CHAPTER 6

ARBITRARY SELECTION FOR AGITATE FLEXIBLE LOAD BALANCING IN HETEROGENEOUS PEER-TO-PEER NETWORKS

6.1 INTRODUCTION

P2P Network access is the integral part of the day to day business for most of the computer users. Preserving load balance on heterogeneous P2P networks is a challenging one and several existing works present load redistribution algorithms for load division and load diversions at the case of new peer arrives. The existing work Saturn concentrates on efficient query processing, load balancing and fault tolerance. But the impact of maintaining extra links to peers routing tables causes network traffic. The principle objective of load balancing in P2P networks is to maintain the workload of the network nodes in quantity but lacks to eliminate traffic occurring in the routes of the P2P network efficiently.

A novel Arbitrary Selection Load Balancing (ASLB) approach with Tremendous Store model is proposed to eliminate the traffic by characterizing behavior. Tremendous store model is investigated to extend the impact of node heterogeneity and agitate to the load distribution in P2P networks. The performance of the Arbitrary Selection Load Balancing (ASLB) Approach with Tremendous Store model reduces the network traffic routes with minimal cost of P2P in contrast to existing Saturn load balancing scheme. An analytical and empirical result offers a collision free system with the
information being quantized by balancing the load in the P2P network system. Performance of the Arbitrary Selection Load Balancing is measured in terms of peer failure probability and traffic control efficiency.

6.2 FACTORS AFFECTING THE P2P NETWORK

In recent years, P2P networks have raised in popularity in the form of file sharing applications. With this popularity comes the load unbalancing and network traffic or congestion.

6.2.1 Load Balancing Strategy

The determination factor supported by the balancing mechanism is the computational load of the hosts and the nodes in the system. The origin of the load is examined exactly for the purpose of choosing the optimal balancing action. There are two basic reasons for the cause of an overload:

i. A request query costing on the local data is too time-consuming.

ii. A request is waiting for processing since other operations are being processed.

While the prior state mainly increases the response time of a single even inaccessible search-query, the second one manipulates the throughput of the system. According to this surveillance, the load is measured as two distinct quantities as follows.

i. The processing load is utilized as a standard time of handing out a search query on the local data of a given node. The extracted value sustains the data volume, the difficulties of the queries and ultimately the quality of the local index.

ii. The waiting load is forecasted as an average time a query is waiting in the queue before processed. This value is
calculated within the range of the complete physical host and it believes the occurrence of queries and the processing load of specific nodes at this host.

Both quantities are also inclined by the computational performance of the physical host and by numerous outer concerns. The average time taken over a time slot or over a number of processed queries in processing and waiting load. As the processing load is self-determining of the requests range this capacity is calculated as an average over numerous last processed requests. While the waiting load considers the request range the average is extracted to form a group of query that suits into time sliding window.

The following representation exposes the load-balancing strategies useful for every host in the framework.

IF the **waiting load** is too large THEN
    IF the waiting is due to more than one node then
        MIGRATE one of these nodes to another host,
    Else REPLICA the load-causing node.

IF the **processing load** of a node is too large THEN
    SPLIT the node to another host.

IF the **waiting load** is small and a replica exists THEN
    Apply the Unifying operation remove a replica.

IF the **processing load** is small THEN
    LEAVE the node.

The representation includes indistinct words like too large or small. The values are relevant to the original load of the complete system of other hosts. Each host preserves original estimations of the overall averages of the load quantities. The information is preserved by a distributed P2P algorithm based on conversations. The algorithm exposes average messages incorporating query, insertion, and management to swap the data and increase the load of the network slightly.
With the original estimation of the overall load values, the term is too large to specify the large means over two times average and the term too small means under half average. Information about the least loaded hosts in the system is distributed together with the average messages additionally. Figure 6.1 provides a simple scheme of the load-balancing module existing at every host.

![Figure 6.1 Work Schema of the Load-Balancing Module](image)

The necessity of the P2P data networks for identical search is beyond the expectation of current load-balancing techniques. There are two metrics required to the load balancing algorithm. One is the time demands of the identical requests and query processing are high and varies for specific network nodes while the existing balancing strategies only considers either the data volume or clean number of accesses, and also not all operations needed by the existing balancing techniques are accessible in the similarity P2P data networks.
Load balancing strategies (Wang et al 2013) for permitting the nodes of the network to provide better communication applications for a huge number of users with hardware devices in a situation of time-varying load distributions and random load fluctuations. But the methods are not applicable to applications including multimedia video streaming applications, pervasive applications, VoIP applications, and environment based advertising applications. The load balancing challenge is faced through novel creation of Arbitrary Selection Load Balancing (ASLB) approach. In addition to all existing load balancing handled in P2P network next challenging factor is the network traffic.

6.2.2 Network Traffic

Applications depending on P2P architectures gained more massive interest due to file sharing or receiver system. The raise of P2P network continues due to various researches on P2P data variations are presently under consideration. This trend is continued because of the fact that numerous projects based on P2P data exchanges are currently insufficient as presented by Vu et al (2009).

But the growth of P2P traffic is already researched by Dedinski et al (2009) and it raises new issues for P2P network. Between them, the number of concurrent Transmission Control Protocol (TCP) links provided by P2P applications is lower other established Internet applications. Certainly communication between peers is generally executed with multiple successive Transmission Control Protocol TCP links in file sharing applications and in most current video streaming systems. As a result, the relation of Transmission Control Protocol (TCP) links for applications such as HTTP or VoIP (Voice over Internet Protocol) over the complete number of Transmission Control Protocol (TCP) connections is lesser, so the fraction of the bandwidth these P2P applications could use becomes weaker. At the same time, the fair bandwidth distributing device of Transmission Control Protocol
(TCP) fails in ensuring a fair allocation among applications. The network traffic congestion in accessing internet is elaborated in the Figure 6.2.

Figure 6.2 describes the congestion occurring in access point due to overflowing access. Every link attempts to establish a connection to the access router in order to access the data resulting in bottleneck situation due to multiple accesses. This situation is ignored if the network bottleneck was far from the end users. Numerous researches (Hullar et al 2011) unfortunately showed the most network bottlenecks in the P2P network or on the links between nodes.
When congestion occurs on the distributed access network which links router or access router to the network, P2P applications with multiple Transmission Control Protocol (TCP) connections illegally steal bandwidth from other standard Internet applications. Besides, the necessity for everywhere or pervasive network makes the number of nodes connected to one access point, and accordingly the number of applications provided by this first router, increases significantly. This aspect is possible for an application level reasonable allocation of bandwidth. This state is illustrated in above Figure 6.2 where three hosts connected to network through a distinct access point are running four applications using eight connections.

In order to address the issue related with aggressive P2P applications, a novel heterogeneous P2P network is efficiently designed for reducing the network traffic using the Tremendous Store model. Arbitrary Selection Load Balancing with tremendous store model defined a set of policies aiming at controlling P2P traffic using Duty Cycle Data Appropriate technique and node selection strategy.

First challenge consists in identifying this traffic, then to plan a better way to break congestion. Some technologies are implemented (Hullar et al 2011), but they act on the strength of character although the congestion occurs in the access network. Therefore the prevention of congestion is not possible and worst. Mostly some P2P connections are broken for clients experiencing idle network usage. As a consequence, the peer nodes implementing such technologies result in alleviation and even losing clients without actually improving the performance of P2P network. The congestion in accessing the network information also increases tremendously. A better technique termed tremendous store model is proposed to control the network traffic.
6.3 ARBITRARY SELECTION LOAD BALANCING APPROACH FOR FLEXIBLE P2P NETWORK

The heterogeneous P2P network is efficiently designed for reducing the network traffic using the Tremendous Store model. The process of tremendous store model is to identify the load demand balance factor of each peer node in the network. It also identifies the server cycle processing requirements of each peer node using Duty Cycle Data Appropriate technique as well as with the load imbalance factor. Then node selection strategy is done based on the DCDA rank representation.

Arbitrary Selection Load Balancing (ASLB) approach with Tremendous Store model in the heterogeneous P2P network consists of three different phases. The first phase is to identify and analyze the peer node server cycle requirements using Duty Cycle Data Appropriate technique with the load demand balance factor outcome. The magnitude of load demand factor of peer server is measured using Dynamic Time Warping Algorithm (DTWA). In addition the Node Selection Strategy intends in improving the load diversions factor if load imbalance occurs. If load imbalance occurs, Load diversions are performed with peer node server with Node Selection Strategy based on DCDA rank representation.

The second phase evolves load balancing algorithm to engender the best load allocation resolution when a novel peer arrives. The load balancing algorithm is also capable of energetically achieving the load reallocation through system running time, if congested peers emerge. The third phase is to utilize the tremendous store model for analyzing balancing workload in a particular distributed environment. Tremendous store model intends in controlling the network traffic. The architecture diagram of the Arbitrary Selection Load Balancing (ASLB) Approach in heterogeneous P2P network is shown in Figure 6.3.
Figure 6.3 Architecture Diagram of Heterogeneous P2P Network Traffic Control System Using Arbitrary Selection Load Balancing Approach
Figure 6.3 illustrates a brief explanation of the Heterogeneous P2P network traffic control system. In network communication, the sending and receiving of packet data and the arrangement of the data policy to the superior layer finds an easy way.

The first phase describes the process of implementing the Duty Cycle Data Appropriate technique in order to identify the peer node server processing cycle requirements through load demand balance factor. The DCDA technique identifies the processing time of the server node for a cycle per second. An analysis is made on the server peer node cycle requirements using Duty Cycle Data Appropriation technique based on load demand balance factor. Determine the magnitude of load demand balance factor using Dynamic Time Warping Algorithm (DTWA) in order to identify efficiently the liveliness of server peer node in the heterogeneous network and the load balance of the peer server node.

It is necessary to depend on the load demand balance factor to analyze the peer node cycle requirements and solve load imbalance due to node diversity. If peer node server is full resulting in load imbalance then load diversity is performed to solve inconvenience. Load diversity of the peer node is done based on node selection strategy. Selecting the node in order to divert the nodes from peer node to neighbor nodes minimizes the load imbalance. The node selection is based on DCDA rank representations. The second phase concentrates on the load allocation problem and balances the peer node and accordingly the zones are separated. Additionally, load reallocation works in neglecting the node congestion due to peer foliages.

The third phase explains the tremendous store model in the ASLB approach. ASLB utilizes the tremendous store model to analyze workload balance in a particular distributed environment. Client requests arrive as a
Poisson stream at a point of collection of servers. For each demand some regular numbers of servers are chosen separately and consistently in random or arbitrary with replacement and the request waits for service at the server presently containing the smallest requirements. Requests are served according to the FIFO protocol, and the service time for a request is exponentially disseminated with mean 1.

6.3.1 Duty Cycle Data Appropriate Technique

Duty Cycle Data Appropriation technique largely concentrates on recognizing the peer server node cycle requirements. The peer node cycle needs to count the number of cycles the peer server nodes requires to access the specific data for improving the load balancing. The process of DCDA technique is to identify the characteristics of the peer nodes in the network. The characteristics of peer node denote the nodes exchange among active and sleeping periods depending on network activity. DCDA spot the appropriate amount of data to be processed by a peer node server for a cycle. If the peer node server cycle requirements are known, then it is simple for the peer node to distribute its data with other nodes in the heterogeneous P2P networks based on the result obtained through DCDA technique.

The DCDA technique is generally a distributed algorithm based on which peer nodes in the heterogeneous P2P networks select the state regarding when to alter from active to sleep, and back. DCDA allows adjacent nodes to be dynamic at a similar time, thus producing packet data exchange feasible even when nodes function with a distinct duty cycle. A duty cycle is the instance of an entity use in a dynamic position as a portion of the total time beneath deliberation. Basically for a cyclic event, duty cycle is the ratio of the period of the occurrence to the total stage of a signal. The duty cycle of an event is shown in Figure 6.4.
Based on the Figure 6.4 illustrating duty cycle of an event the below equation is derived.

$$D = \frac{\tau}{T}$$  \hspace{1cm} (6.1)

where, $\tau$ - Duration that the utility is dynamic and $T$ denotes Period of the function.

The objective of a duty cycle data appropriation technique is to decide a proper duty cycle such that the peer nodes involve its own virtual server in the heterogeneous P2P networks. Frequently, heterogeneous P2P network evolves sufficient hope to transmit its data through the relay node and no jamming ought to happen. On the other hand, if the virtual server is active too often, while the peer server node has only a few messages to broadcast, energy is shattered due to over activity by the virtual server of that node.

### 6.3.2 Enhanced Load Balancing Strategy

The load balancing skill in heterogeneous P2P network is enhanced using comprehensive load balancing algorithm. The choice of improving load balancing approach is to utilize the file access record information to determine the outlook file access frequencies. Load balancing algorithm includes two workings namely enhanced load balance algorithm with load
allocation which is employed when a novel peer enters and a load reallocation algorithm which is employed when a peer leaves the structure or one peer turn out to be overloaded.

The principle behind load allocation is to equalize the workload on peer node possesses on the zone unites. The separation of the workload formerly considered on peer node is determined by various issues like the capabilities of two peers, the projected future workload in the zone possessed by peer node and so on. In conventional DCDA systems whenever a peer node arrives, a unique identifier termed as the peer id (peerid) is assigned to each peer node.

Generally, the peer’s distinctive information is gathered for example IP address is taken as the starting point for the creation of peer id. Then the ranking terms are assigned and load diversion strategies are performed. The enhanced load balancing strategy builds the zone separation based on chronological file access information and the capacities of two peers. If any of the two peer node gets congested then subsequent load reallocation algorithm is performed.

Due to peer foliages the workload is congested at the peer nodes. Load reallocation is performed to control the load congestion. File access calculation and data association initiates considerable quantity of overhead. The load reallocation algorithm is performed only when a load diversity or inconsistency occurs. That is, if one or two peers gets congested. The congestion is due to peer foliages the system and also because the descendant is congested or the change of file access behavior causes peer congestion slowly. The load reallocation algorithm is also capable of vigorously achieving the load redeployment through system running time if congested peer node emerges.
6.3.3 Tremendous Store Model

In P2P systems, an enormous number of nodes share the resources and concern queries to each other in network. More frequently nodes contain heterogeneous configurations of storage capacity and processing speed P2P networks. Dynamic characteristic of P2P networks is to join and leave the node. Load balancing in such large-scale and dynamic distributed environments demand heterogeneous environment. The capacity information of all active nodes is gathered before dispatching jobs to the most evenly loaded nodes and is less expensive in ASLB approach.

By applying arbitrary snooping algorithms, the ASLB makes a dispatch decisions based on the load dynamics of a small number of nodes that are selected arbitrarily. In ASLB method, the number of load query messages that are exchanged is condensed considerably. The scalability of arbitrary snooping algorithms is guaranteed because the number of control messages for each decision making is approximately constant even when the system scale expands.

The Tremendous Store model analyzes the effect of e-way arbitrary snooping in load balancing of P2P systems with heterogeneous capacity. But, in practical P2P networks, participant nodes generally have different configurations. The problem with nodal heterogeneity is to tackle the preceding model where the behavior of peer nodes with different capacities is analyzed. In addition, the arbitrary snooping algorithm balance load on P2P network.

Let \( s^* \) denote the maximum values in the sequence

\[
\{S_i\}_{i=1}^M
\]  

Then, the residue capability is calculated as
Thus, value of the situation variables $C_i$ for $0 \leq i \leq s^*$ at time $t = 0$ equals to

$$C_i(0) = \frac{|\{s_i | s_i \leq s^* - i\}|}{M}$$  \hspace{1cm} (6.4)

During the initial load allocation of nodes in heterogeneous P2P network, the residual capacity should not be less than $i$. When the system runs and queries join / leave, the area of residue capability is reserved and load changes in the rest area within $s^*$. With this transformation, peer nodes with heterogeneous capability as $s^*$ are used to model the dynamic system.

**Algorithm of Arbitrary Snooping for ASLB approach**

**Input:** Set of peer heterogeneous nodes ‘N’, virtual servers ‘VS’ duty cycle’ $d_1, d_2, \ldots, d_n$',

$T$: Maximum time needed to process the data

$S^*$ - Maximum values in sequence

**Output:** Reduction of Network Traffic using situation variable ‘$C_i$’

**Begin**

Step 1 : Implement the duty Cycle Adaptation Technique

Step 2 : For Assigning peer node selection

Step 3 : Using DCDA rank method

Step 4 : End For

Step 5 : For Load balancing Strategy

Step 6 : Call function Peer Join (P)

Step 7 : End For

Step 8 : For each Tremendous Store (TS) Model

Step 9 : \[ \{S_i\}_{i=1}^{M} \]

Step 10 : If (Residue capability >Initial Load)

\[ \{S^* - s_i\}_{i=1}^{M} \]

Step 11 : End If
Step 12 :  If time (T=0) \\
Step 13 :  Situation variable $C_i(0) = \frac{|\{S_i : S_i \leq S^* - i\}|}{M}$ computed \\
Step 14 :  End If \\
Step 15 :  End for each \\

End

The above algorithm describes the process of duty cycle adaptation technique and tremendous store model. It describes the process of identifying each peer server node cycle requirements to identify how long the peer server node takes to access the appropriate data in the heterogeneous P2P networks. DCDA technique describes the process of assigning the peer node selection strategy based on ranking representation.

In Tremendous Store Model $\{S_i\}_{i=1}^M$ assigns the maximum value in sequence of P2P network. If residual capability is greater than initial load then $\{S^* - S_i\}_{i=1}^M$ is computed to obtain the load balancing in minimal cost with less traffic. The Situation variable $C_i(0) = \frac{|\{S_i : S_i \leq S^* - i\}|}{M}$ computed when the time equals to zero. The load balancing through DCDA rank method and node selection strategy reduces the traffic. In addition the load allocation and ASLB with tremendous store model balances the load with minimal cost resulting in less traffic congestion.

6.4 EXPERIMENTAL EVALUATION

The experimental evaluation is tested to determine the performance of Arbitrary Selection Load Balancing (ASLB) approach with Tremendous Store model in the NS2 simulator. ASLB with tremendous store model is estimated with different set of nodes and compared with an existing Saturn method (Pitoura et al 2012). The experiments are conducted with the
USC / ISI dataset, network traffic from anarchy online and network traffic dataset. From the dataset, several key characteristics are extracted such as payload sizes, packet charge, data delivery latencies, retransmission statistics, loss rates and stream association.

First, it explains the simulation environment and then provides the simulation results. In simulation, the holdups of intra-transit field links, stub-transit relations and intra-stub domain relations are locate to 100, 20 and 5ms respectively. In all the research, the number of peers in replicated system is 5000. It varies with the number of files in the scheme from 20K to 100K. The proposed arbitrary selection load balancing strategy in heterogeneous P2P network is efficiently designed and implemented in NS-2 simulator.

In addition simulations are carried out to show the effectiveness of the ASLB work with DCDA technique, node selection strategy and load balancing strategy. With the load-demand factor obtained from the peer nodes processing abilities, peer servers processing cycle requirements are identified. Load diversions are made to peer server if the peer node gets overloaded. The diverse data behavior of the command required by the peer servers is exactly identified. A priori of Data Format Load Points are coordinated to present data design insisting of particular nodes. The proposed arbitrary selection load balancing (ASLB) approach with tremendous store model is measured in terms of (i) Peer Failure Probability, (ii) Traffic Control Efficiency and (iii) Execution Time.

6.5 RESULTS AND DISCUSSION

The work elaborates the existence of traffic in the heterogeneous network and a method termed Arbitrary Selection Load Balancing (ASLB) approach with tremendous store model to balance the load in P2P networks. The below table and graph describes the performance of the Arbitrary
Selection Load Balancing (ASLB) Approach with Tremendous Store model. In this consequence, ASLB approach is compared against Saturn method (Pitoura et al 2012).

### 6.5.1 Peer Failure Probability

Each peer fails with a specific probability in heterogeneous environment that is defined as peer failure probability and denoted in percentage (%).

Table 6.1 describes the peer failure probability based on the messages per query on ISI dataset and network traffic from anarchy online dataset. The ASLB approach is compared against Saturn method for analyzing the peer failure probability.

**Table 6.1 No. of Message per Query vs. Peer Failure Probability**

<table>
<thead>
<tr>
<th>No. of Message per Query</th>
<th>Peer Failure Probability (%)</th>
<th>Saturn Method</th>
<th>ASLB approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>14</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>13</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.5 describes the peer failure probability based on the messages per query on ISI dataset and network traffic from anarchy online dataset. Experiments shows that ASLB is capable to accomplish vastly precise results and its performance is normally reliable with less failure in the peer networks about 5-8% compared to Saturn method. As arbitrary snooping algorithm in ASLB method efficiently identifies the traffic and balances the load using the function $C_i(0) = \frac{|\{S_i | S_i \leq S^* - i\}|}{M}$ and capacity accurately in a variety of situations. But Saturn method with query processing in OP-Chord is significantly faster but this result is largely artificial as its performance in terms of recall is significantly lower causing peer failure.

![Figure 6.5 No. of Message per Query vs. Peer Failure Probability](image)

6.5.2 Traffic Control Efficiency

Traffic Control Efficiency is defined as the amount of rate controlled in the network when the traffic occurred in the heterogeneous system.
Table 6.2 describes the traffic control efficiency based on the number of packets transferred from the sender to the receiver. The traffic control efficiency is explained in USC (University of Southern California) / ISI (Information Sciences Institute) dataset, and network traffic dataset. The ASLB approach is compared against Saturn method for analyzing the result of the traffic control efficiency. Based on the Table 6.2 a graph is shown in Figure 6.6.

Table 6.2 No. of Packets vs. Traffic Control Efficiency

<table>
<thead>
<tr>
<th>No. of Packet (1 unit =100 packets)</th>
<th>Traffic control Efficiency (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Saturn Method</td>
</tr>
<tr>
<td></td>
<td>ASLB approach</td>
</tr>
<tr>
<td>1</td>
<td>81</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>79</td>
</tr>
<tr>
<td>4</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>77</td>
</tr>
<tr>
<td>6</td>
<td>76</td>
</tr>
<tr>
<td>7</td>
<td>75</td>
</tr>
<tr>
<td>8</td>
<td>72</td>
</tr>
</tbody>
</table>

Figure 6.6 describes the traffic occurring while transferring the packets in the network traffic dataset. The load is balanced efficiently in the tremendous store model of the ASLB scheme is higher for about 10-15% compared to Saturn method. The Situation variable \( C_i(0) = \left| \left\{ S_i \mid S_i \leq S^d - i \right\} \right| / M \) balanced the load through DCDA rank method and node selection strategy reducing the traffic.
Figure 6.6 No. of Packets vs. Traffic Control Efficiency

In addition the load allocation and ASLB with tremendous store model balances the load with minimal cost resulting in less traffic congestion. Saturn tuned replication costs in different workloads and different network sizes to achieve load balancing increasing the cost and network traffic.

6.5.3 Execution Time

Execution time is defined as the amount of time taken to perform the load balancing factor, estimating the load demand factor, discarding the node congestion and traffic control process in the heterogeneous network. Execution time is measured in seconds (sec).

Table 6.3 describes the execution time based on the number of nodes in the network route used to transform the data from the source to the destination with the anarchy online dataset. The execution time of the Arbitrary Selection Load Balancing (ASLB) Approach with Tremendous Store model is compared against Saturn method to analyze the result.
Table 6.3 No. of Nodes vs. Execution Time

<table>
<thead>
<tr>
<th>No. of Nodes</th>
<th>Execution Time (sec)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Saturn Method</td>
<td>ASLB approach</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>100</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>102</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>105</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>108</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>110</td>
<td>83</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>111</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>113</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>115</td>
<td>108</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.7 describes the execution time based on the number of nodes. Compared to an existing Saturn method, the ASLB consumes lesser time to execute the process and the variance is approximately 20-25 % lesser. The tremendous store model uses the efficient storage process to fetch the results quickly within the short span of time. The lesser time taken consumes a lesser cost in the P2P network to transfer the data from the source to destination. But Saturn method uses extra parameter in order to enhance data availability prolonging the execution time in measuring metrics.
Finally, the Arbitrary Selection Load Balancing (ASLB) Approach with Tremendous Store model balances the load without traffic occurrence in a minimal cost and is proved with the experimental evaluation. The performance metrics like peer failure probability, traffic control efficiency and execution time proved the better performance of ASLB with Tremendous Store model.

6.6 CONCLUSION

The work efficiently achieves load balancing with the reduction in the network traffic of heterogeneous P2P network using Arbitrary Selection Load Balancing (ASLB) Approach with Tremendous Store model. The arbitrary snooping algorithm influence by nodal heterogeneity, capability distribution and agitate on search efficiency is investigated. The arbitrary approach is less sensitive to the node agitation and heterogeneity in terms of the number of exploration conducted before finding suitable nodes and the average response time of queries.

Simulation and experiment results confirm better performance analysis of ASLB approach. The performance of the Arbitrary Selection Load
Balancing (ASLB) Approach with Tremendous Store model reduces the network traffic routes with minimal cost of P2P in contrast to existing Saturn load balancing scheme. An analytical and empirical result offers a collision free system with the quantization of information by balancing the load in the P2P network system. Performance of the Arbitrary Selection Load Balancing is measured in terms of peer failure probability, execution time and traffic control efficiency.