Centralized Algorithm to achieve a maximum Utilization for Successful Speculative Transmission

Professor Dr. Sudan Jha
Professor, School of Computer Engineering
KIIT University, Patia
Bhubaneswar, Odisha, India

Abstract- A Centralized Algorithm to achieve a maximum Utilization for Successful Speculative Transmission concept was motivated by the need to achieve low latency in an input-queued centrally-scheduled cell switch for high-performance computing applications; specifically, the aim is to reduce the latency incurred between a request and response arrival of the corresponding grant. The minimum latency in switches with centralized scheduling comprises two components, namely, the control-path latency and the data-path latency, which in a practical high-capacity, distributed switch implementation can be far greater than the cell duration. We introduce a speculative transmission scheme to significantly reduce the average control-path latency by allowing cells to proceed without waiting for a grant, under certain conditions. It operates in conjunction with any centralized matching algorithm to achieve a high maximum utilization. Using this model, performance measures such as the mean delay and the rate of successful speculative transmissions are derived. The results demonstrate that the latency can be almost entirely eliminated between request and response for loads up to 50%. Our simulations confirm the analytical results.
Keywords – Latency, Switch, Speculative Transmission, OSMOSIS Architecture, Brikoff-von-newmann Switch,

I. INTRODUCTION

A key component of massively parallel computing systems is the interconnection network (ICTN). To achieve a good system balance between computation and Communication, the ICTN must provide low latency, high bandwidth, low error rates, and scalability to high node counts (thousands), with low latency being the most important requirement. On this research work, the "Optical Shared Memory Supercomputer Interconnect System" (OSMOSIS) architecture is been proposed resistively to reduce the latency incurred between issuance of a request and arrival of the corresponding grant. Using this model, performance measures such as the mean delay and the rate of successful speculative transmissions are derived. The results demonstrate that the control-path latency can be almost entirely eliminated for loads up to 50%. The simulations confirm the analytical results. This work has a scope to reduce the latency and also achieve high maximum throughput because the current Existing System a.**Brikoff-von-newmann Switch** which eliminates the scheduler. It incurs a worst-case latency penalty of N time slots. It has to wait for exactly N time slots for the next opportunity; and b. Control and data path-latencies comprise serialization and de-serialization delays, propagation delay, processing delay between request and response.

However, the disadvantages that has yielded during the course of this research work are listed as below: -

- The existing system is if n_packets sending source to destination, it is exactly wait for n_time slots.
- Serialization and deserialization delays between request and response

Hence, the proposal is a novel method to combine speculative and scheduled transmission in a cross bar switch.

a. Speculative modes of operation reduced latency at low utilization.

b. Scheduled modes of operation achieve high maximum throughput.

Advantage

i. The speculative transmission that does not have to wait for grant hence low latency.

ii. The scheduled transmission achieve high maximum throughput.

II. PROPOSED ALGORITHM

A. OSMOSIS Architecture –

The routing fabric of OSMOSIS is entirely optical and has no buffering capability. It operates in a synchronous, time-slotted fashion with fixed-size packets (cells). Switching function is implemented using fast semiconductor optical amplifiers (SOAs) in a broadcast-and-select (B&S) structure using a combination of eight-way space- and eight-way wavelength-division multiplexing, thus providing bidirectional connectivity for 64 nodes. Electronic buffers store cells at the ingress of the switch, resulting in an input-queued (IQ) architecture. To prevent head-of-line (HOL) blocking, the input queues are organized as virtual output queues (VOQs).The B&S switch fabric structure is the optical equivalent of an electronic crossbar switch. To resolve crossbar input and output contention, central scheduling is required, which is also electronic.

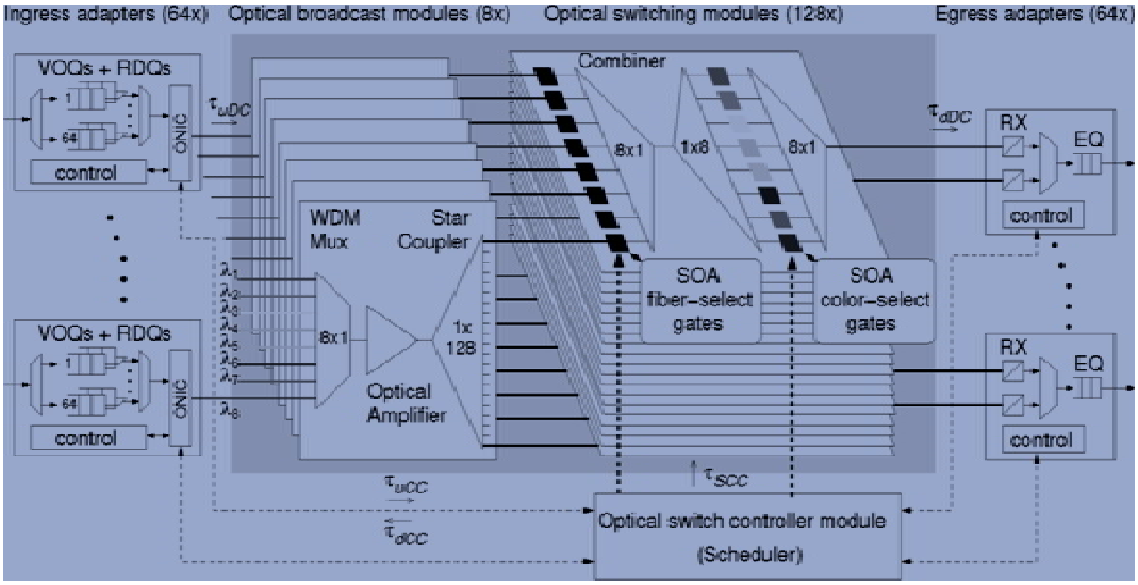


Fig 2.1 OSMOSIS Architecture

In addition to a low minimum latency, OSMOSIS must also be able to achieve a high maximum throughput. Therefore, the scheduler must implement an appropriate bipartite graph matching algorithm able to sustain close to 100% throughput. Using appropriate deep pipelining techniques it is possible to obtain maximal matching even for switches with many ports and short cells. The input adapters receive cells from the incoming links and store them according to their destinations in the VOQs. Upon cell arrival, a request is issued to the scheduler via the control channel (CC), which is operated in a slotted fashion with the same time slot duration as that of the data path. When the round-trip time (RTT, expressed in time slots) is greater than 1, both the data and the control path must be operated in a pipelined fashion to maintain 100% utilization without increasing the cell size. This implies that multiple cells and request/grants may be in flight on the data and control paths, respectively. To cope with a long Round Trip Time (RTT) without loss of performance, we employ an incremental VOQ state update protocol that allows deep pipelining of requests and grants without a performance penalty.

B. Control Path Latency–

This classic centrally-scheduled, crossbar-based IQ architecture, however, incurs a latency penalty: The minimum latency of a cell in the absence of contention comprises two components, namely, the control-path latency and the data-path latency .1 The former consists of the latency from the issuance of a request to the receipt of the corresponding grant, whereas the latter consists of the transit latency from the input adapter to the output adapter. The switch-configuration-path latency represents the latency from the issuance of a configuration command by the scheduler until the SOAs are switched accordingly. These latencies comprise serialization and deserialization (SERDES) delays, propagation delays (time of flight) on the physical medium, and processing delays in the switch and the adapter. The processing delays typically include header parsing delays, routing delays, scheduling delays, pipelining delays, etc. In an output-queued (OQ) switch, on the other hand, the minimum latency comprises only the data-path latency. The difference is that in an IQ switch, a newly arriving cell must first request permission to proceed and then wait for a grant, whereas in an OQ switch, a cell can immediately proceed to its output when there

is no contention. The physical implementation and packaging aspects of OSMOSIS (and high-capacity switches in general) have important consequences that imply that the above latencies are significant. In the OSMOSIS demonstrator, we estimate the involved data- and control-path latencies to amount to a minimum cell latency of approx. 1.3 s which is much larger than the cell duration (51.2 ns). *This already exceeds our latency target of 1 ms without taking into account the latencies of the driver software stack and the network interface card.*

Parallel ICTNs often operate at low utilization, or are subjected to highly orchestrated (by the programmer or compiler) traffic patterns. Under such conditions, the mean latency is dominated by the intrinsic control- and data-path latencies rather than by queuing delays. Hence, optimizing latency for such cases improves overall system performance. The main contribution of this work is a hybrid crossbar scheduling scheme that combines scheduled and speculative modes of operation, such that at low utilization most cells can proceed speculatively without waiting for a grant, thus achieving a latency reduction of up to 50%. Moreover, the scheduled mode ensures high utilization without excessive collisions of speculative cells in the B&S switch fabric.

C. OperationalFeasibility–

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

i. *Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.*

ii. *Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.*

iii. *Have the user been involved in the planning and development of the project?*

iv. *Early involvement reduces the chances of resistance to the system and in*

v. *General and increases the likelihood of successful project.*

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

III. SOFTWARE DESIGN PROCESSES

D. Introduction–

The main focus of the analysis of the objects discovered that can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution. Emphasis shifts from the application domain tyo implementation and computer such as user interfaces or view layer and access layer. During analysis, the physical entities or business objects in the system are well taken care-of, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business. During the Design phase, the model into logical entities have been elevated, some of which might relate more to the computer domain as people or employees. Here the goal is to design the classes that are needed to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way to interact with a user or present information.

E. Design Process–

During the design phase the classes identified in object-oriented analysis must be revisited with a shift focus to their implementation. New classes or attribute and Methods must be an added for implementation purposes and user interfaces. The object-oriented design process consists of the following activities:

1. Apply design axioms to design classes, their attributes, methods, associations, structure
2. Design the access layer
3. Iterate refine the whole design process.

F. Design Axioms–

Axioms are a fundamental truth that always is observed to be valid and for which there is no counter example or exception. Such explains that axioms may be hypothesized form a large number of observations by nothing the common phenomena shared by all cases; they cannot be proven or derived, but they can be invalidated by counter examples or exceptions. A theorem is a proposition that may not be self-evident but can be proven from accepted axioms. If therefore, is equivalent to a law or principle. A corollary is a proposition that follows from an axioms or another proposition that has been proven. Again, corollary is shown to be valid or not valid in the same manner as a theorem. In the two important axioms axiom 1 deals with relationships between system components and axiom 2 deals with the complexity of design.

The following the two important axioms:

Axiom 1: The independence axiom, which maintain the independence of the components. Axioms1 states that, during the design process, as we go from requirement and use case to a system component, each component must satisfy that requirement without affecting other requirements.

Axiom 2: The information axioms that maintain the information content of the design. An axiom 2 is concerned with simplicity. Scientific theoreticians often rely on a general rule known as Occam's razor, after William of Occam. He says, "The best theory explains the known facts with a minimum amount of complexity and maximum simplicity and straightforwardness."

The best designs usually involve the least complex code but not necessarily the fewest number of classes or methods. Minimizing complexity should be the goal, because that produces the most easily maintained and enhanced application. In an object-oriented system, the best way to minimize complexity is to use inheritance and the systems built in classes and to add as little as possible to what already is there. From the two design axioms, many corollaries may be derived as a direct consequence of the axioms. These corollaries may be more useful in marking specific design decisions, since they can be applied to actual situations.

1. Uncoupled design with less information content: Highly cohesive objects can improve coupling because only a minimal amount of essential information need be passed between objects. The degree or strength of coupling between two components is measured by the amount and complexity of information transmitted between them.

2. Single purpose: Each class must have single, clearly defined purposes.

3. Large number of simple classes: Keeping the classes simple allows reusability. Large and complex classes are too specialized to be reused.

4. Strong mapping: There must be a strong association between the physical system and logical design. During the design phase, we need to design this class, design its methods, its association with other objects. So a strong mapping links classes should be identified.

5. Standardization: promote standardization by designing interchangeable and reusing existing classes or components.

6. Design with inheritance: Common behavior must be moved to super classes. The super class-sub class structure must make logical sense.

G. Refining attribiutes and methods–

Attributes identified in object oriented analyzed must be refined in the design phase. In the analysis phase, the name of the attributes was sufficient. But in the design phase, detailed information must be added to the model. The three basic types of attributes are:

1. Single valued attributes: This has only value or state.

2. Multiplicity or multivalve attributes: This has a collection of many values at any point in time.

3. Instance connection attributes: This is required to provide the mapping needed by an object to fulfill its responsibilities.

H. UML Attribute Presentation–

Visibility indicates either public visibility or protected visibility or private visibility. The public visibility indicates that the attribute can be accessible to all classes. The protected visibility indicates that the accessibility is given to the subclasses and operations of the class. The private visibility indicates that the accessibility can be given only to the operations of the class only. Type expression is a language dependent specification of the implementation type of an attribute. Initial value is a language dependent expression for the initial value is optional.

The design may be defined as “the process of applying various techniques and principles for the purpose of defining a device, a process or a system with sufficient details to permit its physical realization”. The importance of software design can be stated in a single word “quality”.

IV. Software Design Implementation

A. Data Flow Diagram–

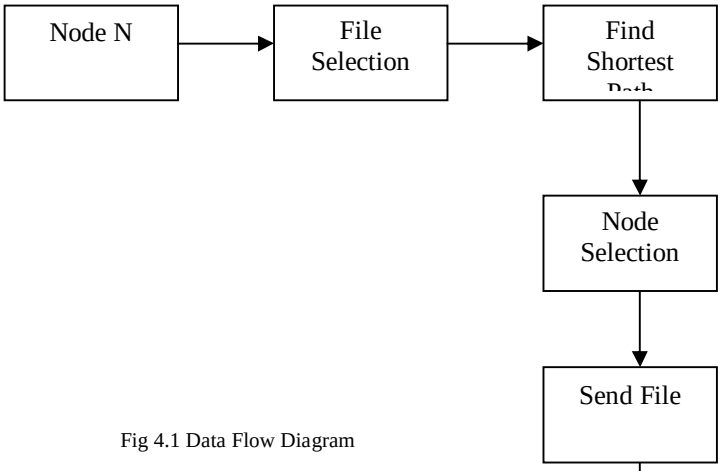


Fig 4.1 Data Flow Diagram

B. Architecture Diagram–

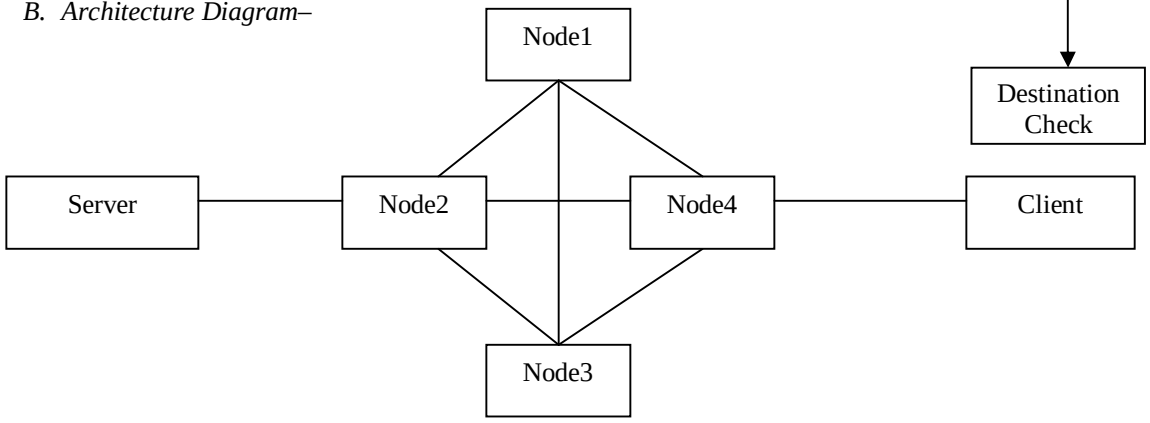


Fig 4.2 Architecture Diagram

C. Input and Outpiut Design–

Interface design: The current trend in software industry is user friendliness and flexibility. The two main factors contributing towards this are Screens and Menus. The screens in the system are designed to be self-descriptive so as to direct the user while using the system. The screen format is user friendliness. **Input design:** Inaccurate input data are most common cause of errors in data processing. Input interface design takes an important role in controlling the errors. Messages are generated using the exception handling feature of JAVA. The input forms are designed in flexible way. **Output design:** It describes how to show all the information and data after processing input data.

D. UML Diagram–

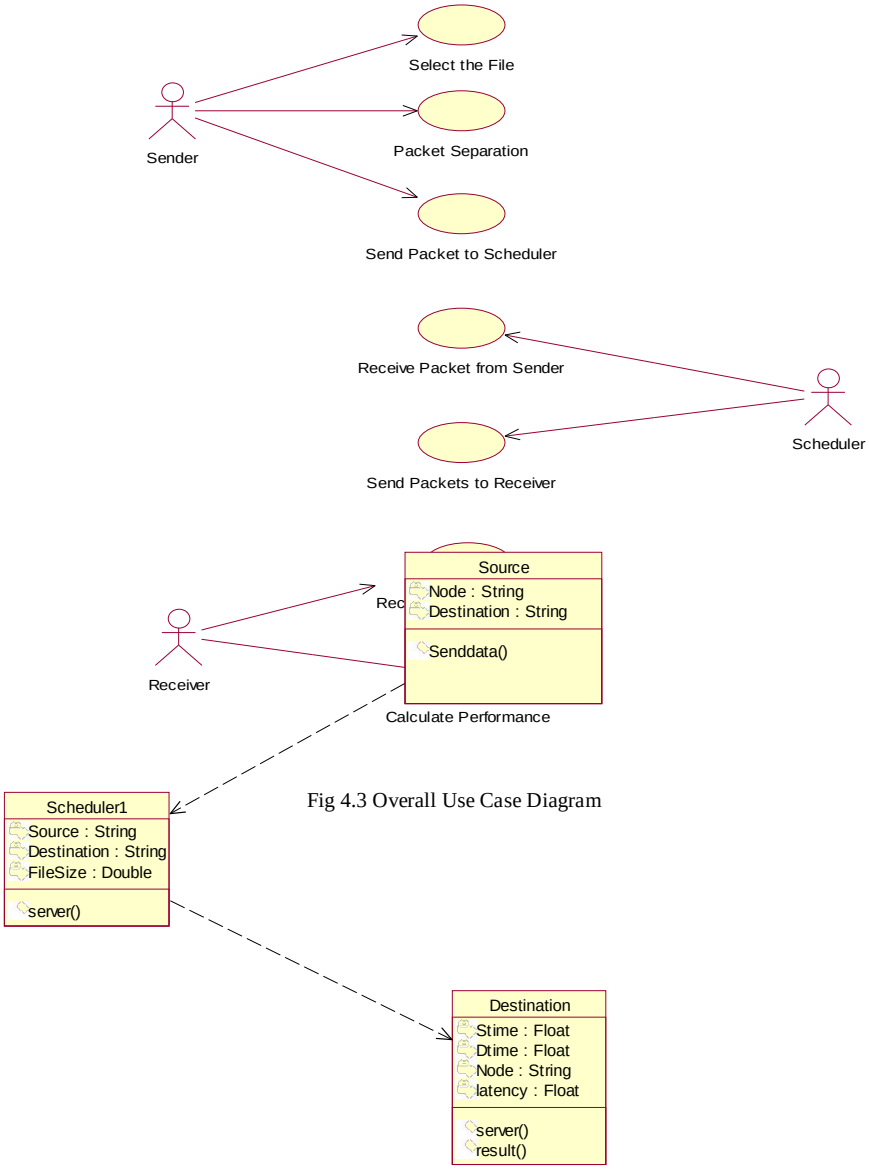


Fig 4.3 Overall Use Case Diagram

Fig 4.4 Class Diagram

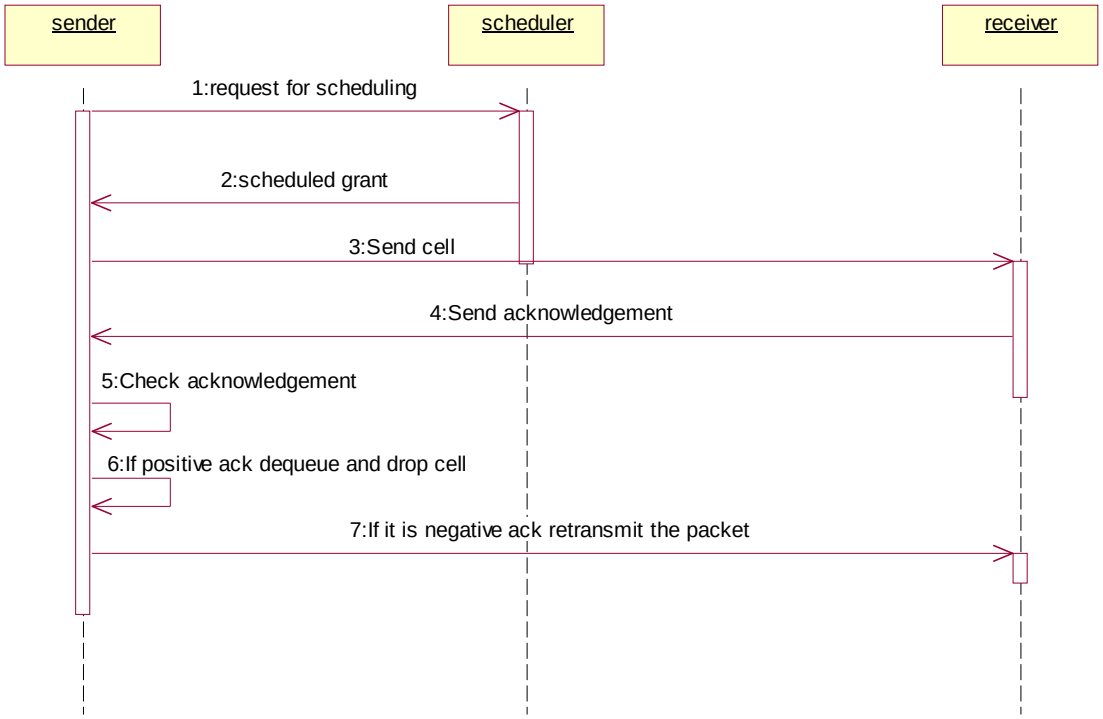


Fig 4.5 Sequential diagrams for Retransmission

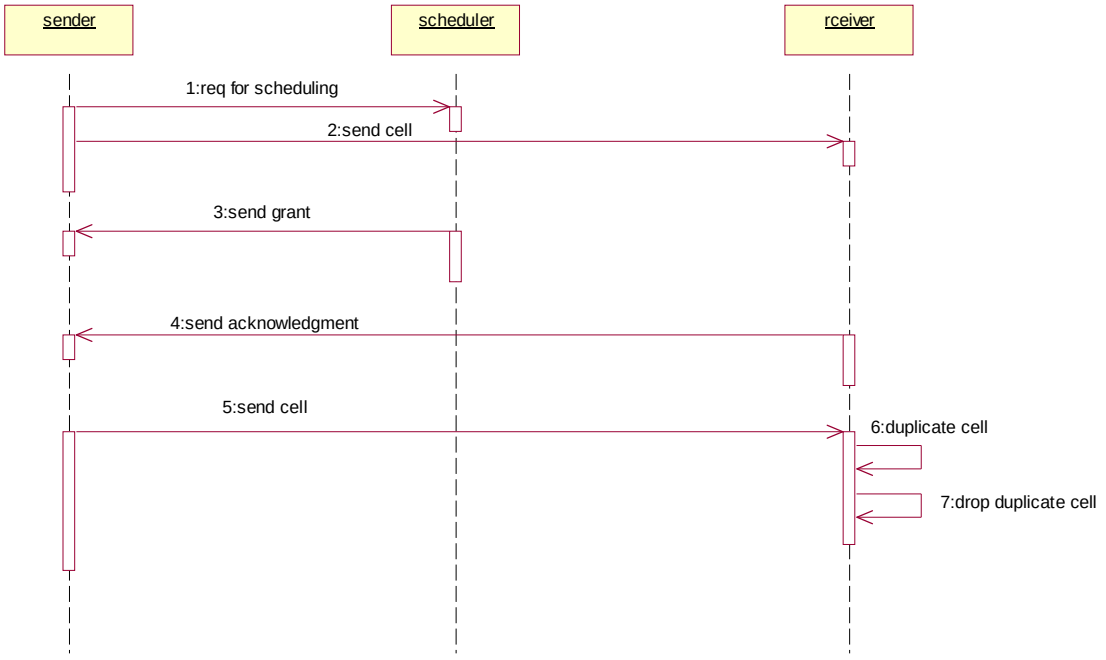


Fig 4.6 Sequential diagram for duplicate packet

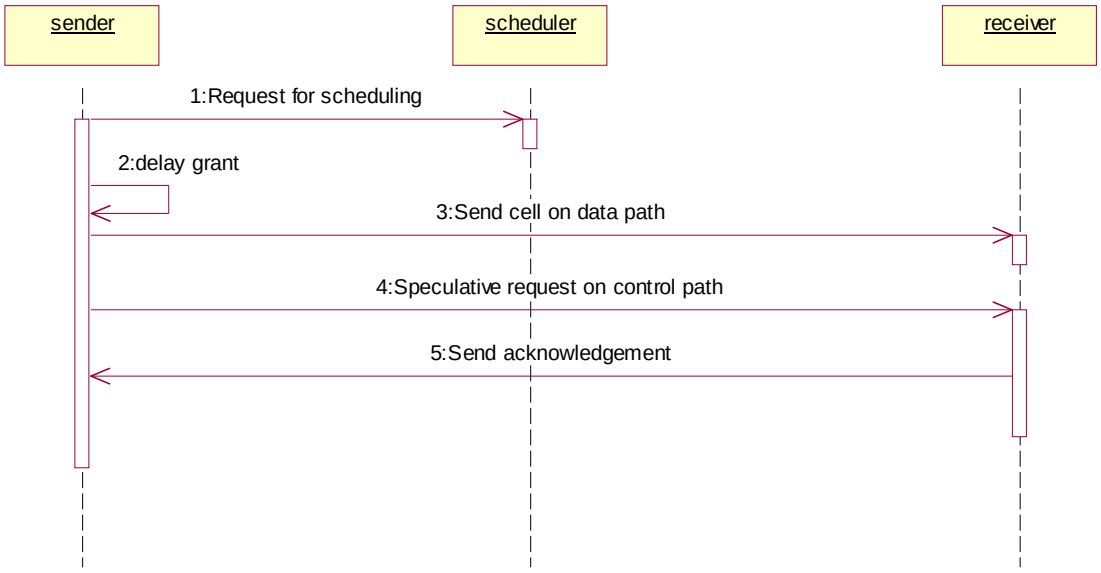


Fig 4.7 Sequential diagram for delay grant

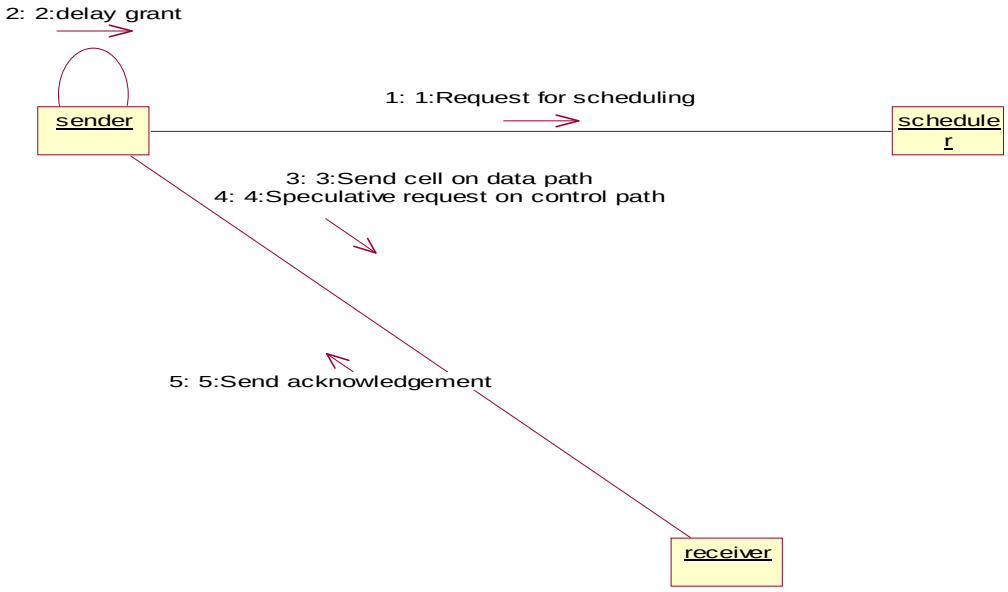


Fig 4.9 Collaboration diagram for delay grant

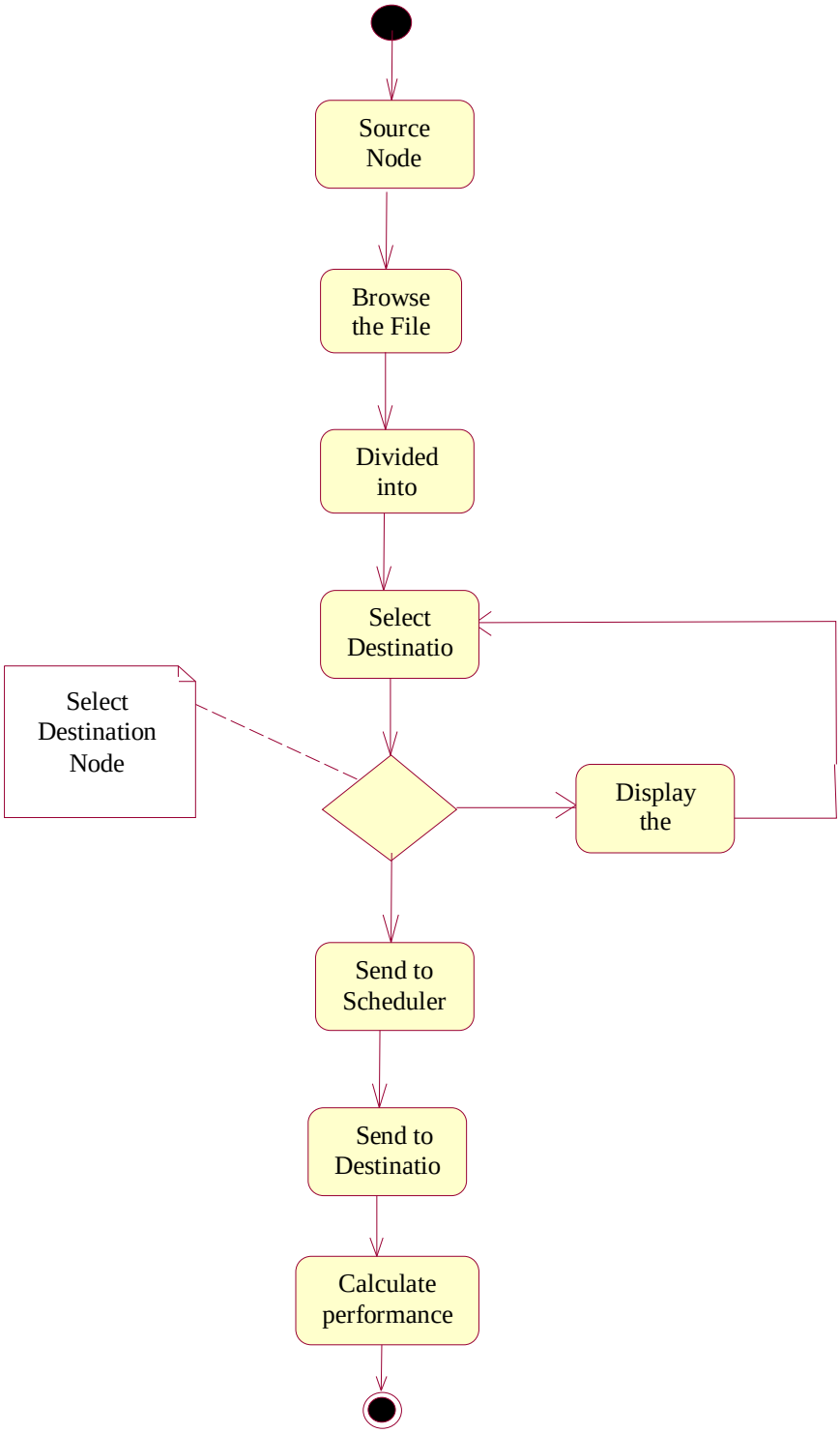


Fig 4.10 Activity Diagram

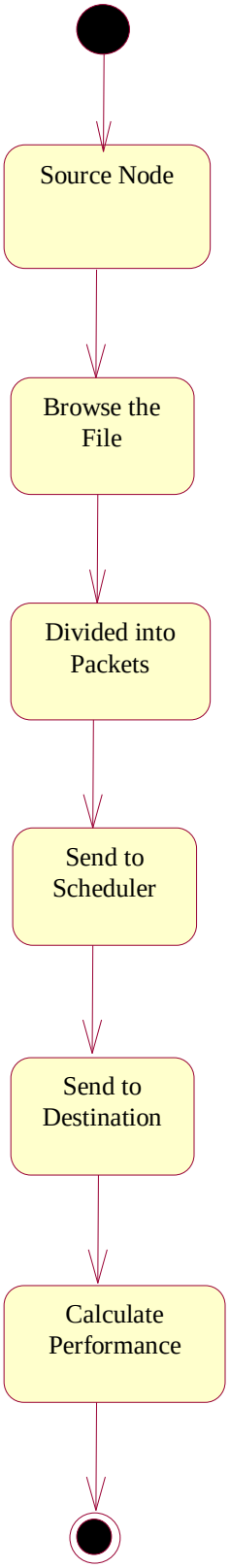
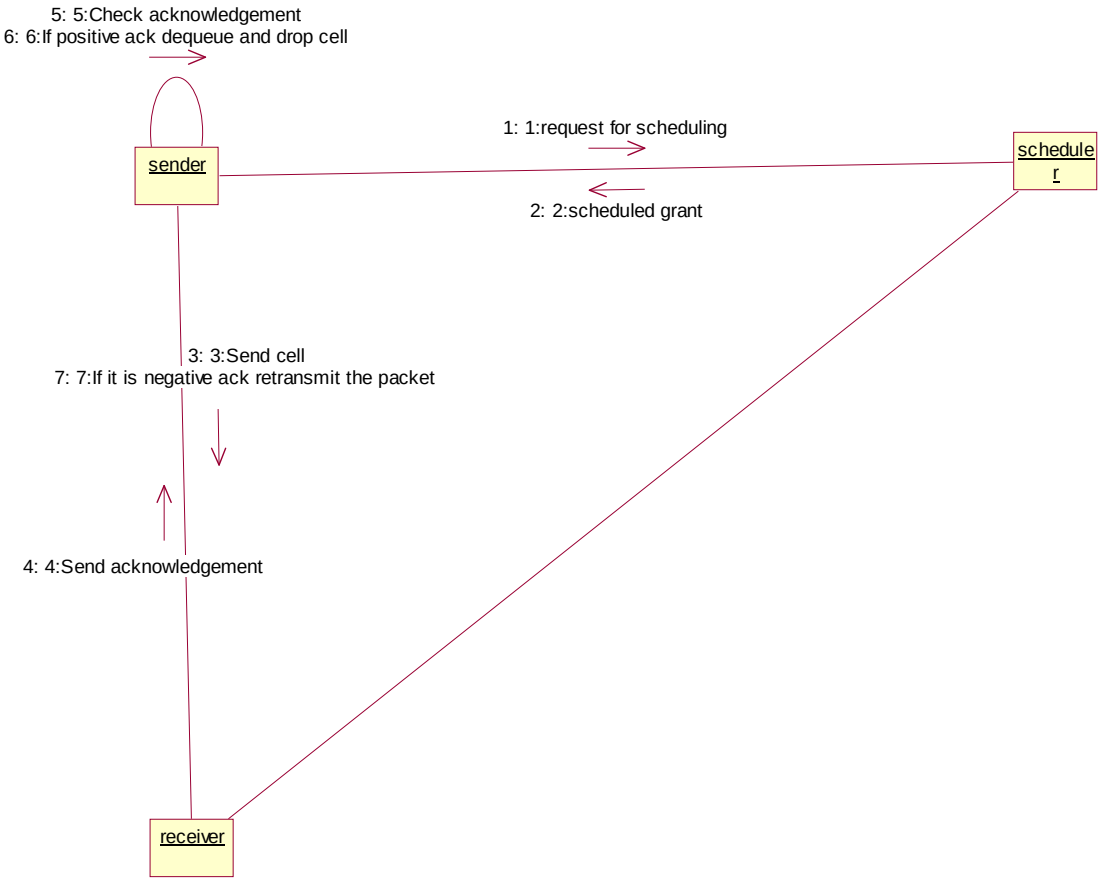


Fig 4.11 State diagram


5: 5:Check acknowledgement
6: 6:If positive ack dequeue and drop cell
sender
1: 1:request for scheduling
2: 2:scheduled grant
schedule
r
3: 3:Send cell
7: 7:If it is negative ack retransmit the packet
4: 4:Send acknowledgement
receiver

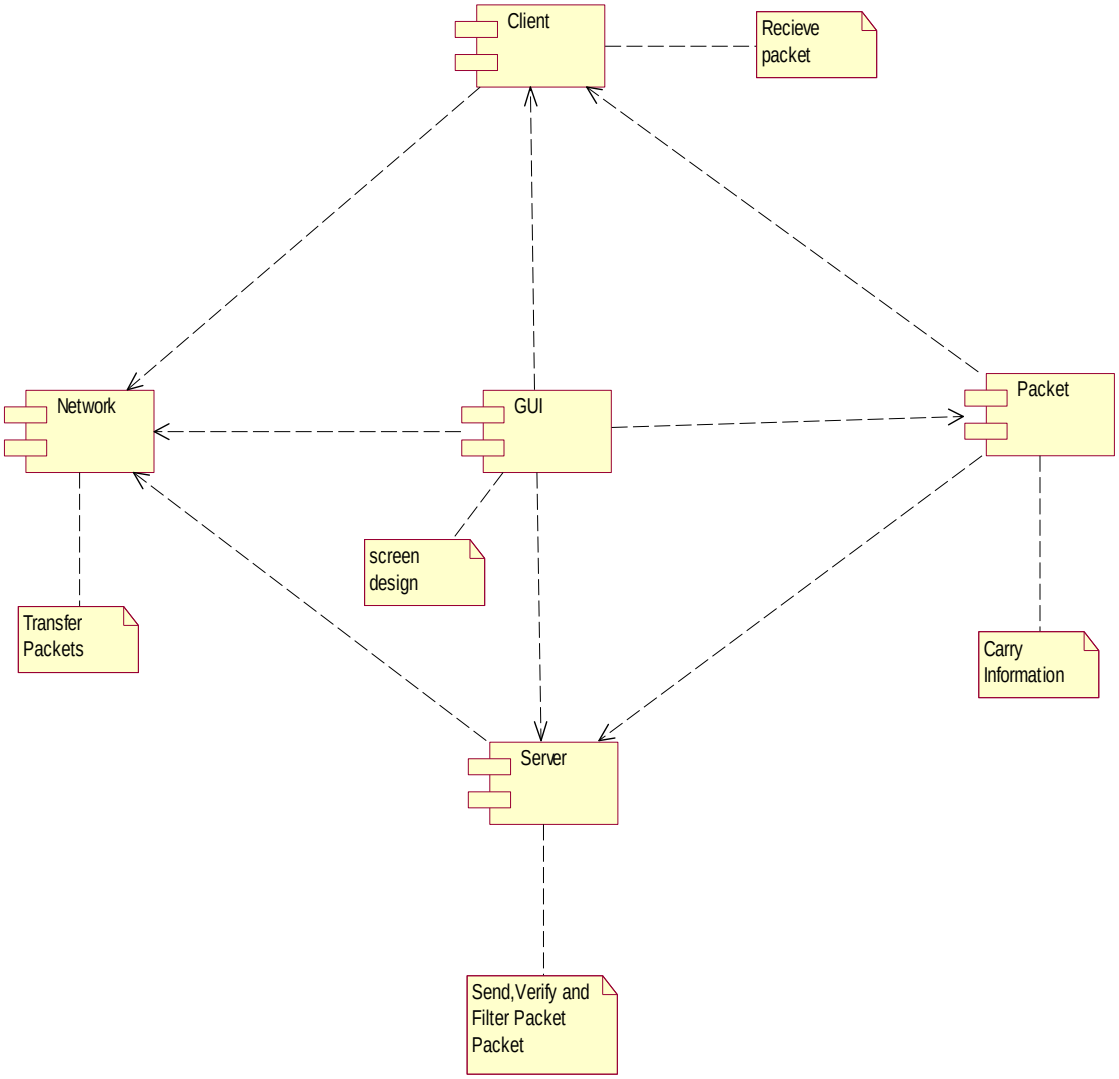


Fig 4.12 Component Diagram

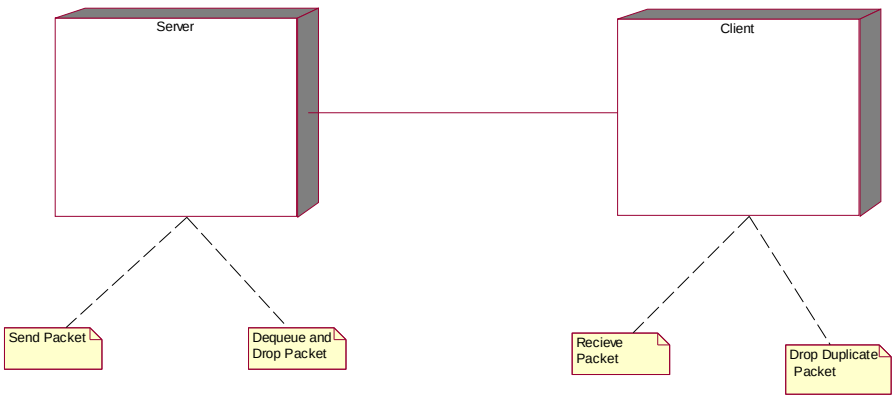


Fig 4.13 Deployment Diagram

IV. Software System Testing

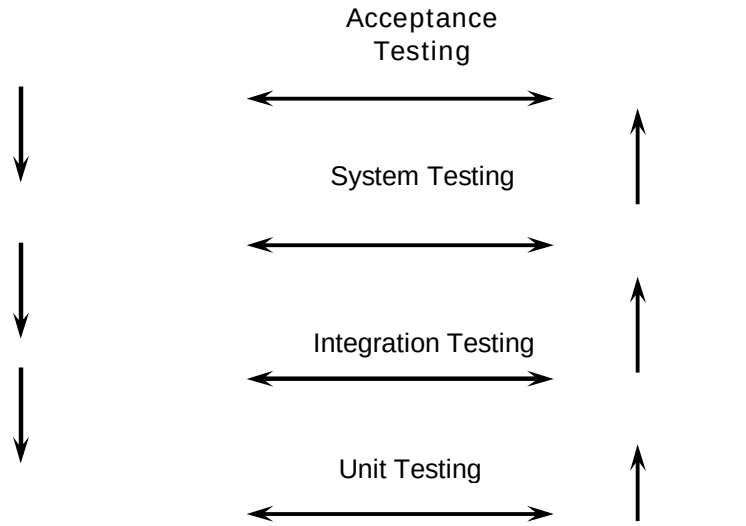
A. *Testing Objectives–*

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn't work. Testing is the process of executing a program with the intent of finding errors. The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

- Testing is a process of executing a program with the intent of finding an error.
- A successful test is one that uncovers an as yet undiscovered error.
- A good test case is one that has a high probability of finding error, if it exists.
- The tests are inadequate to detect possibly present errors.

B. *Levels of Testing–*

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…



V. Conclusion And Future Enhancements

Fig 4.14 Levels of Testing

This work was motivated by the need to achieve low latency in an input-queued centrally-scheduled cell switch for high- Performance computing applications; specifically, the aim is to reduce the latency incurred between issuance of a request and arrival of the corresponding grant. The proposed solution features a combination of speculative and scheduled transmission modes, coupling the advantages of uncoordinated transmission that is not having to wait for a grant, hence low latency, with those of coordinated transmission, which is high maximum utilization. In this project by using speculative transmission scheme we can send the data files from one end to another end with low latency. But we cannot send the images and video files from one end to another end. So in future we transmit the images and video files from one end to another end.

VI. Conclusion And Future Enhancements

[1] LA. Tanenbaum, *Computer Networks*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1996.

[2] C. Minkenberg, F. Abel, P. Müller, R. Krishnamurthy, M. Gusat, P.Dill, I. Iliadis, R. Luijten, B. R. Hemenway, R. Grzybowski, and E.Schiattarella, "Designing a crossbar scheduler for HPC applications,"*IEEE Micro*, vol. 26, no. 3, pp. 58–71, May/Jun. 2006.

[3] C. Minkenberg, I. Iliadis, and F. Abel, "Low-latency pipelined crossbar arbitration," in *Proc. IEEE GLOBECOM 2004*, Dallas, TX, Dec. 2004, vol. 2, pp. 1174–1179.

[4] A. A. Reddy and B. N. Chatterji, "A new wavelet based logo-watermarking scheme," Pattern Recognition Letters, vol. 26, pp. 1019-1027, 2005.

[5] P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, "Robust spatial watermarking technique for colour images via direct saturation adjustment," Vision, Image and Signal Processing, IEE Proceedings -, vol. 152, pp. 561-574, 2005.

[6] F. Gonzalez and J. Hernandez, " A tutorial on Digital Watermarking ", In IEEE annual Carnahan conference on security technology, Spain, 1999.

[7] Kunder, "Multi-resolution Digital Watermarking Algorithms and Implications for Multimedia Signals", Ph.D. thesis, university of Toronto, Canada, 2001.