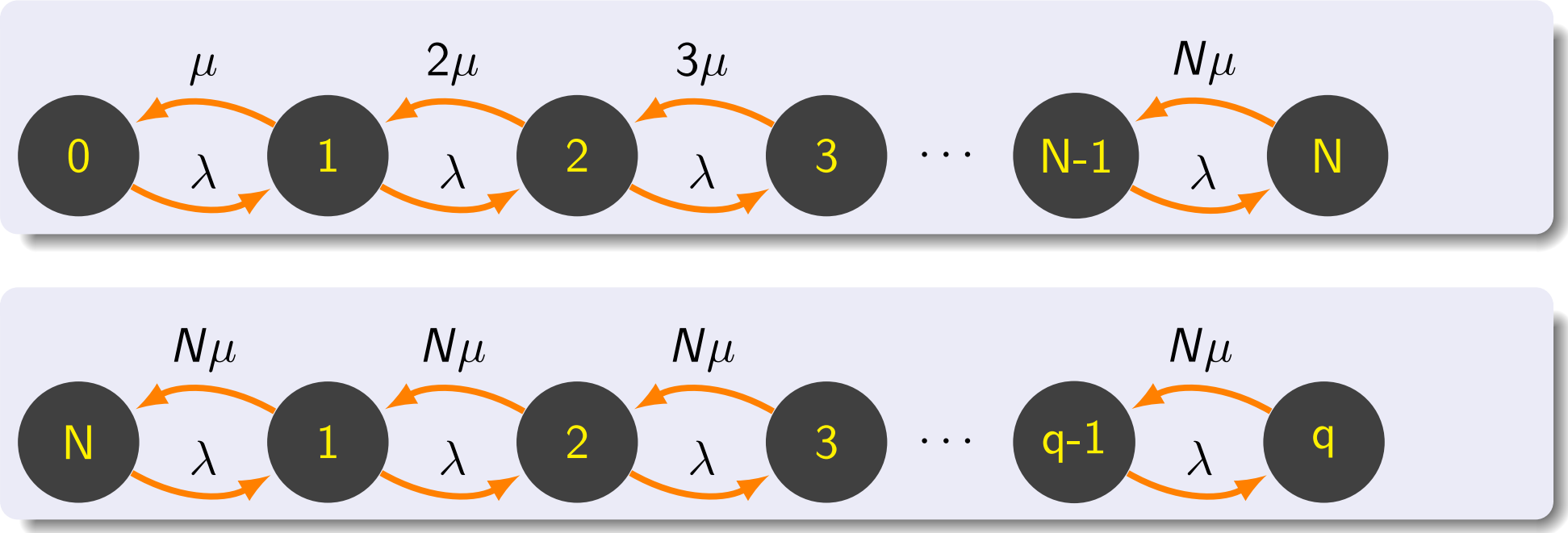
$$P_B = \frac{\frac{A^N}{N!}}{\sum_{k=0}^{N}\left(\frac{A^k}{k!}\right)} \tag{2}$$

Erlang C Queuing Model

Definition

In the Erlang C queueing model, customers arrive at a queueing system having n servers and q waiting positions. We also assume Poisson arrivals and exponentially distributed service times. If all servers are busy when a user tries to make a call, he is places in the queue and the call eventually gets connected.

Birth Death Model



Let P_k denote the probability that there are k callers in the system.

Birth Death Contd.

The regression in probaility for Erlang C queue is as follows,

$$P_k \;\; k\mu = P_{k-1} \;\; \lambda \quad 1 \leq k \leq N$$
$$P_k \;\; N\mu = P_{k-1} \;\; \lambda \quad N+1 \leq k \leq N+q$$

Which gives,

$$P_k = P_0 \frac{A^k}{k!} \quad 0 \leq k \leq N \tag{3}$$

$$P_k = P_{k-1}\left(\frac{A}{N}\right) = P_{k-2}\left(\frac{A}{N}\right)^2 \cdots = P_N\left(\frac{A}{N}\right)^{k-N} \tag{4}$$
$$N+1 \leq k \leq N+q$$

Blocking Probability in Erlang C Queues

We can replace (4) with P_N expression and we again use the normalisation to find P_0.

Normalisation

$$\sum_{i=0}^{N+q} P_i = 1 \tag{5}$$

$$\sum_{i=0}^{N} P_0 \frac{A^i}{i!} + \sum_{i=N+1}^{N+q} P_0 \frac{A^N}{N!} \left(\frac{A}{N}\right)^{i-N} = 1 \tag{6}$$

Blocking Probability contd.

Probability that the call doesn't get immedietly connected is the probability that the number of users present in the network is greater than or equal to N.

P_B

$$P_B = P_N + P_{N+1} + \cdots + P_{N+q} \tag{7}$$

$$= P_N + P_N\left(\frac{A}{N}\right) + + P_N\left(\frac{A}{N}\right)^2 + \cdots + P_N\left(\frac{A}{N}\right)^q \tag{8}$$

$$= P_N\left(\frac{1-\left(\frac{A}{N}\right)^{q+1}}{1-\left(\frac{A}{N}\right)}\right) \tag{9}$$

P_B Contd.

$$P_B = P_0 \left(\frac{A^N}{N!}\right) \left(\frac{1 - \left(\frac{A}{N}\right)^{q+1}}{1 - \left(\frac{A}{N}\right)}\right) \tag{10}$$

$$P_B = \frac{\left(\frac{A^N}{N!}\right) \left(\frac{1-\left(\frac{A}{N}\right)^{q+1}}{1-\left(\frac{A}{N}\right)}\right)}{\sum_{i=0}^{N} \frac{A^i}{i!} + \left(\frac{A^N}{N!}\right) \left(\frac{1-\left(\frac{A}{N}\right)^{q+1}}{1-\left(\frac{A}{N}\right)}\right)} \tag{11}$$

MEt

MEt algorithm finds the resource which has minimum execution time to complete the task. It allocates the work to the resource based on first comes first serve basis. The algorithm is least concerned with the amount of data to be transferred. The user requesting for connection first is alloted the least time consuming server predicted by the algorithm.

1. Enter user,server positions, bandwidth and link rates
2. Evaluate the coverage area and maintain routing table
3. Run the algorithm and find the server with least time
4. Connect the server to the demanding user by providing wavelength and link rates
5. Evaluate performance using Blocking Probability and latency

Results and Graphs

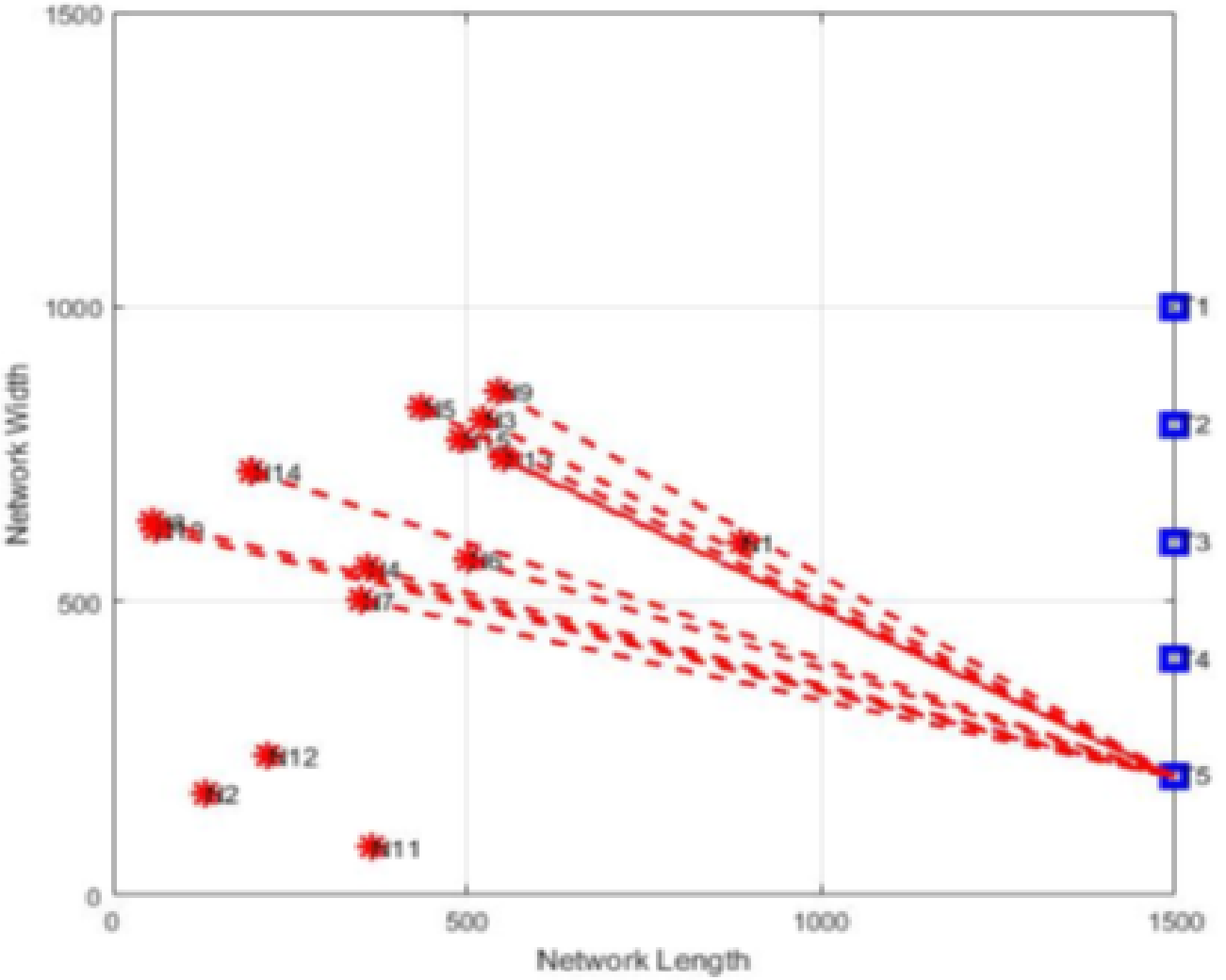


Figure: Simulations using said model and algorithm

Figures

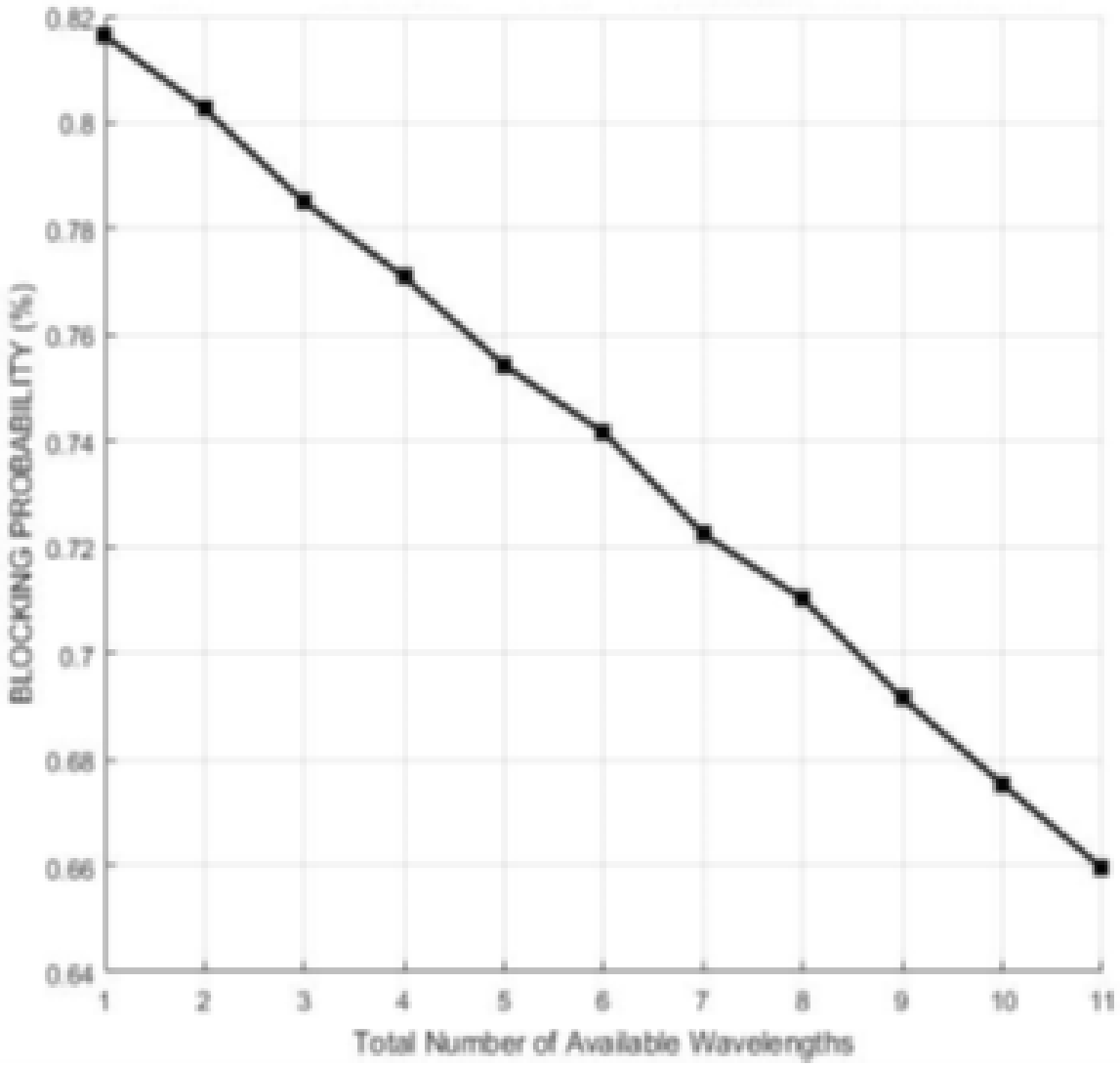


Figure: Variation of BP with number of wavelength

Figures

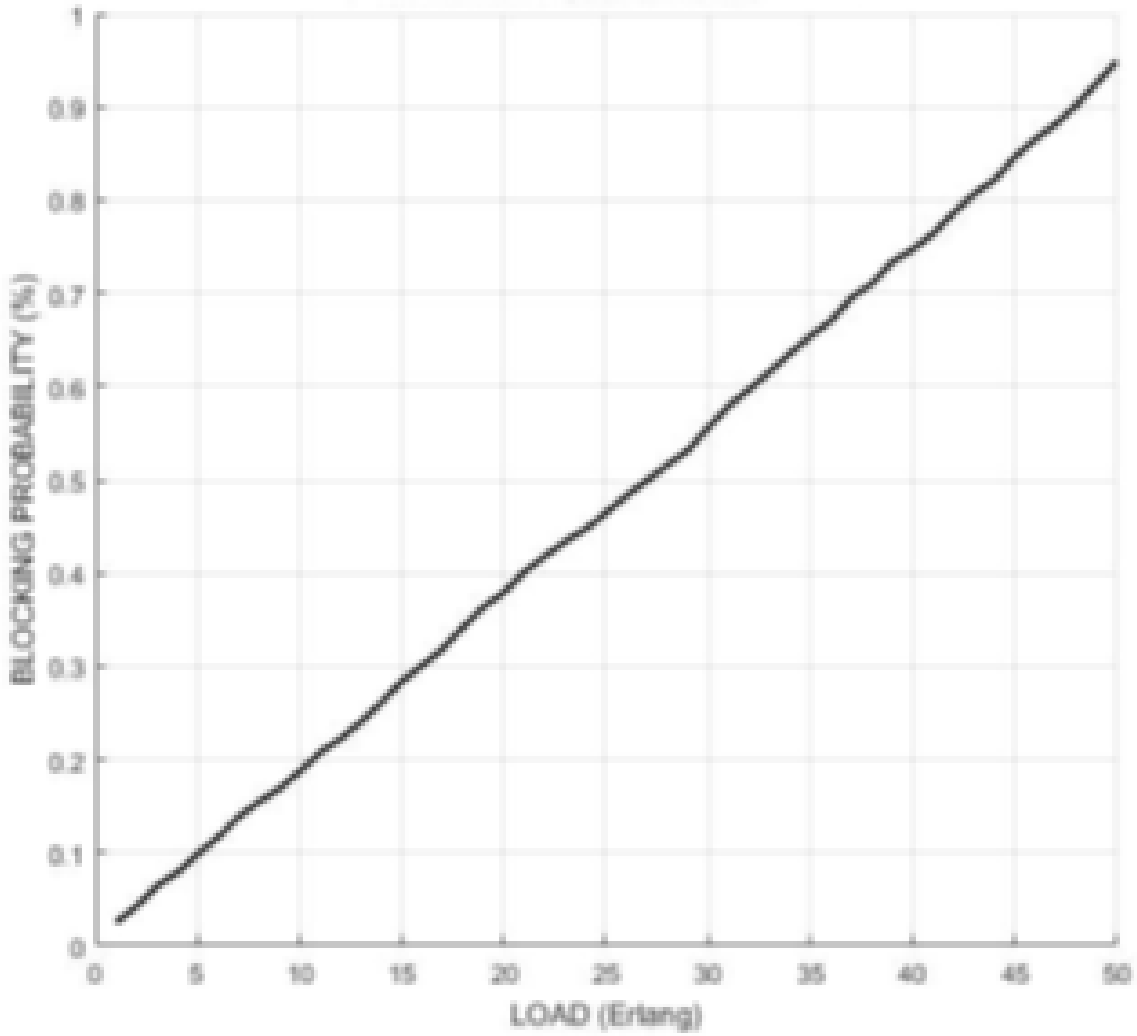


Figure: Variation of BP with traffic