ISSN XXXX XXXX © 2016 IJESC

Research Article Volume 6 Issue No. 12

Applying Load Separation Method in Structured Peer To Peer overlay Networks

S .Venu Gopal[1], N. Sambasiva Rao[2]
JNTUH, Hyderabad[1], JNTUH, Hyderabad[2]
JNUTH, Hyderabad[1], SRIT for Womens, Warangal[2]

Abstract:
A Peer to Peer environment is established when 2 or more computers are linked and share packets without central server. A P2P connection can be ad-hoc in an environment. In a p2p network number of nodes can join and leave the network in a random way. In a P2P network nodes are both clients and servers called workers. Here exchange of information is direct between nodes at the edge of the internet. On top of underlying network environment the connected nodes establish a virtual overlay network. Examples of overlays are CDNs, P2P apps; application-level multicasting etc. the problem in this environment is, if number of nodes is more in a p2p overlay network sharing of content also more. Here, while sharing content may lose due to different reasons. This paper explains why content is going to lose while sharing of packets or content. To know this here I am applying load separation method to structured overlay networks.

Keywords: Peer to Peer, Load, Caching.

1. INTRODUCTION

Difference between structured Vs unstructured P2P.

Structured P2P

The topology of the network is tightly established and files are placed at specified locations. Provide calibrating between the file location and identifier.

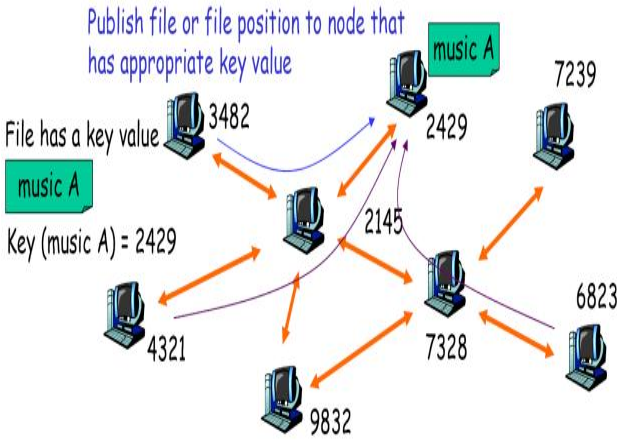

Figure.1. Example for Structured P2P representation

In a structured [1] P2P, there is a concept of fixed connections in the overlay and DHT indexing.

Unstructured P2P

Random data is distributed over the peers and broadcasting approaches are used for searching. In a overlay topology placement of data is unrelated.

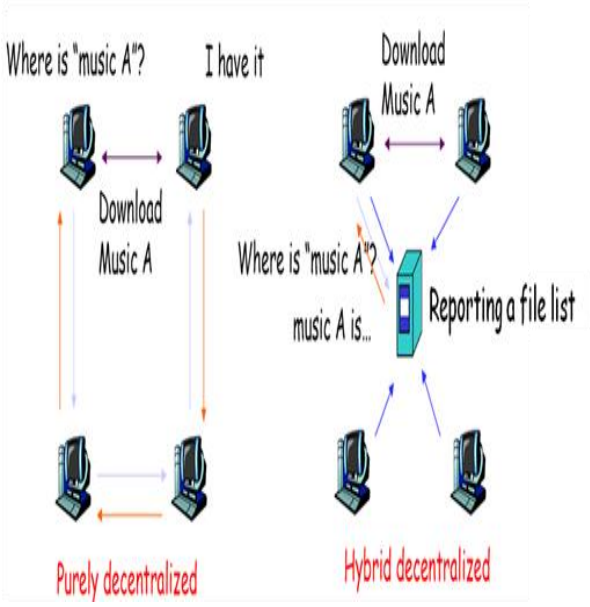

Figure.2: Example for Un Structured P2P representation

In unstructured p2p links are arbitrarily connected between nodes to node. When a node wants to search the file, the request is to be throwing in the network to find as number of nodes as possible that to share requested data. In a P2P[2][3] environment each of the node has same vantage, capabilities and responsibilities. This type of environment is different from client/server environment. In c/s architectures, computers store info and can access to resources, which other systems in network can access through the network. In c/s not required for full time administration. Every individual user act as administrator to his system. User can easily control his machine and resources. Building and maintaining of this type of structured P2P network is very less cost. Here I am taking the one peer to peer network to show the simulation of load separation method.

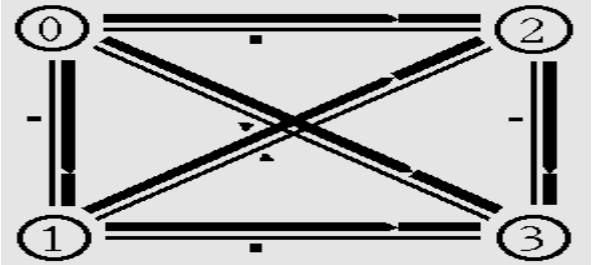

Figure.3: Example for Simulation

Here we consider first number of nodes available in the network[4]. Number nodes means average number of nodes.
Avgn=Average number of nodes in the network
ln=load at every node in the network, *avgln*=calculating average load in the network from_node=No. of packets send from a node, end_node=No. of packets received at node calculating load at every node by calculating load at every node. That is number of packets holding and number of packets sending[5][6] as per the request from the other node. Also finding number requests receiving and number of responses from the node. Here we are considering different events; based on the events we will calculate the above.
Here we consider sample data and we will represent simulation by using graph

From Node	To Node	No. of Pkts			
		Dropped (d)	In Queue(+)	Dequeue (-)	Sent
0	1	45	29306	29306	0
0	2	28	14688	14688	17
1	**2**	**34**	**4684**	**4679**	**34**
1	3	45	29511	29511	0
2	3	45	29511	29511	0
2	**0**	**42**	**14671**	**14671**	**3**
3	2	45	29511	29511	0
3	1	45	29511	29511	0

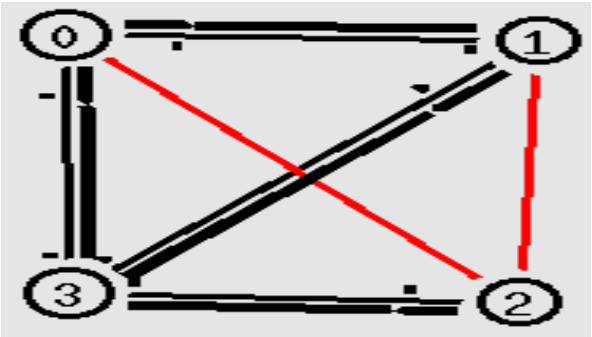

Figure.4: 2 links fails

$ns duplex-link $node(1) $node(2) 100Mb 1ms DropTail
$ns duplex-link $node(1) $node(3) 100Mb 1ms DropTail
$ns duplex-link $node(1) $node(4) 100mb 1ms DropTail
$ns duplex-link $node(2) $node(1) 200mb 1ms DropTail
$ns duplex-link $node(2) $node(3) 200mb 1ms DropTail
$ns duplex-link $node(2) $node(4) 200mb 1ms DropTail
$ns duplex-link $node(3) $node(1) 250mb 1ms DropTail
$ns duplex-link $node(3) $node(2) 250mb 1ms DropTail
$ns duplex-link $node(3) $nod5e(4) 250mb 1ms DropTail.
here we considering every packet size 1024 bits. assuming link failure between node(1) to node(2), node(2) to node(1) also assuming packets[7][8] dropping. after simulation the above table is derived. by considering the above table we will preparing simulation graph with respect to packets sent and dropped between all the nodes[9][10].

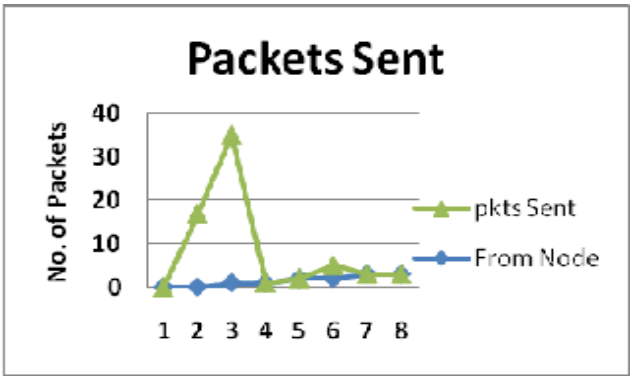

Figure.5: Number of Packets Sent

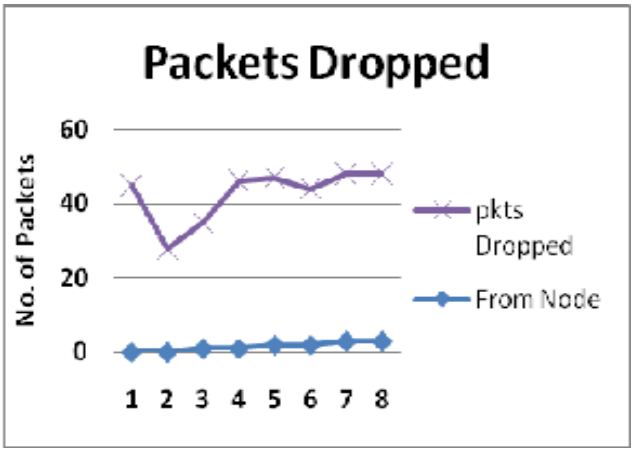

Figure.6: Number of Packets Dropped

Based on the trace file so, here if you observe packets dropping is very high. To avoid this we will find heavy traffic lines. Heavy traffic network [11] [12] lines and connected nodes are separated. We will find heavy loaded nodes are kept separately.
Algorithm: finding heavy loaded peers
Pseudocode for finding heavy loaded peers
Pseudo code for the Peer Selection

Algorithm

n=number requests, enqueue, dequeue
for all MyPeers
if(enqueue>max)
queryHit(n) is very high, heavy load at node
if (enqueue<min)
queueHits is low, queryHit(n) is low, low load at node in the network

Here heavy loaded peer nodes are grouped and low loaded peer nodes are grouped. So, separating low loaded peer nodes from heavy loaded peers.

II. SUMMARY

In this paper the main focus is replication of packets. While replication data may lose due to the above mentioned reason. For that purpose we will apply load separation method by finding heavy loaded nodes from low loaded peers.

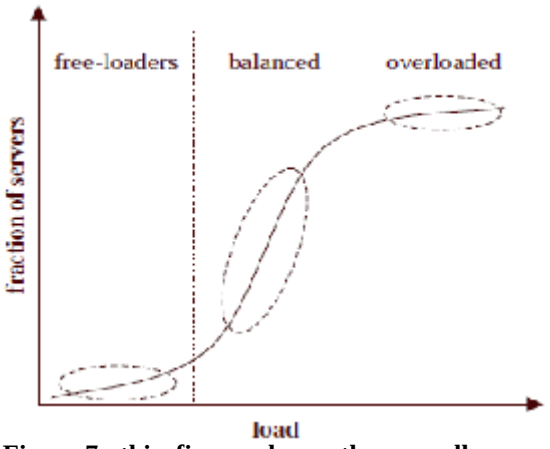

Figure.7. this figure shows the overall my paper that is separating free-loaded, balanced nodes from overloaded nodes. Based on this information we can send and receive more data without any lose.

III. CONCLUSION:

This paper helps to share the data between number of nodes. by applying separation method we can easily find heavy loaded nodes from low loaded nodes. if node is heavy load we can avoid that node as a mediator. so that we can't loss our data.

IV. REFERENCES:

[1] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.

[2] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker, "A scalable content addressable network," in *In Proceedings of the ACM SIGCOMM 2001 Technical Conference*, 2001.

[3] Antony Rowstron and Peter Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, 2001.

[4] B. Zhao, K. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing," Tech. Rep. UCB//CSD-01-1141, University of California at Berkeley Technical Report, 2001.

[5] Antony Rowstron and Peter Druschel, "Storage anagement and caching in PAST, a large-scale, ersistent peer-to-peer storage utility," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, 2001.

[6] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, USA, January 2002.

[7] Pete Keleher, Samrat Bhattacharjee, and Bujor Silaghi, "Are virtualized overlay networks too much of a good thing?," in *The 1st InternationalWorkshop on Peer-to-Peer Systems (IPTPS'02)*, 2002.

[8] David R. Karger, Eric Lehman, Frank Thomson Leighton, Rina Panigrahy, Matthew S. Levine, and Daniel Lewin, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web," in *ACM Symposium on Theory of Computing*, May 1997, pp. 654–663.

[9] "Gnutella home page," http://gnutella.wego.com.

[10] K. Sripanidkulchai, "The popularity of Gnutella queries and its implications on scalability," February 2001.

[11] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of the 16th ACM International Conference on Supercomputing*, New York, USA, June 2002.

[12] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," in *The ACM SIGCOMM'02 Conference*, August 2002.

V. AUTHORS PROFILE:

I, Venu Gopal S, working for Vardhaman College of Engineering in the Department of CSE since 10 years, with M. Tech (CSE), and presently pursuing PhD in CSE at JNTUH, Hyderabad.

I, Dr. N Sambasiva Rao for king for SRITW as Principal at Warangle, with M. Tech & PhD and twenty plus years of teaching experience and 6 years of research experience.