wjert, 2019, Vol. 5, Issue 1, 74-96. *Original Article* **ISSN 2454-695X**

World Journal of Engineering Research and Technology
WJERT
www.wjert.org **SJIF Impact Factor: 5.218**

THROUGHPUT OPTIMIZATION FOR A LAMBDA GRID NETWORK USING ADAPTIVE RESOURCE SCHEDULING TECHNIQUE

ILO Somtoochukwu F.*, Prof. H. Inyama and Dr. K. Akpado

Electronic and Computer Engineering Department, Nnamdi Azikiwe University Awka, Anambra State, Nigeria.

Article Received on 01/11/2018 **Article Revised on 22/11/2018** **Article Accepted on 13/12/2018**

***Corresponding Author**
ILO Somtoochukwu F.
Electronic and Computer Engineering Department, Nnamdi Azikiwe University Awka, Anambra State, Nigeria.

ABSTRACT

A grid network that employs optical wavelength division multiplexing (WDM) technology and optical switches to interconnect computing resources with dynamically provisioned multi-gigabit rate bandwidth light-path is called a Lambda Grid network. Data-intensive Grid applications require huge data transfers between grid computing nodes. Many data-intensive e-science applications like electronic very long Baseline Interfemetry (e-VLB[3] and Genomes – to – life (GTL) require aggregating several hundred Gigabytes of data files from distributed databases to computing resources (such as super computers) frequently in real time. Since data is aggregated at the time of computation, the time required to transfer the data over the network may be the main computational bottleneck. The problem of reserving bandwidth in a lambda grid has been studied extensively in the literature resulting in the proposition of a number of resource scheduling algorithm. Unfortunately, none of the existing lambda grid scheduling algorithms dynamically readjusts the scheduler to accommodate the actual amount of time that is required to transfer a file. This dissertation is focused on the investigation of an adaptive resource scheduling techniques to minimize the delay in the data aggregation task required by the computational and data intensive e-science application running on lambda grid network. Simulation was carried out using the digital model of the 24 node National Lambda Rail (NLR) lambda grid network topology created with Cisco packet tracer 7.0. Results obtained showed that the proposed algorithm achieved 14% and 30% Average Finish Time improvement over the VBLS and ViFi algorithms respectively. The proposed technique achieved a substantial reduction in blocking probability. It achieved 15.4% and 23% improvement in blocking probability over the VBLS and ViFi algorithms respectively. Results obtained for the effect of connection duration on blocking probability showed improvement of 7.5% and 18% improvement over the VBLS and VIFi algorithm respectively. At 7.55%, the proposed algorithm showed a very low job blocking rate and indicates a 16% and 29% improvements compared to the VBLS and ViFi algorithms respectively. Reduction in the variation of the effectiveness of the algorithm with job size was found. It achieved 18% and 21% improvement over the VBSL and ViFi algorithms respectively. Simulation results also indicated that the proposed algorithm gives a substantially low reservation delay as per impact of request of lambda arrival rate.

KEYWORDS: Lambda, Grid, light path, Optical circuits, Scheduler, Varying bandwidth Scheduler, Virtual finish heuristics, Cisco packet tracer, Algorithm, C++ program.

1. INTRODUCTION

Computational and data intensive e-Science and e-Collaboration applications involve special class of scientific services or instrument (located across various organizations) that are geographically distributed. These resources could be computational systems (such as super computers, clusters, or even powerful ultra-high end engineering workstations), special class of devices (such as remote sensors) and even storage systems. A number of data and computational intensive applications need more computing power than can be offered by a single resource in order to solve Problems within feasible/reasonable time and cost. The LAN/SWITCH connected clusters (of computers) platform has been employed to solve computationally intensive problems (Bunya .R, 1999) however, they alone cannot offer the computational power demanded by such applications. All these means that these geographically distributed resources need to be logically coupled together to make them work as a unified resource. This led to the popularization of a field called grid computing – i.e. grid computing is a computational technique to harness distributed resources as a unified process. Grids consist of the aggregation of numerous dispersed computational, storage and network resources, able to satisfy even the most demanding computing jobs (Pieter, .T et.al, 2007). Grids using optical transport networks are commonly referred to as Lambda Grids.

Many data-intensive e-science applications like electronic Very Long Baseline Interferometry (e-VLB (Wolfgang, S and Behrend, D, 2007) and Genomes – to – life (GTL) require aggregating several hundred Gigabytes of data files from distributed databases to computing resources (such as super computers) frequently in real time. Since data is aggregated at the time of computation, the time required to transfer the data over the network may be the main computational bottleneck. Even a single second of idle time, during which the data is being aggregated, may result in the loss of several teraflops of computational power. Therefore, minimizing the delay in data aggregation is the key to improving the overall system throughput (Shen, S. et.al., 2008)(Savera .T.W et.al., 2008). A reliable and dedicated infrastructure available on demand is a key resource for data intensive e-science application. Lambda Grid Networks, which are backbone networks supported on optical fiber technology can provide such an infrastructure – since it offers an end – to – end optical circuit (also called wavelength or lambda – an optical connection established over a certain wavelength) between two end points (Shen, S. et.al.,2008).

2.0 Summary of related Literature

(Liu, X. et.al., 2009), proposed an algorithm that uses Deadline constant for task scheduling and light path establishment in the lambda grid. This was reported to be effective in minimizing the lambda grid resource usage and improving file aggregation time. However in (Page, A.J, et.al., 2005), it was considered unviable since it is based on the assumption that edge node and computing node have unlimited buffer to store sets which cannot be realized in practice.

Work done by the author (Castillo, C. et.al., 2007), focused on the use of advanced reservation of resources using Best Fit strategy for lambda grid resource scheduling. Results obtained indicate a substantial improvement of grid resource utilization. However as reported in reference (Lakshmiraman, V. and Ramamurity, B. 2009), the job blocking rate based on this scheduling technique increases (though marginally) with every newer scheduling cycle.

In (Ho, P.H and Mouftah, H.T, 2003), the authors proposed a grid scheduling technique which is based on exchanging information about critical optical paths (i.e. links) in the network and avoiding those links during wavelength assignment. It was shown that this method reduces the blocking probability compared to a fixed wavelength assignment scheme. However the limitation of the scheduling method has to do with the overhead imposed by the exchange of network link-state information.

(Nageswara, S.V. et.al., 2004), Proposed scheduling algorithm that compute the quickest path with a minimum end-to-end delays to transfer file of a given size from its source to destination when bandwidth and delay constraints on the links are specified. Using similar assumption a Varying Bandwidth List Scheduling (VBLS) heuristic to compute circuit over a lambda grid was studied in (Veeraraghavan, M. et.al., 2003). (Buyya, R. (ed), 1999) (Pieter, T. et.al., 2008) showed that the basic problem with most of these grid scheduling algorithm is that they assume an ideal communication system where all the resource are fully connected and communication between two resource can be used whenever needed.

2.1 Lambda Grid Network Topology Model

Graph representations of the networks topology of the lambda grid abound in the literature. A lambda-grid networks topology, an example of which is the VSN (www.net.gov/ultranet, 2007), may be represented as a graph G (V, E), where each node V represents a core switch, and the edge 'E' represents the connectivity between core switches. Core switches are connected with single or multiple lambdas (a lambda is an optical connection established over a certain wavelength). A core switch is attached to a Multi Service Provisioning Platform (MSPP).

MSPPs provide a Synchronous Transport Module (STM)/Synchronized Digital Hierarchy (SDH) and Ethernet channels at sub-lambda granularities to end devices such as Storage Area Networks (SANs), data warehouses, or host computers. Thus, a lambda may provide an end-to-end connection between two end-to-end machines via the MSPPs and core switches (Nageswara, S.V, et. al., 2004). The connection from the core switch to the MSPP to the end lost is not represented in graph 'G'.

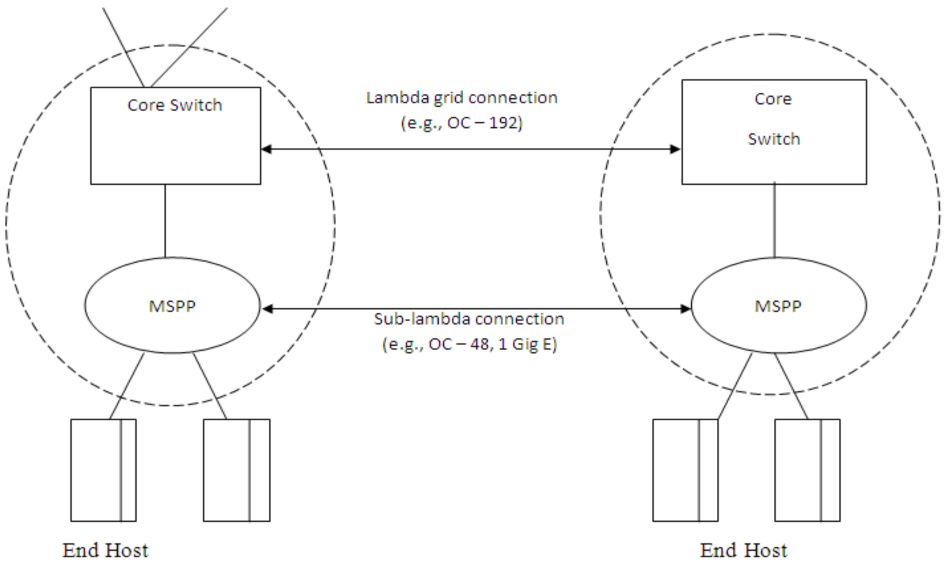

Fig. 2.1: Illustration of a connection between two end host using a Lambda Grid.

The layout of the end-to-end connectivity is shown in figure.2.1 for example; a simple way by which an end host may connect to a lambda grid is by using a Gigabit Ethernet Interface Card over a Local Area Network (LAN) connected to the MSPP. Alternatively, it may be connected via a 2.5 Gbps (OC-48) STM connection. This connection from the MSPP to the end host is termed as a sub-lambda connection. It is suggested in (Banerjee, A. et.al., 2008) that in order to simplify the problem setting, that the assumption should be made that all end hosts are connected to the MSPPs with the same connection bandwidth (that is 1, or 2.5 Gbps), and therefore, the granularity of each sub-lambda connection is the same.

2.2 Modeling Grid Scheduling Problem

Various literatures on file aggregation on lambda grid, one way or the other attempt at describing the problem of resource scheduling for the Lambda grid. Two things evident in these writings are the representation of lambda grid network topology as graphs and the formulation of the problem of aggregating large data files from distributed data bases in a lambda grid network as a time-path scheduling problem (TPSP). Furthermore, available research reports (due to somewhat shot comings in TPSP) describe the modification of TPSP, calling the new problem N-destination TPSP (NDTPSP).

However, (Coffman, E.G, et.al., 1985) (Taesombut, N. et.al., 2006) (Banerjee, A. et.al., 2008] hold that both TPSP and NDTPSP (being NP-complete problems) cannot be solved in polynomial time necessitating the need for the proposition of heuristics for large scale file transmission tasks in a lambda grid.

III. MATERIALS AND METHODS

We describe our methodology that is used in achieving the work, the reason for the particular method to be used and design of the adaptive lambda grid resource scheduling algorithm. The lambda grid scheduling problem is formulated as a time-path scheduling problem (TPSP). The scheduler is composed of three main algorithms. Algorithm 1 is run to allocate lambdas; the function of algorithm 2 is to determine the file transmission path in the lambda grid. Algorithm 3 implements the lambda grid resource scheduler. The scheduler is the module that schedules the actual file transfer. Every run of the path determination algorithm (Algorithm 2) is integrated with the running of the lambda allocation algorithm i.e (Algorithm 1).

The scheduler iteratively runs the path algorithm to dynamically re-establish the shortest path from the source to the destination (with the consequent re-allocation of wavelengths). This reallocation of wavelengths ensures that idle lambdas can be re-provisioned for ongoing or later file transfer task in the lambda grid.

A. Description of Materials/Tools Used

- Cisco Packet tracer 7.1
- C++ Source Code
- CORBA Protocol
- System Model: Hp Elite book 8440P, Intel® core™ i7 cpu, Memory 6144MB RAM, Page file 5862.
- Operating System: Windows 7 Ultimate 64 bits (6.1,Build 7600).

B. Choice of Methodology

Graph theory is adopted for the systems analysis presented. The system fundamentally being a network scheduling and routing algorithm makes graph theoretic method most suitable for its analysis. Graph theory is best suited to the analysis and development of network routing algorithm furthermore specifically the formulation of the TPSP problem is basically based on graph algorithm.

C. Data / Information Gathering

The Lambda grid being the next generation of scientific computing platform is mainly found at government science project sites around the world – the developed economies. Technical data on the operations and problems of the lambda find is best obtained from report and review documentation. Hence the data gathering technique for this work entails review of documentation on the HS department of Energy's Ultra-science Net (DOE: Ultra Science Net Test bed. www.csm.ornl.gov/ultranet/overview.pdf), the National Lambda Rail (NLR) networks (Doug, H., 2003), and the Teragrid now replaced with Extreme Science of Engineering Digital Environment (XSEDE) (Travostino, F., et.al., 2006). The data obtained from the lambda grid project document are:

- The file transfer profiles at each transmitting node to super computers.
- File sizes and associated transfer times in the gird.
- Mean transfer time
- Link utilization and link capacity (OC – 192 $\longrightarrow$ 10Gbps, each sub lambda OC–48 $\longrightarrow$ 2.5Gbps
- Sample lambdas gird network topologies.

A. Data Analysis

The analysis of the lambda grid data is done to find parameters for accommodating the variance in file transfer times. (in order to predict optimal circuit holding time). To do this different number of standard deviation (σ) away the mean (m) (i.e. mean of past data transfers), which would correspond to the upper limit of a confidence interval in a normal distribution is computed.

Analysis included estimating the following parameters.

- **Targ:** average offline schedule finish time
- **Tmax:** Maximum observed actual finish time in N transfers
- **Nmax:** Number of incomplete files transfers when Tmax was measured.
- **Tmin**: Minimum observed actual finish time in N transfers.
- **Nmin**: Number of incomplete files transfer when Tmin was measured.
- **Job Blocking Rate:** Percentage of jobs blocked divided by the total number of job submitted.
- **Fairness:** Metric that shows performance of the heuristic for smaller and larger jobs.
- **Effectiveness:** It is calculated as the percentage of latest finish time of the job scheduled and the blocking rate to the maximum time slot S.

The lambda grid scheduling problem is formulated as a time-path scheduling problem (TPSP). The scheduler is composed of three main algorithms. Algorithm 1 is run to allocate lambdas; the function of algorithm 2 is to determine the file transmission path in the lambda grid. Algorithm 3 implements the lambda grid resource scheduler. The scheduler is the module that schedules the actual file transfer. Every run of the path determination algorithm (Algorithm 2) is integrated with the running of the lambda allocation algorithm i.e (Algorithm 1).

IV. RESULT AND ANALYSIS

The foundation of NLR is the dense wave division multiplexing (DWDM) based national optical footprint using Cisco systems 15808 optical electronic system with a capacity of 40channel (wavelength) per fiber pair. Each wavelength can support transmission at 10 billion bits per second! (10Gbps).

The single line diagram of the NLR 24-node lambda grid sample network topology used in the simulation carried out in this chapter is given in figure 4.1.

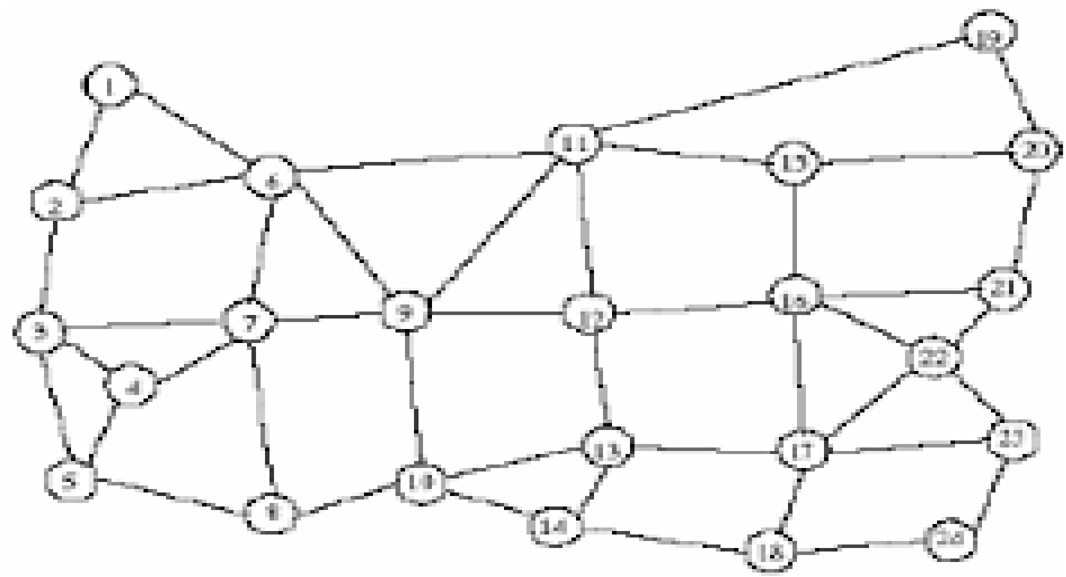

Figure 4.1: Single line diagram of the NLR 24-node lambda grid sample network topology used in the simulation.

The Cisco packet tracer 7.1 (Wang Xia-hong., 2012). is used to create the digital model of the 24-node lambda grid network topology. This is shown in figure 4.2.

The nodes in figure 4.1 are multilayer switches. These nodes perform layers 2, 3and other upper layer networks functions. The lines between nodes are 0C-192, while that between node and super computers are 0C-48 specification.

The Cisco network simulator has application programming interface (API) support for the C++, java and C programming languages. For this work the C++ programming language was used. The proposed lambda grid scheduler is coded in the C++ language. It interfaces and communicates with the kernel of the Cisco packet tracer network simulator using inter procedural communication based on the CORBA protocol. C++ code is used to automate the scripting of file transfer and file aggregation jobs between nodes in the network during the simulation run.

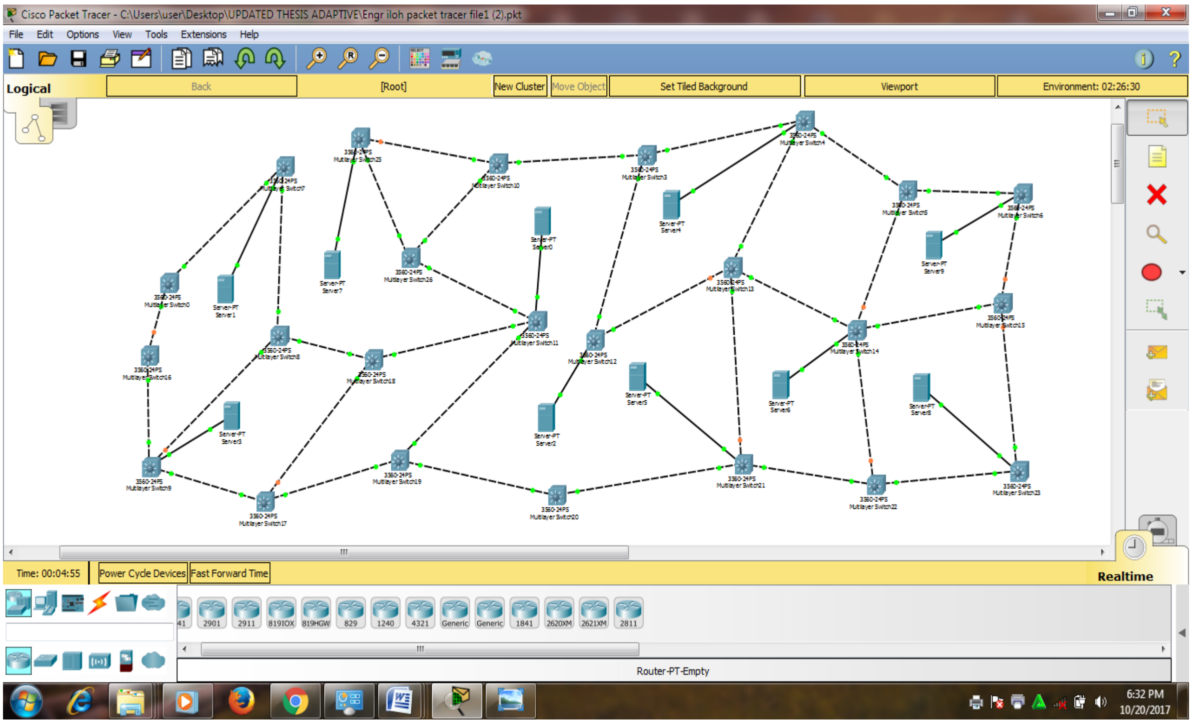

Figure 4.2: Model of the 24-Node lambda Grid network topology for evaluating the performance of the proposed adaptive lambda Grid Scheduling Algorithm.

4.2 File Transfer Time

In the simulation, the program code increases the number of files gradually from 50 to 300. For each setting, the program measures the file transfer finish time. All link capacities are C= O C-192 (10 Gigabytes).

Figure 4.3 compares the performance of the proposed adaptive lambda gird scheduling algorithm; the Varying Bandwidth List Scheduler (VBLS) and the Virtual Finish time (ViFi) grid scheduler. It can be observed from figure4.3 that the proposed adaptive scheduler performs better than the VBLS and the ViFi algorithms. The VBLS algorithm performs better than the ViFi algorithm. Furthermore, it can be noticed that the different in performance gets even more distinct as the number of files increases. In other words, with increasing number of files, the performance of the proposed adaptive algorithm increases and the margin with which it outperforms the other two algorithms widens. This is to be expected: that as the number of data to be transferred between nodes increases, the algorithm adapts by rerouting data and **re-provisioning** idle lambdas (i.e wavelengths) in order to **effectively service the additional work loads**. This gives the proposed algorithms an edge over the other two algorithms. The finish time for transferring same amount of data **is less** for the proposed adaptive scheduler compared to the other two schedulers.

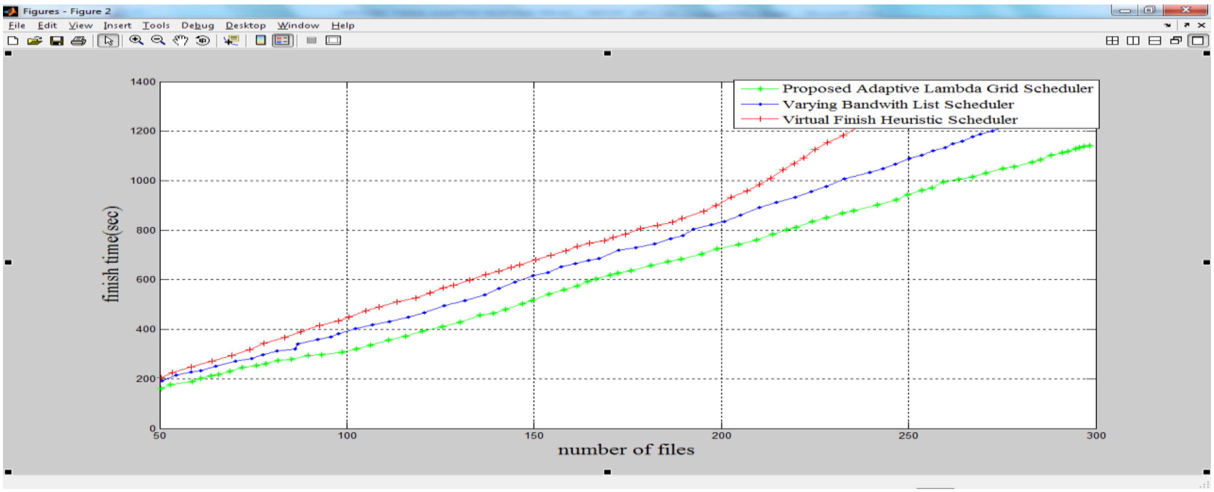

Figure 4.3: Comparison of the File Transfer Finish Time of the Proposed Adaptive Lambda Grid Scheduler, the Varying Bandwidth List Scheduler and the Virtual Finish Scheduler.

To compute and compare averages for the three schedulers, ten readings are taken off the graph of figure 4.3. The readings are taken at intervals of 25 along the horizontal axis (number of files) starting from 50. The outcome is tabulated in table 4.2.

Table 4.2: File Size versus Finish Time for File Aggregation using the Proposed Adaptive Schedule, the Various Bandwidth List Schedule and the Virtual Finish Heuristic Scheduling Algorithms.

Number of files	Finish time (sec.).		
	Proposed Adaptive Gird Schedule Algorithm	VBLS algorithm	ViFi algorithm
50	161.80	189.40	199.50
75	249.60	284.80	322.40
100	307.30	390.10	447.80
125	397.70	485.50	555.70
150	513.10	615.50	678.70
175	628.50	718.80	922.00
200	723.80	829.20	1125.30
225	836.00	944.60	1300.90
250	944.60	1077.60	1353.60
275	1037.50	1208.10	1376.30

From the tabulation, the average finish time for the aggregation of the files randomly distributed between 5 Gigabytes and 20 Gigabytes is computed.

From the tabulation, the average finish time for the file aggregation using the proposed adaptive lambda gird scheduling algorithm, is the VBLS scheduling algorithm and the Virtual

Finish (ViFi) algorithm are **579.99** seconds, **674.36** seconds and **828.22** seconds respectively. With this average, the proposed adaptive lambda gird scheduler provides approximately a **14%** and **30%** improvements in grid file aggregation finish over the VBLS algorithm and the virtual finish algorithm respectively.

4.3 Evaluation of Effect of Late Arrival Rate on Blocking Probability

The rate of arrival of request for allocation of optical wavelengths (lambdas) has impact on the job blocking probability (i.e. the probability of not scheduling a request within its window). As this is an online scheduling problem, the request arrive dynamically and for each request, the scheduling algorithms must compute a path (routing) and then check if a wavelength (i.e. a lambda) on each link of this path can be reserved for a duration within the scheduling window (the scheduling window is specified by the schedule **start time** and **end time**). The scheduling algorithm allocates a lambda on each link along a path from the source to the destination nodes. If a lambda along the path for the specified period of time is not available, another path has to be determined.

The objective of the algorithm is **to determine the schedule** to route each incoming light-path connection request dynamically while **minimizing the probability** that a connection request will be refused due to lack of available light path and maximizing the overall network throughout. Figure 5.4 shows the effect of the arrival blocking probability.

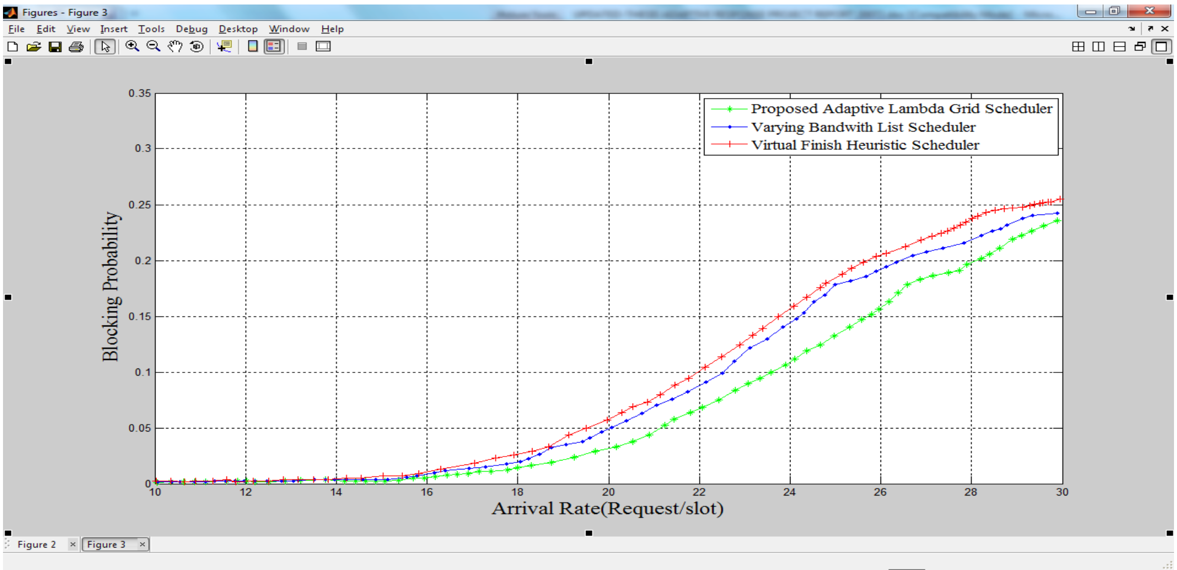

Figure 4.4: Effect of lambda grid request arrival rate on blocking probability.

The general observation from the graph is that blocking probability increases with increase in request arrival rate. It seems to increase exponentially. It can be observed for the three algorithms that the probability of not scheduling a lambda request within its window is very infinitesimal (almost zero in this case) for arrival rate below 14 request/slot. With low values of arrival rate the blocking probabilities of the three algorithms seemed almost equal. However, with increasing arrival rates the difference in the performances of the algorithms in terms of blocking probabilities begin to significantly stand out.

Even more prominent is the significant difference in the blocking probabilities of the proposed adaptive scheduler with those of the VBLS and ViFi. This means that, at increasing arrival rate, for the algorithm to meet its objective of minimizing blocking probabilities it has to bring in adaptability.

To compute and compare averages of blocking probabilities of the three algorithms, readings are taken off the graph of figure 4.4 at 10, 16, 22 and 28 along the horizontal axis. The outcome is given in table 4.3.

Table 4.3: Tabulation of Request/Slot for Allocation of Optical Wavelength against the Blocking Probabilities of the Proposed Adaptive Scheduling Algorithm, the VBLS Algorithm and the ViFi Algorithm.

Request/slot	Blocking probability		
	Proposed Adaptive Grid Scheduling Algorithm	**VBLS Algorithm**	**ViFi Algorithm**
10	0.0007	0.0007	0.0012
16	0.0047	0.0078	0.0100
22	0.0658	0.0886	0.1007
28	0.1958	0.2185	0.2355

Based on table 4.3, the average blocking probabilities for the proposed scheduler, the VBLS scheduler and the ViFi scheduler are 0.0667, 0.0789 and 0.0866 respectively. It is clear the proposed adaptive scheduler was the lowest blocking probability. This value indicates a 15.4% and 23% improvement over the VBLS algorithm and the ViFi algorithm respectively.

4.4 Evaluation of Effect of Connection Functions on Blocking Probability.

Figure 4.5 gives the effect of the connection duration **d** on the blocking probability. It can be seen that the blocking probability increases as connection duration increases. It can also be observed that the blocking probability significantly increases for connection duration >5 for the considered topology. As can be observed the proposed adaptive scheduler gives the best performance of the three, followed by the VIBLS algorithm.

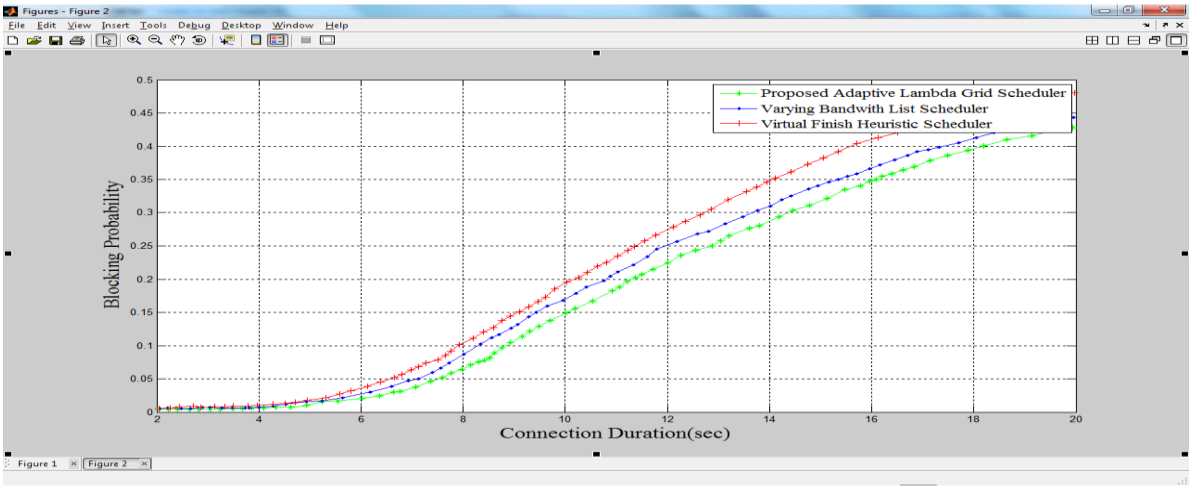

Figure 4.5: Connection Duration verses Blocking Probability.

Table 4.4 is setup by taking readings at location 6, 10, 14 and 18 along the horizontal axis in figure 4.5.

Table 4.4: Comparison of the Effect of Connection Duration on Blocking Probability on File Aggregation for the Proposed Adaptive Scheduler, the Virtual Bandwidth List Scheduler and the Virtual Finish Heuristic Scheduling Algorithms.

Connection duration (sec)	**Blocking probability**		
	Proposed Adaptive Gird Scheduling Algorithm	**VBLS Algorithm**	**ViFi Algorithm**
6	0.0174	0.0228	0.0371
10	0.1469	0.1710	0.1960
4	0.2879	0.3121	0.3478
18	0.3969	0.4121	0.4549

Based on table 4.4, the average blocking probabilities as a result of the impact of connection duration are **0.2122, 0.2295** and **0.2589** for the proposed adaptive scheduler, the VBLS and the ViFi algorithms respectively. These figures indicate that the proposed algorithm has the lowest blocking probability. These values represent a **7.5%** and **18%** improvement over the VBLS algorithm and the ViFi algorithm respectively.

4.5 Evaluation of light-path Reservation Delay as a Function of Wavelength Request Arrival Rate.

Figure 4.6 shows the reservation delay, i.e. the time elapsed from the requested start time s to the time equal to the start time plus the time slot, as a function of lambda request arrival rate for the proposed adaptive algorithm, the VBLS algorithm and the ViFi algorithms.

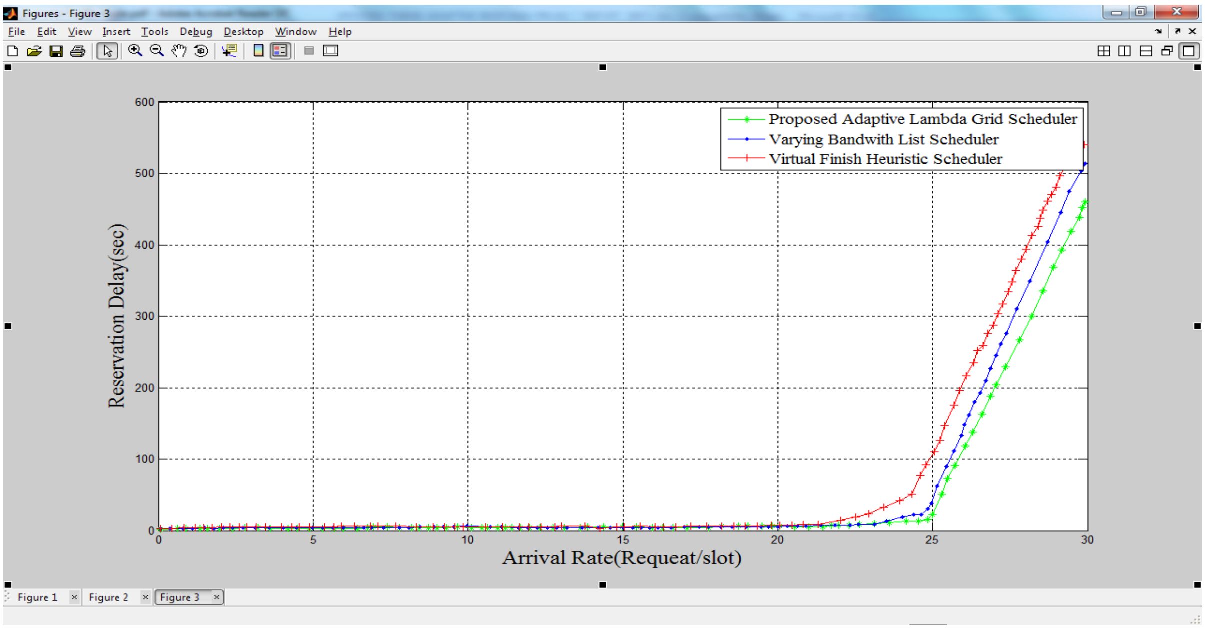

Figure 4.6: Reservation Delay as a function of wavelength Request arrival Rate.

It can be observed that for arrival rates below the threshold point of 20, the reservation delay algorithms are almost equal. Then around 25 request/slot of the reservation delay for each of the algorithms rises rapidly. It can be seen that based on the reservation, the proposed algorithms stands out from the other two algorithms. In comparison, it gives the least reservation delay as per impact of request for lambda arrival rate. This means that the proposed adaptive algorithm always tries to schedule close to the start time of the scheduling window as possible.

4.6 Evaluation of Job Blocking Rate

The numbers of jobs simulated are varied up to 200. The number of tasks per job is varied in the program code up to 6. The job size was determined in program code based on the number of jobs that were submitted to the scheduler. Figure 4.7 shows the job blocking rates of the three grid scheduling algorithms. The job blocking rate is the percentage of jobs blocked divided by the total number of jobs submitted.

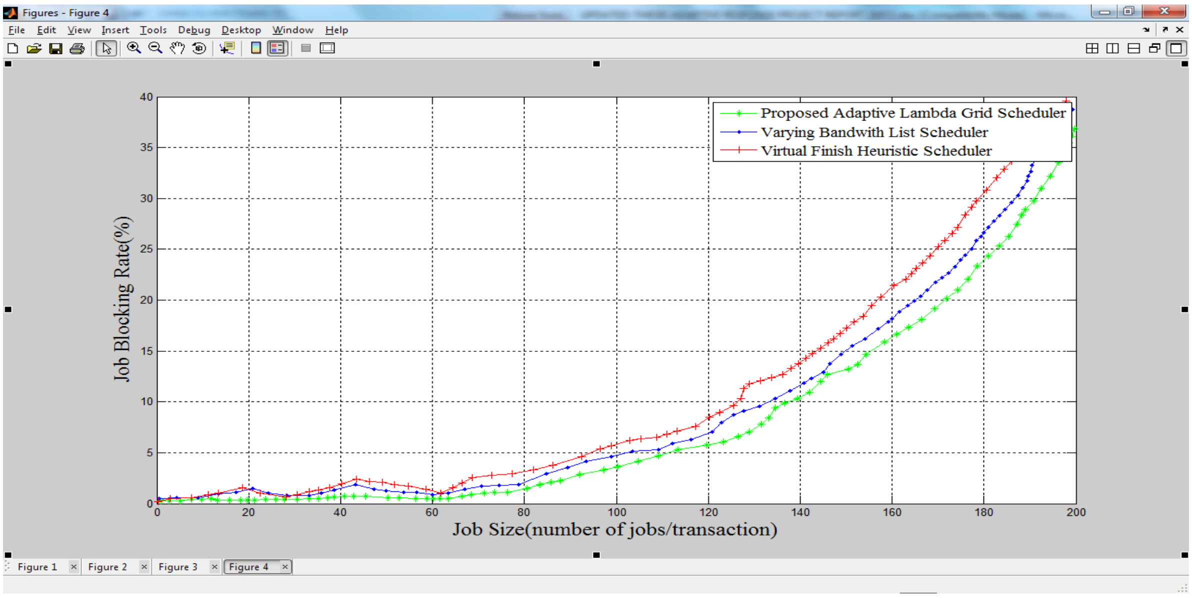

Figure 4.7: Average Job Blocking Rate.

As can be observed, the proposed adaptive scheduler clearly out performs the other two algorithms. As can be observed, the job rates using the VBLS and ViFi algorithms are more than that using the proposed algorithm. That is, the proposed algorithm has minimal blocking rate compared to the other two algorithms. It is also evident from figure 4.7 that the blocking rate of all three algorithms increases dramatically with the increase in job size. However, from the result; it is evident that the proposed algorithm reduces the blocking rate in comparison. Furthermore, it can be observed from figure 4.7 that the job blocking rate using the proposed adaptive algorithm does not vary and oscillate as those of the VBLS and ViFi algorithms.

To estimate the average for job blocking rate, readings are taken off the graph at 20, 60, 100, 140 and 180 positions along the horizontal axis. Table 4.5 gives the readings for the three algorithms.

Table 4.5: Job Blocking Rate versus Job Size.

Job size	Job blocking rate		
	Proposed adaptive grid scheduling algorithm	VBLS algorithm	ViFi algorithm
20	0.1786	0.9643	1.4643
60	0.2500	0.8214	1.3214
100	3. 6071	4.6786	5. 7500
140	10.2500	11.8214	14.0357
180	23.3643	20.8214	30.6071

From the tabulation, the average job blocking rates are **7.53%, 9.02%,** and **10.64%** for the proposed adaptive algorithm, the VBLS algorithms and the ViFi algorithms respectively. The computed average for the proposed adaptive scheduling algorithm represents a **16%** and **29%** improvement over the VBLS and the ViFi algorithms respectively.

4.7 Evaluation of the Effectiveness of the Scheduling Algorithms

The effectiveness is calculated as the percentage of latest finish time of the job scheduled and the blocking rate to the maximum time slots. The higher the percentage, the more effective the algorithm.

The algorithm computes the effectiveness using equation (2.1).

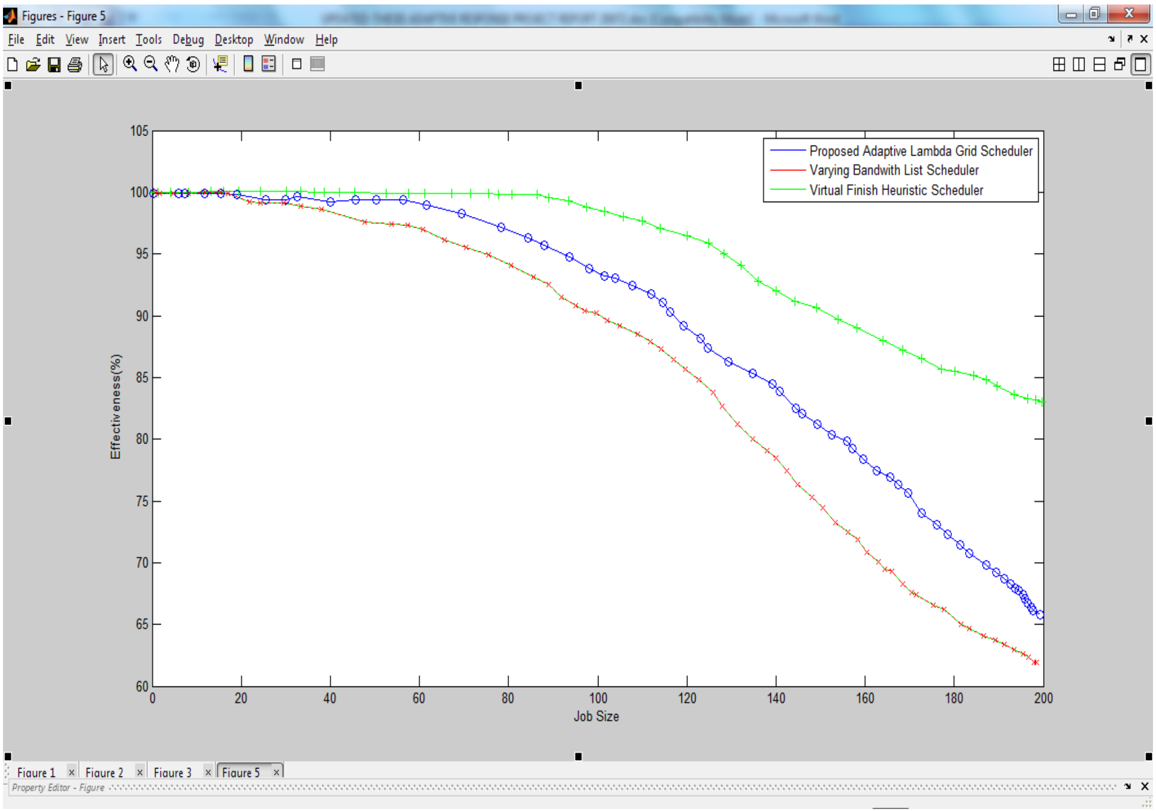

Figure 4.8: Shows the variation of effectiveness with job size for the three algorithms.

It can be observed from the graph that effectiveness reduces with increase in job size. The simulation result indicates that the proposed adaptive scheduling algorithm has the best effectiveness of the three algorithms. As can be seen the effectiveness of the algorithms reduced from almost **100%** to about **83%, 65%** and **61%** for the proposed adaptive algorithm, the VBLS and the ViFi algorithms respectively. It can be inferred from this that the proposed adaptive algorithm performs 18% better than the VBLS algorithm and 21% better than the VIFI heuristic grid scheduling algorithm respectively.

5.1 CONCLUSION

This work focused on the development of an adaptive resource scheduling technique to minimize the delay in the data aggregation task in a computational lambda grid network.

The problem of data aggregation delay in the lambda grid has enormous impact on the viability of certain e-science application that have critical timing requirements, the loss of teraflops of super computer computing power as a result of the scheduling related delay in the lambda grid has the impact of increase in research and development (R & D) cost, delays in proceeding with vital research (especially related to chronic disease research etc.). This problem not only leads to delays in vital break through for mankind but also, in some cases outright project cancellation and wasted investment. The lambda grid scheduling problem has far reaching impact on the accuracy and validity of e-science grid applications required in making vital forecast relating to natural disaster, evolving disease postures, climate change, technical issues relating to our ability to explore deep space etc.

Grid computing emerged as a means of coupling together numerous heterogeneous and geographically distributed computational and storage resource to make them work as a unified resource. By coupling numerous heterogeneous computational and storage resource distributed over various locations, Grids are able to satisfy the ever increasing demand of both processing and storage power, surpassing the capabilities of each of its individual resources. This allows a grid to accommodate even the largest and most resource- demanding applications. Grids making use of optical circuit switched transport network are usually denoted as lambda grids.

One is said to be pushing the current increase in the development of lambda grids networks to the resource intensive requirements of e-science application. Many of these data-intensive, e-science Gids applications like electronic very long Baseline interferometry (e-VLB) and Genomes to life (GTL) requires aggregating several hundred Gigabytes of data files from distributed databases (usually geographically separated) to computing resource (such as supercomputers) frequently in real time since data is aggregated at the time of computation, the time required to transfer the data over the network is the main computational bottleneck.

In this work the lambda grid scheduling problem is formulated as a time-path scheduling problem. The design carried out constructed the scheduler as a three algorithm system. Algorithm 1 is designed to allocate lambdas, the formation of algorithm 2 is to determine the file transmission path in the lambda grid, and Algorithm 3 implements the lambda grid resource scheduler. The scheduler is the module that schedules the actual file transfer. Every run of the path determination algorithm (Algorithm 2) is integrated with the running of the lambda allocation algorithm (i.e Algorithm 1). The scheduler iteratively runs the path algorithm to dynamically re-establish the shortest path from the source to the destination (with the consequent re-allocation of wavelength). This re-allocation of wavelength insures that idle lambdas can be re-provisioned for ongoing or later file transfer in the lambda grid. The proposed scheduler design is coded in the C++ programming language.

In the work, simulation was setup and carried out to test and evaluate the proposed lambda grid scheduling algorithm, for the required data, the 24-node National lambda Rail (NLR) lambda grid topology was used. Cisco packet tracer for network modeling software was used to create the digital model of the 24-node lambda grid network topology. The Cisco network simulator with application programming interface (API) support for the C++, java and C programming languages.

In the simulation, the program code increases the number of file gradually from 50 to 30. For each setting, the program measures the file transfer finish time. All link capacity are OC-192 (10 Gigabytes). Source and destination node where the supercomputer exists are automatically selected by the program. A specified number of file size are randomly distributed between 5Gigabytes and 20Gigabytes and are located randomly across and the remaining node in the lambda grid network.

The performance of the proposed adaptive lambda grid scheduling algorithm was analogically compared with the varying Bandwidth List scheduler (VBLS) and the virtual finish (ViFi) grid scheduler. The observation is that the proposed adaptive scheduler performs better than the VBLS and the ViFi algorithms. The VBLS algorithm performs better than the ViFi algorithm. The difference in performance gets more destruct as the number of file was increased as during the simulation carried out. With increasing number of file, the performance of the proposed adaptive algorithm increased and the margin with which it outperforms the other two algorithms widened. This is to be expected, that as the amount of data to be transfer between nodes in the lambda grid increases, the proposed algorithm adopts by searching data and re-provisioning idle lambdas (i.e. wavelength) in other to effectively service the additional work loads. This gives the proposed algorithm an edge over the other two algorithms.

The finish time for transferring same amount of data is less for the proposed adaptive scheduler compared to the two schedulers.

Result obtained by the evaluation of the average finish time for file aggregation showed that the proposed algorithm achieved 14% and 30% improvement over the VBLS algorithm and the ViFi algorithm respectively.

The rate of arrival on blocking probability (ie the probability of not scheduling a request within the window) was evaluated. Comparative analysis carried out show that the proposed scheduler has the lowest blocking probability. The blocking probability of the proposed algorithm, the VBLS algorithm and the ViFi algorithm are 0.0667, 0.0789 and 0.0866 respectively. This value showed that the proposed algorithm achieved a 15.4% and 22.9% improvement in blocking probability over the VBLS and ViFi algorithm respectively. It was observed that the blocking probability increases with increase in request arrival rate. The increase seemed to be exponential. With low values of arrival rate the blocking probability of the algorithm seemed almost equal. However, with increasing arrival rates the difference in the performances of the algorithm in terms of blocking probability begins to significantly stand out.

The effect of connection direction on blocking probability was evaluated. Result obtained indicates that blocking probability increases as connection duration increases. The proposed algorithm has the least blocking probability with increase in connection duration. Numerical results show that the proposed algorithm achieved 7.5% and 18% improvement over the VBLS algorithm and the ViFi algorithm respectively.

Light-path reservation delay as a function of wavelength request arrival rate was evaluated. Result obtained indicates that the proposed algorithm gives the lowest reservation delay as per impact of request for lambda arrival rate. This means, of the three algorithms compared, the proposed adaptive algorithm always tries to schedule close to the start time of the scheduling windows as possible.

The proposed algorithm was found to have the least blocking rate in comparison with the other two algorithms. Furthermore findings indicate that the job blocking rate of the proposed algorithm does not vary and oscillate as those of the VBLS and ViFi algorithm. The average blocking rates are 7.55%, 9.02% and 10.64% for the proposed algorithm, the VBLS and the

ViFi algorithm respectively. These values show that the proposed algorithms achieved a 16% and 29% improvement over the VBLS and the ViFi algorithm respectively.

Evaluation of the variations of effectiveness of the algorithm with job size was carried out. Finding show that the effectiveness reduces with increase in job size, Simulation results indicates that the proposed adaptive algorithm performs 18% better that the VBLS algorithm and 21% better than the ViFi heuristic grid scheduling algorithm.

The main contribution of this work is the algorithm for the adaptation of the LIST scheduling algorithm to blend the Dijikistra algorithm and TPSP algorithm. This scheme is found to improve on the large file first and to optimize the light path determinations computation.

5.2 Recommendation

It is of vital importance that a high end national optical transport is made available to Nigerians and international researchers for measurement, experimentation and business operation purposes. A key recommendation here is that a national lambda grid network (probably with the code name **NigerGrid**) should be constructed. Specific wave length should be allocated for intensive e-science project, specific computational grids, Tele-presence or other scientific experiments. NigerGrid is to provide the real physical environment not only to move the algorithm proposed in this work to operational status, but to also enable research in to innovative optical transport technique and to aid high end, complex e-science research and enhance e- collaboration among Nigeria Universities and research centers across the country.

This project report recommends that the management of NigerGrid be constituted under a joint arrangement comprising Nigerian Universities, Research Centers and Nigeria Defense incorporation.

In the present work minimization of finish time is the main objective function in the design of the adaptive lambda grid scheduling algorithm, it is here recommended that further work should expand on the objective function to include minimization of transmission energy consumption and the compute cycles on the OC 192 core network node device during file aggregation.

REFERENCES

1. Aanchal Bawa, Sonam Bhata and Varinder Kamr., Altr: "A review of Agent, April, 2015.
2. Oriented software engineering" *International Journal of advanced Research in computer and communication Engineering*, 4(4).
3. Adami, D., Giordana, S., Repeti, M., Coppola, M., Laforenza, D. and Tonellotto, N., Design and Implementation of a grid network-aware resource broker. In: *proceeding of the IASTED international Conference on parallel and distributed computing and networks as part of the 24th IASTED International Multi-conferences on Applied information*, 2006.
4. Banerjee, A et.al, "*A time –path scheduling problem (TPSP) for aggregating large data file from Distributed database using an optical Burst-Switched Network*", Proc. ICC 2004, parts, france, 2004.
5. Banerjee, A., Ferg, W., Ghosal, D. and Mukherejee, B. Algorithms for integrated Routing and Scheduling for aggregating data from distributed resources on a Lambda grid.*IEEE Transaction on parallel and Distributed Systems*, 2008; 19: 14-34.
6. Bunya, R. *High Performance Cluster Computer: Architecture and Systems*. Vol.1 and 2. PTR upper saddle River, NJ, USA: Prentice Hall, 1999.
7. Buyya, R., David, A. and Jonathan, G., Grid resource management Scheduling and Computational Economy. In: *School of Computer Science and Software Engineering, Monash University, Australia*, April 10, 2002.
8. Castillo, C., Rouskas, G.N and Hanfoush, K., On the design of online Scheduling Algorithm for Advance Reservation and Qos in Grids. In: *Processing of 21st IEEE Termination Parallel and Distribution Processing Symposium*, Aveiro, Portugal, 2007.
9. Coffman, E.G., Garey, M., Johnson, D and Lapaugh, S. (Aug.) Scheduling File Transfers. *SIAM Journal on Computing*, 1985; 14(3).
10. Defanti, T., LOAT, C., Mambretti, J., Naggers, K., Arnomd, B., and Ranslight, T., Global scalar Lambda grid for e-science. In: *Communication of Association of computing machinery*, 2003; 46.
11. DOE: *Ultra Science Net Testbed*. http://www.csm.ornl.gov/ultanet/overview.pdf.
12. Gu, Y., Grossman RL., UDT: UDP- based data transfer for high speed area Network. *Journal of Grid Computing*, In: *proceeding of the 2004 ACM/IEEE Conference on Super Computing*, 2004; 1(4): 377-386.
13. Guerin, R. A. and Orda, A., Network with Advance Reservations: The Routing Perspective In: *Proceeding of Information Communication.* Tel Aviv, Israel, 2000.

14. Hall, N.G and Sriskandarajah, C., A Survey of Machine Scheduling Problems with blocking and no – wait in process. *Journal operations Research*, June, 1996; 44(3): 510-525.
15. He, E., Leigh, J., Yu, O. and Defanti T.A., Reliable Blast UDP: *Predicable high performance bulk data transfer*. IEEE 1st Conference on Cluster Computing, 2002.
16. Lakshmiraman, V. and Ramamurity, B., Joint Computing and Network Resource Scheduling in a Lambda Grid Network. In: *Proceedings of IEEE International Conference on Communications (ICC 2009),* 2009.
17. Larry, S., *Optiputer*: Universty of Califonia, San Diego, 2004. Ismarr@ucsd.edu. Availble at http://www.evl.uic.edu/cavern/optiputer.
18. Lee, H., Veeraraghavan, M., Li, H. and E.K. P. Chong. *Lambda Scheduling Algorithm for file transfers on high speed optical circuits*, 2004. Available at http://www.ece.virginia.edu/mv/pdf-files/ccgrid-gan04.pdf.
19. Liu, X., Wei, W., Qiao, C., Wang, T., W.Hu, Guo, W., and Hu, W., Task Scheduling and Light path Establishment in 0ptical Grid. In: *proceeding of the INFOCOM 2008 Mini. Conference Program.* Phoenix, Arizona, USA, 2008.
20. Nagaswara, S.V. Rao, William, R. W, Steven, M. C, and Qishi, W., Ultra Science Net: *Network Testbed for large-Scale Scince Application*. DOI:10.1109/MCMOM.2005.1541694. Source: IEEE Xplore, 2005.
21. Nagaswara, S.V.Rao, William, G, Bang, Y and Redhakrishnan, S., *Algorithm for Quickest Path under Different Routing modes*. IEICE Transaction on Communication, 2004; 87(4).
22. Page, A.J., and Naughton, T.J., Dynamic task Scheduling using Genetic Algorithm for Heterogeneous Distributed Computing: *Artificial Intelligence Review*, 2005; 24.
23. Pieter T. S, Marc, D. L., Bruno V, Filip, D. T, Bart D, Piet, D., *Scalable dimensioning of resilient Lambda Grids Science one*, 2007. Available at www.sciencedirect.com. April 12, 2015.
24. She, O., Huang, X., Kannasoot, N., and Zhang, O., Multi-resource many cost over optical burst switched network. In: *Proceedings of 16th National Conference on Computer Communications and Networks* (ICCCN 2007), Honolulu, Hawaii, USA, 2007.
25. Shen, S., Wei, G., Weiqiang, S., Yaohu, J., Weisheng, H., Scheduling and Routing Algorithm for Aggregating Large Data Files from Distributed Databases to Super Computers on Lambda Grid. In: *Proceeding on Network Architectures, management and Applications* V1, 71372D, Nov.19, 2008.

26. Smarr, L., Chien, A., Defanti, T., Leigh, J., Philip, M., and Papadopoulos, *The Optiputer Communication of the Association for Computing Machinery*, Nov.2003; 46(11): 58-67.
27. Taesombut, N., Wu, XR., Chien, A., Nayak, A. Smith, B., D.kilb, T.lm, Kent, D., kent, G., and Orcutt, J., *Collaborative data visualization for Earth Science with the optiputer future Generations Computer system*, 2006; 255-63.
28. Veerarghavan, M., Lee, H., Chong, E.K.P., and Li, H., "*A varying-Bandwidth List Scheduling heuristic for file transfer*" Researchgate DOT: 10.11081cc, July, 2004.
29. Wang Xia-hong., *Application of Cisco Packet Tracer Simulation software in Network project Teaching*, 2012; 32-25. en.cnki.com.cn/Article en/CJFDTOTAL-DNKF201205024.htm.