

## **Analysis of Optical Ring Network Performance using ARTIFEX Tool**

Uzma Quadri<sup>1</sup>, Anam Quadri<sup>2</sup>, Ruhina Quazi<sup>3</sup>  
1,3(Department of Electronics & Telecommunication Engineering,  
Anjuman College of Engineering & Technology, Nagpur)  
2(Department of Electronics & Telecommunication Engineering,  
Nagpur Institute of Technology, Nagpur)

**ABSTRACT** : The paper presents an optical fiber ring structure consisting of interconnecting nodes connected using optical links. Optical ring networks communication are often confronted with the problem of routing and wavelength assignment. To eliminate the routing problem a simple unidirectional ring network with 8 nodes along with controller node is considered. The overview of ARTIFEX tool used to model the system has been presented. The simulation results have been shown in support of the work.

**Keywords** : ARTIFEX , object, place, transition.

### **I. INTRODUCTION**

In a Ring network, each machine or computer has a unique address that is useful for its identification purpose. The signal or information to be transmitted passes through each machine connected to the ring in a particular direction. Ring topologies usually utilize a token passing scheme, which is used to control the access to the network. Using this scheme, only one machine can transmit on the network at a time. The machines connected to the ring act as signal boosters (or repeaters) who strengthen the signals that transverse through the network.

Routing and wavelength assignment are serious issues to be resolved in optical ring. A simple unidirectional ring network with 8 nodes along with controller node is considered to overcome the routing problem. This will cause routing of all the calls in the clockwise direction. Thus the nodes are connected together with finite-speed unidirectional links. The controller node is used for the assignment of the wavelength.

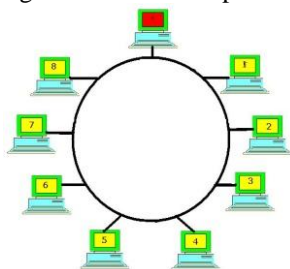


Figure 1. General Structure of Optical Ring Network

Each node requests at random intervals to the controller node the transmission resource (i.e. a wavelength path) in order to send a stream of data of random duration to a destination node which is different from itself. The Controller node exchanges messages with the other nodes in the network and handles the request messages to open a connection as well as to the end a particular connection. Each time a new request of connection arrives wavelength is allocated for the communication to take place.

Wavelength can be allocated using various algorithms like the Random algorithm, First Fit algorithm, Most Used algorithm, Wavelength conversion algorithm.

In the random algorithm the wavelength is selected across the ones available in a random fashion. [5] In the first fit algorithm the first wavelength available is selected by scanning a wavelength allocation table. Most Used algorithm looks for available wavelengths and the one that is most used over the ring is selected however the use of this algorithm leads to additional storage requirements.

Wavelength conversion algorithm assigns wavelength using wavelength converters. Thus the value of incoming and outgoing wavelengths can be controlled using wavelength converters that can be made available at respective nodes when Wavelength Conversion Algorithm is to be used.

Once the wavelength has been selected, the control node sends back to the requesting node an acknowledgment message containing the wavelength that the node should use to setup the connection. Absence

of wavelengths to be allocated for a connection represents a block condition for the system. This determines a very important parameter for the analysis of the network performance called the Blocking Probability. The Blocking Probability is dependent on the route length, free wavelengths and the number of available channels and is defined as the ratio of the requests rejected (calls blocked) to the total connection requests. The Blocking Probability is directly proportional to the length of route but inversely proportional to number of wavelengths.

**Artifex Tool Fundamentals**

ARTIFEX is an efficient general purpose simulation tool which has an outstanding graphical user interface that can be used to implement network topologies. The ARTIFEX software can be effectively used to model large or huge systems without losing control.

The ARTIFEX language is based on Extended Petri Nets which are a well-known formalism to describe discrete event systems.

Basically, a Petri Net can be defined in terms of three symbols:

A Place is used to model states, conditions and events. Places contain units of information called *tokens* whose standard type is NUL. Each place can contain several tokens at a time, but they must all be of the same type. Places are fundamentally queues of tokens. The queues may be FIFO (first in-first out), LIFO (last in-first out) and Token delay.

A Transition is the processing units that is used to model state transitions and activities. They act on the places to which they are connected. The necessary condition for a transition to *fire* is that each of its input places must contain at least one token. Whenever a transition fires, it removes one token from every input place and adds one token to each of its output places. [1][2]

**Graphical Representation**

A place is represented graphically by a circle, which is labeled with its name followed by the type of tokens to be contained in it as shown in Fig. 2(a).

A transition is represented graphically by a rectangle (vertical or horizontal) which is labeled with its name and, optionally, the priority number (default is 0) Fig. 2(b).

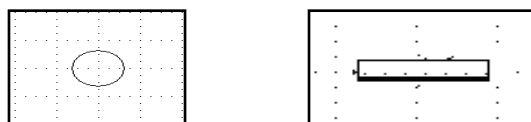


Figure 2 .Graphical repres

A transition can be defined using:

**Predicate:** It is an optional Boolean condition that can be associated with a transition. If the predicate is true then the transition fires.

**Priority:** It is an optional non-negative integer that can be associated with a transition so as to give it precedence over all transitions with a lower priority with a default value 0.

**Action:** It is an optional block of C/C++ statements that can be associated with the transition.

**Firing Delay:** It is defined as the delay from the instant at which the transition is enabled to the instant when it fetches the tokens from its input places.

**Release Delay:** It is defined as the delay from the instant at which the transition fetches the tokens from its input places to the instant when it releases the tokens in its output places.

Arc is used to define cause-effect relationships by connecting places and transitions.

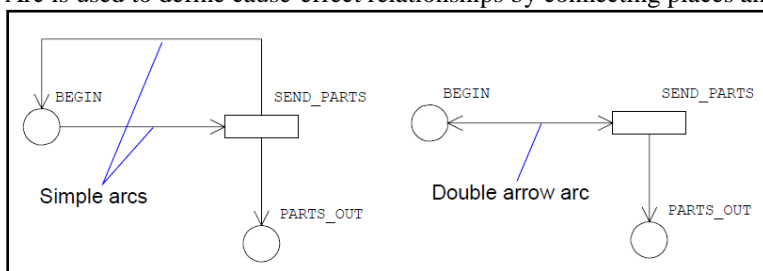


Figure 3. Graphical representation of an Arc

The other elements that can be used to model a ring network are as follows:

**Object:** The interface and behavior of a model is defined using Objects. Objects interact by exchanging tokens. These tokens flow from output places of one object to input places of another object through links.

ARTIFEX also includes large number of library models that makes design of complex networks easier. It allows simulation in real time simulating environment making use of the Time Advancement Mechanism.

## II. SYSTEM DESIGN

The model of dynamic ring network using ARTIFEX is as shown in Fig.4, it consists of 8 objects with same class "NODE" and a control node.

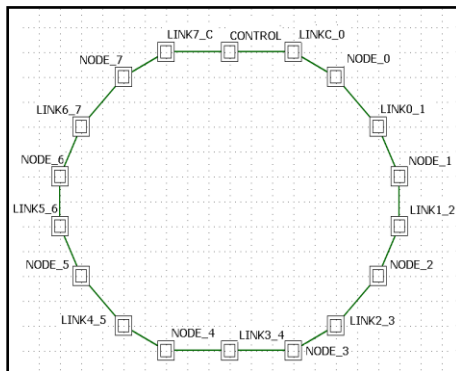


Figure 4. Main Page of Optical Ring Structure

Each of these objects represents an individual node in the network which can be initialized as sender. Each of the nodes is connected to their neighboring node in a clockwise direction using a linkset tool.

There are 8 more objects in between the nodes that represent links. Length of the link is expressed in kilometers. In order to calculate the delay introduced by the fiber, the light speed is assumed to be 200,000 km/s.

In this model few assumptions were made:

The length) of the control packets can be neglected respecting the speed of the links.

If no wavelength is available for a particular node request, the request is simply dropped.

Control node is used for the wavelength allocation. Opening the object representing control node gives the internal model which comprises of 3 transitions meant for handling, discarding and closing of setup requests.

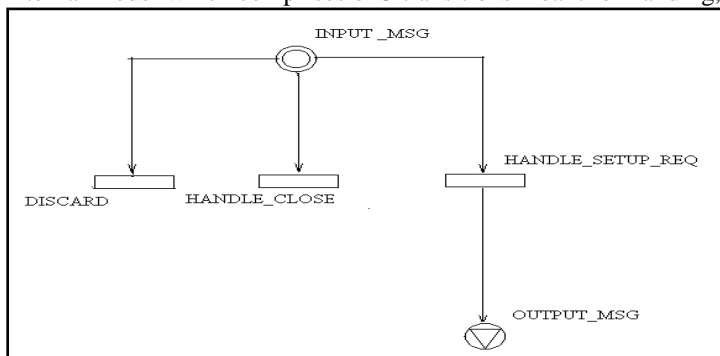


Figure 5. Control Node Model

The LINK\_T class implements a basic link. Any incoming token is delayed by an amount of time calculated by taking into account the length parameter and the light speed on the link itself.



Figure 6. Link Model

A node generates connection requests according to a Poisson distribution. In order to setup the connection it sends a SETUP\_MSG to the control node. This message contains the following information:

The message identifier that identifies this message uniquely.

The sender\_id that identifies the node (itself) requesting to setup the connection.

The destination\_id that identifies the node that the sender\_id node would like to connect to.

The controller, upon receiving a setup message replies with a SETUP\_ACK\_MSG. This message contains the following valid information:

message\_id that is the identifier of the SETUP\_MSG that the SETUP\_ACK\_MSG is replying.

wl that is the assigned wavelength (-1 if no wavelength are currently available for the connection).

destination\_id that is the id of the node that is the destination for this acknowledge message.

When the node would like to close the connection it sends a CLOSE\_MSG to the controller node. Any connection is completely identified by the 3 values: wl, sender\_id, destination\_id.

### III. VALIDATION OF MODEL WITH ARTIFEX

Once a model is created, it can be validated by means of a simulator to verify its dynamic behavior. The model is first updated then it is required to build the simulator i.e. create the *context file* with extension "v1". Then simulator is run through the Artifex.Validate tool. The context files are simple sets of data needed to define the compilation and linking parameters, which get involved by the *build* command.

The Artifex.Validate is the tool used to verify the behavior of the designed models. With this tool we can control and debug the model using step-by-step simulation, setting the breakpoints and examining the contents of tokens and other object data.

On running the simulator with the default settings of ARTIFEX, the color of the objects change indicating the different nodes, links and the controller as shown.

Once the simulation comes to an end, the states of the objects change indicating the number of requests generated or messages handled. The blocking probability of each node can now be measured and plotted. The measurements can be performed using the Artifex.Measure tool.

The simulation can be run with another powerful ARTIFEX tool - Artifex.Batch Run tool .The graphs obtained can also be superimposed to enable a comparative study. Fig.7 (b) above shows the superposition of the node blocking probabilities of node1 and node 0.

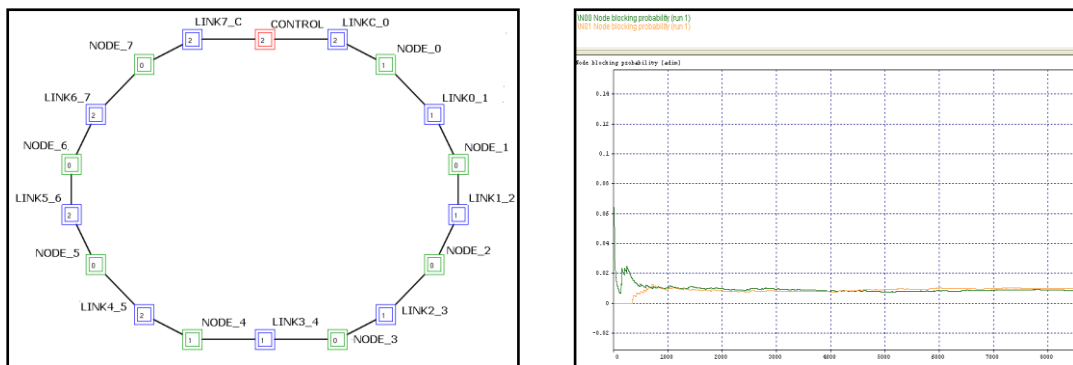


Fig1

(a)

(b)

### IV. CONCLUSION

An optical ring network has been designed using ARTIFEX tool. The work highlights the features of the ARTIFEX software and its use as a simulation and validation tool thus making it suitable for discrete event systems modeling where timing is also taken into account.

### REFERENCES

- Artifex 4.4 Getting Started, ARTIS Software Corporation.
- Artifex 4.4 User Manual, ARTIS Software Corporation.
- Artifex 4.4 Reference Manual, ARTIS Software Corporation.
- 0.Gerstel, R. Ramaswami, and G. Sasaki, "Benefits of Limited Wavelength Conversion in WDM Ring Networks"(1997)
- 0.Gerstel and S. Kutten, "A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters"(2003)