International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

An Effective Routing Algorithm for Load balancing in Unstructured Peer-to-Peer Networks

Dr. Anil V. Turukmane[1], Naresh Tangudu[2], Mr. B. Sreedhar[3], Dr. D. Ganesh[4], P. S. Sagarika Reddy[5], Umamaheswararao Batta[6]

Submitted: 03/10/2023 **Revised**: 21/11/2023 **Accepted**: 01/12/2023

Abstract: Both Academics and industry experts in computer networking are showing a growing interest in peer-to-peer (P2P) networking. In recent years, researchers have attempted to use decentralized peer-to-peer networks to deliver Live Streaming (LS) & Video-on-Demand (VoD). Most of these studies have centered on the properties of the overlaying graph (P2P overlay) that connects the set of interested peers and on the creation of distributed P2P blocks scheduling algorithms for content transfer among the participating peers. Both excessive network traffic and unpredictable delay are significant issues for unstructured p2p systems. Flooding and dynamic query, two of the most used search methods in decentralized p2p networks, are ineffective in dealing with these issues because they lack a heuristic. Existing congestion control designs are not well suited for P2P live streaming traffic, and there has been a paucity of study on the network congestion of these systems. In this work we are proposing an efficient load balancing scheme called Routing Algorithm for Covering Dynamic Time to avoid congestion control for transmitting large amount of data in unstructured Peer-to-peer networks. We are evaluating the proposed efficient Load Balancing mechanismin terms of load distribution rate with virtual servers & performance rate and proved that our proposed scheme works efficiently when compared to existing load balancing schemes.

***Keywords**: Unorganized P2P networks, load distribution, routing, and congestion management*

1.Introduction:

Due to their capacity as a generic framework for widespread applications like sharing of files, data storage, and naming services, organizational P2P networks are currently attracting a lot of attention (see, for example, [1], [2], [3], [4], [5], and [6]). All the nodes and objects in such a network have unique identifiers, and the objects are linked to their respective nodes using hash functions that are consistent across the network. In addition, a structured P2P network keeps track of the topological connections among nodes & provides a routing protocol to find the node that is in charge of the needed object.

[1]Professor, School of Computer Science & Engineering, VIT- AP University, Amaravati, Vijaywada, AP, India
anilturukmane@gmail.com
[2]Associate Professor, Department of IT, Aditya Institute of Technology and Management, Tekkali-532201, A.P, India
itsajs@gmail.com
[3]Assistant professor, Department of CSE Srinivasa Ramanujan Institute of Technology, Anantapur, AP, India
E-mail: sreedhar731@gmail.com
4Associate Professor, Department of CSE, School of Computing, Mohan Babu University (erstwhile Sree Vidyanikethan Engineering College), Tirupati, AP, India.
dgani05@gmail.com
[5]UG Scholar, Department of CSE, Sree Vidyanikethan Engineering College, Tirupathi, AP, India.
sagarikareddypadakula@gmail.com
[6]Department of CSE, KoneruLakshmaiah Education Foundation, Vaddeswaram, AP, India.
umamaheshbatta@gmail.com

An issue with load balancing exists in the structured P2P networks. To begin, the uniform hash function allows for an O(logN) difference in the sum of its stored objects. As a second point, the heterogeneity of P2P systems is not taken into account by the current structured P2P networks. Recent research (e.g., [7]) has revealed that the query pattern and node capacity in real-world P2P systems is imbalanced. This can cause congestion even if there is spare capacity in the network as a whole if nodes' capacities are used disproportionately.

There is a wide variety of paradigms for distributed systems, and Networks are a relatively new addition. P2P systems, an alternative to traditional client-server networks, typically facilitate software that allows users to share and exchange media files such as music, movies, etc. Distributed computing and Internet-based telecommunications are two other areas where this idea has been put to good use. When it comes to file sharing, one of the biggest advantages of P2P systems is their scalability. A node (peer) can take on the roles of either a client or a server. The nodes that are taking part designate some or all of their capabilities as "shared," so that their fellow contributors can use them. For this reason, if node A broadcasts some data &node B obtains it, next node C can obtain the similar data either from node A or node B, depending on which one it contacts first. Therefore, the system's capacity to offer a given file grows as more people use it [8].

For P2P networks, the primary design options are centralized, decentralized structured, &unstructured decentralized (Fig 1). For the centralized model, which is used by services like Napster [9] to hold a directory of file shares kept on peers, index server are used to keep track of where these files are located so that peers can easily find the content they're looking for. However, because to the centralization of the service and the single point of failure it introduces, such a design leaves networks open to DoS assaults. Without needing to rely on a centralized server, users of decentralized P2P systems can instead freely trade resources and information amongst themselves. Network topology and the location of shared data are tightly regulated by distributed hash functions in decentralized structured models like Chord [10], Pastry [11], and CAN [12]. The peer-to-peer (P2P) systems Gnutella [13] &KaZaA [14] operate under are examples of unstructured P2P systems, in which neither the network topology nor the placement of files are strictly regulated. The network is made up of the nodes that have joined it and have agreed to a set of informal rules. The arrangement of items is not informed by the topology, but the final topology has specific qualities [15].

As a result of the network's decentralization, idle resources at its peripheral can be put to good use. This improves scalability and lowers the system's total cost of ownership.

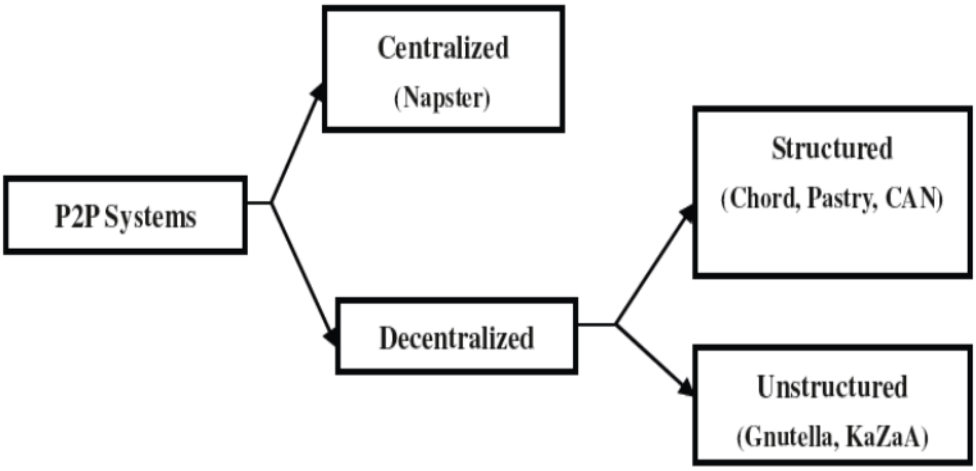

Fig 1. The Varieties of P2P System Architectures

File sharing is the primary function of Gnutella. Gnutella, like other P2P systems, creates a virtual network infrastructure at the software level using its own routing algorithm [16]. Thus, it serves as a decentralized file storage system, letting users specify directories on their local machines that they wish to make available to other peers. Since there is no central authority overseeing network operations, users are free to establish direct connections with one another through a "servant" program that plays the dual role of client and server [17]. The topology of the Gnutella system is depicted in Fig.1 [18] To join an existing network, a servant must first establish a link, or connections, with other nodes. Connections are unstable because nodes often enter and depart the network. Peers address this problem by periodically pinging their neighbors after joining the network in order to learn about other nodes that are part of it. With this data, any node that loses network connectivity can restore its connection. An dynamic, self-organizing system of independent agents [18] is formed when nodes make connection decisions based on just local knowledge. Overload in data networks may be caused by a variety of factors. One of the main sources of network congestion is the spurious retransmission of packets that are already in transit. This violates the packet preservation principle and causes severe congestion by injecting a high number of data sections into the system within one round-trip period. When packets aren't delivered to their intended destination, they take up valuable bandwidth and storage space on the network yet never get there. Congestion collapse happens when there is a mismatch in fragment sizes between the transmission units at the link level. Large quantities of control traffic, including as packet headers, route updates, DNS communications, multicast join and pruning messages, and stable process messages etc., are available on the internet.

Congestion get worse by the enormous proportions of controlled traffic. When there are no dropped packets and an infinite number of available buffers, congestion collapse might nevertheless occur as a result of undesirable or stale data. In the event that packets no longer required by the user were being carried by the system's overloaded links, then congestion breakdown from old packets would occur.

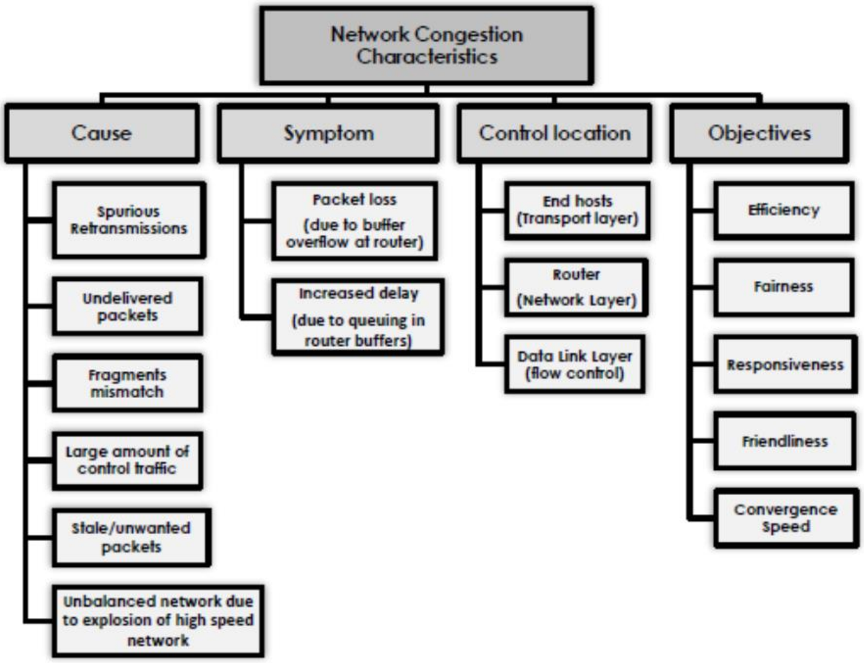

Fig 2 Demonstration of Congestion control characteristics

There are millions of different types of asynchronous networks that make up the Internet, including wired and wireless connections, high-speed connections, mobile ad hoc connections, and even satellite connections. Congestion in the net is a direct result of the speed disparities that result from this diversity. The remainder of the paper is structured as follows:Section2 Mainly describes about the literature survey of various load balancing algorithms and congestion control schemes.Section 3 discusses about the genesis of the problem. Section 4 provides the detailed description about the proposed Methodology i.e. allocation and RAWDT algorithms. Section 5 evaluates the proposed system on various performance factors and proves that the proposed system performs well.Finally section6 concludes the paper.

2. Related Works:

Load balancing systems, in general, have different results in organized and disorganized P2P networks. Structured P2P networks employ DHTs to create connections between the addresses of peers and the resources they have saved. For example, Chord, CAN, and Pas-try are all examples of structured networks that use DHT algorithms [19] to efficiently identify resources using previously known hash keys. However, due to the DHTs' resource partitioning in the identification (ID) space across peers, they incur enormous communication costs under increasing network churn. As a result, structured P2P networks shrink and become less adaptable.

No such supplemental space partition mechanism is needed in unstructured P2P networks. This allows for more adaptability and scalability in unstructured P2P networks. Unstructured P2P networks, on the other hand, have difficulty discovering resources due to the fact that queries can only be routed to peers in close proximity. Also, load balancing problems between peers are common in advanced resource finding techniques like flooding [20] & random walk [21]. Improving the efficiency of a popular search method called random-walk sample [22] and providing more guidance to the search strategies through data sets [23,24] are two potential approaches to this issue[43].

Unstructured P2P networks have problems with uneven workloads, however load balancing techniques can be used to efficiently even things out. Some studies [25,26,40] employed su- per peers to manage network-wide queries being broadcast. The loss of these super peers, however, could cause a domino effect and reduce the network's scalability [27]. Finding the popular files that are freeloaders over their peers can turn a flat networks into a star network [28,29] and aid to dynamically modify the desired network overlay topologies. In some of the most recent publications, we see the use of such techniques for dynamically adjusting the network topologies. DANTE is a network introduced by Merino et al. [30] that creates a symmetrical cluster overlay topology through a self-organized reconnection mechanism. The improved reconfiguration gnutella overlay (ERGO) network, suggested by Pournaras et al. [31], employs a rewiring method based on the use of practical servers to track the load of the peers. Overwhelmed peers in a system can have their incoming connections redirected to less-active nodes thanks to ERGO. Another network, called AVMON [32][41]

(Availability Monitoring Overlay), keeps an eye on what peers in the network are sharing and how they're sharing it. To combine search and trust data and route queries to the most reliable peers, Mashayekhi& Habibi [33] [42]designed a system that maintains restricted size routing indexes. The learned automata-based resources discovery [34] network is also an unstructured P2P network, with nodes deciding on their own which communication link to use to send a query to the providers of the resources they need. The method awards the picked route if its length is shorter than the median extent of the already chosen routes, and penalizes it otherwise. As a result, our algorithm ultimately settles on the path with the shortest predicted duration [35,36,37].

Meruguet al.[38] &Wu et al.[39] introduce the methods of employing small-world overlay topologies. Short links were employed by Merugus et al. [38] to connect more closely-situated peers, while longer linkages were used to link peers that were selected at random. Somewhat similarly, The state-based shortcut listing system created by Wu et al. [39]. In this approach, you can use the associated connections or the state data for a more effective search.

3. Problem Formulation:

Unstructured P2P networks have no governing authority and no clear overlay topology. The links between nodes are established at random. Nevertheless, as the graph develops, long-lived nodes acquire more connections than newer ones do because they are more accessible. This is because the degree distribution is significantly skewed, with just a small subset of nodes having a large number of neighbors. Despite having some positive characteristics, such as a small diameter and resistance to higher failure, rapid node arrivals and departures, and congestion caused by inefficient load balancing algorithms, these networks are not without their drawbacks. Furthermore, they have low topology maintenance costs and are easy to install.

4. Proposed System:

Node with Time-Varying Peers For effective load balancing in dispersed heterogeneous peer networks, one solution is to have the peer node leave the network, as is done in the Heterogeneous Data Processing scheme. The network's peer node's vitality must be indicated. In a diverse Peer to peer network, Routing Algorithm for Covering Dynamic Time [RAWDT] is employed to track the local time of each peer node. DWA measures the size of load - balancing factors of network participants by analyzing their dynamic time variation capacity. DWA can quickly and accurately determine whether or not a peer node in a network is currently active. The basic validation of the dynamic time is used as a foundation for the peer node studies, so the nodes are examined before distribution.

In a heterogeneous P2P network, a peer node will first use DWA to determine the DT (Dynamic Time) of its adjacent peer node before exchanging its virtual nodes with the other nodes. Based on the results of the analysis, the packet data is kept in common across multiple virtual nodes in order to balance out the load imbalance. DWA is used to assess the amount of load-demand balance elements of peer nodes by measuring their time variant capacity. Using the load-demand factor derived from DWA, the Duty Cycle Data Acquisition (DCDA) approach may determine what kind of processing cycles are needed by peer nodes. In terms of the additional money needed to run virtual nodes and the load imbalance factor, DCDA is superior than alternative solutions. Distributed heterogeneity peer networks can benefit greatly from the load balancing provided by the Heterogeneous Data Processing Scheme.

4.1 Reallocation Algorithm:

Let's pretend that a DHT's total hash space is [0, 1], & that every DHT server has an ID picked arbitrarily but consistently from [0, 1]. In a DHT, the collection of peers P is the group of users, and group of virtual servers S is the group of users that act as nodes. Let's call the collection of virtual servers Si. With the use of Equation (1), each peer I in P figures out the load L that it should acknowledge, denoted by Ti, where Ti = ~A*C_i +, where A is an estimate of predicted load per units with a capability 'C,' and so on.

$$\sum_{s \in S} {}^{L_s}/_{C_i} \text{ -------------(1)}$$

where ε is a predefined operator

If I is now experiencing a total load in excess of Ti (that is, if I is overflowing), Then I'll move S to some other peers, i. Otherwise, I will do nothing and continue to believe the migrant virtual server S. If my peer I were to become overworked, I would choose to reassign some of my virtual servers to group S) I becomes under loaded& 2) the entire transfer cost, MC, is reduced. If peer I is not overloaded, it is required to accept the virtual server Si, and it does if the additional burden imposed by Si does not cause I to become overloaded.

Algorithm (Reallocation (i))

Input: P, S, i, T_i

Output: allotment of S_i to P_i

Switch (Load (i)) do

Case > Ti

Ui $<>$ 0

While Load (i) $>T_i$ and Si $<>$ U_i do

S $\leftarrow$arg min { l_v | v ε $S_i <> U_i$ }

Break;

Case $< T_i$

While Load (i) $<T_i$ do

Get s from the host.

$S_i \leftarrow S_i$ U {s}

Break;

End

When using a P2P network, however, it is impossible to tell if a peer node will remain stable or not. Considering that a peer node could disappear from the network at any time, it's important to mark the peer node's time. Dynamic Time Warping is a technique used to time-stamp all the P2P nodes so you know when they've updated, regardless of how they may be time-zoned.

4.2. Routing Algorithm for Covering Dynamic Time[RAWDT]:

The RAWDT measures the size of load - balancing factors of peer servers by analyzing their dynamic time variation capacity. Figure 3 is a schematic depiction of the dynamic warping technique used in time-variant peer-to-peer networks. This explains the Routing Algorithm for Covering Dynamic Time in Unstructured Peer-to-Peer Networks. With a VS assigned to each node, a reallocation procedure is run, and then RAWDT is executed if the load on the nodes grows. In this section, we will look at the RAWDT, which may be used to calculate a peer node's time-variant capacity:

int RAWDT time variation amount (C (p_1), C (p_2)…. C (p_n), int w)

{

DeclareRAWDT (C(p1),C(p2)…..C(p_n))

Intw:MAX-time(C(p_i)),i,j

{

Fori= 1ton

EstimatewforC(p_i)

EndFor

Forj= 1ton

Do

Algorithm(Reallocation(j))

While(wfor(C(p_j)))

End do

End For

}

}

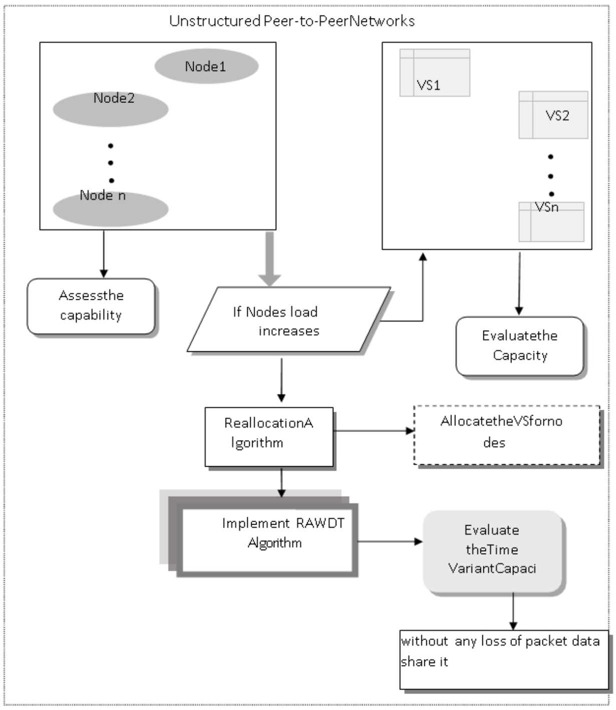

Fig 3 Proposed Architecture of Unstructured Peer-2-Peer networks implementing Routing Algorithm for Covering Dynamic Time

5. Experimental Evaluation and Results:

Here, we conduct simulation experiments with various network configurations typical of unstructured P2P networks in order to evaluate the efficacy of the RAWDT Load balancing Algorithm, the Adaptive load balancing(ALB) technique algorithm, and the Weighted Response Time Algorithm for Web Application Load Balancing(WRTA) [37,38]. Simulations run in real time with the help of the NS-2 Network Simulator are used to assess how well the RAWDT technique works. We start with a rundown of the simulation's setup, then define some performance measurements, and finish off with displaying some sample output from the simulation.

5.1 Simulation Setup

In our experiments, we used networks with a certain amount of nodes that were spread randomly over a 500x500 area. There are 25, 50,..., 250 nodes. At first, the nodes are placed randomly in the given spot. In Table 1, you can see what simulation parameters were used.

Table 1: Elements of Simulation with its Description

S.No.	**Parameter**	**Description**
1.	Network Area	500 x 500 m^2
2.	Cognitive Radio Nodes	250
3.	Data Packet size	2500 bytes
4.	Bandwidth of each channel	1 Mbps
5.	Available Spectrum Bandwidth	36 MHz-48 MHz
6.	MAC Layer standard	IEEE 802.11
7.	No of Cycle Simulations	10
8.	Traffic Type	Constant bit rate flow
9.	Simulation time	500 seconds
10.	Data Packet Generation	5 Packet/sec
11.	Nodes Distribution	Random Distribution

5.2 Performance Metrics& Simulation Results:

Two separate performance indicators are presented in this section for Evaluation. They are

(i)Load Distribution Rate (%):

Load imbalance factor, which characterizes the rate of load distribution, is the performance parameter employed here to enhance heterogeneous P2P networks. Node I out of node N has a Load Imbalance if and only if

Load Distribution Rate $=\sum_{v\in V_i} {}^{L_v}/_{C_i}$ ----------------(1)

The rate of load distribution is shown in relation to the total number of nodes in Table 2. A time-varying peer heterogeneity P2P network is proposed, RAWDT keeps the load unbalance factor of the nodes constant even as the number of nodes grows. Table 2 is used to create a comparison graph, which is seen in Figure 4.

Table 2. Simulation results of Load Distribution Rate

Number of Nodes	Load Distribution Rate (%)		
	RAWDT	ALB	WRTA
25	89	79	75
50	82	74	70
75	75	69	64
100	71	61	58
125	69	57	52
150	64	52	48
175	61	46	42
200	58	41	38
225	52	39	32
250	48	34	27

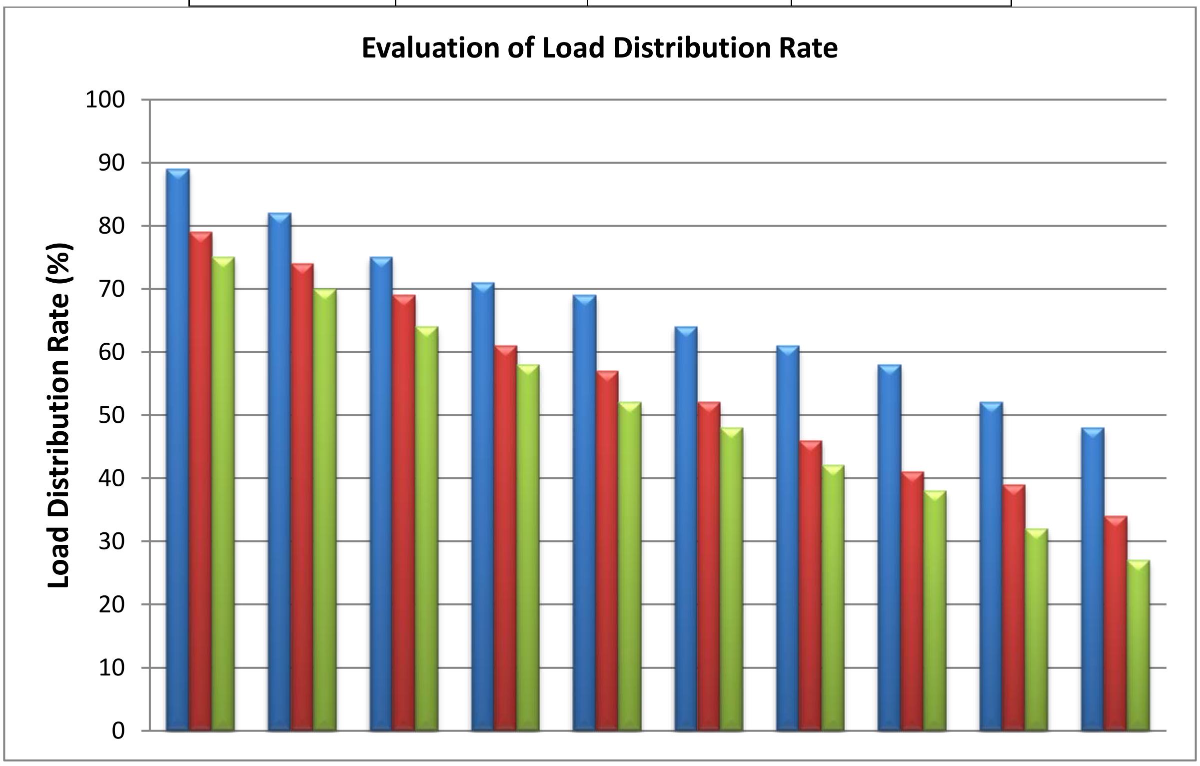

Fig 4. Comarison of Load Distribution Rate with Existing Algorithms

Figure 4 shows how the network's load is distributed across nodes across different simulation iterations. The suggested technique is around 8-12% more effective at keeping the node's capacity constant when compared to other Load Balancing algorithms developed for use in unstructured P2P networks. In the proposed time-variant P2P network, the virtual servers used to store packet data are pooled such that the workload imbalance factor is minimized.

(ii) Performance Rate

The improved load balancing efficiency provided by RAWDT is used to determine rate of performance in a time-varying P2P system. The capacity for load balancing contributes to the system's excellence as an unstructured P2P network. The performance rate according to time spent in dynamic peer nodes is shown in Table 3. Diagrammatic representation of data from Table 3

Table 3. Simulation results of Performance Rate

Number of Nodes	Performance Rate (%)		
	RAWDT	ALB	WRTA
25	96	89	82
50	94	87	78
75	91	82	74
100	87	80	71
125	84	78	69
150	81	74	65
175	79	71	60
200	77	69	57
225	75	64	54
250	73	62	51

Figure 5 shows how fast peer entities in the network work based on how time changes. The suggested RAWDT-based time-varying P2P network, if the dynamic peer endpoint time goes up, the peer node's performance rate goes up by about the same amount. The suggested approach improves performance by 12-16% compared to a state-of-the-art Load Balance for Inherently incorrect Data in Un-Structured Peer-to-Peer Networks. As shown in Figure 5, RAWDT does a good job of figuring out whether a peer base station that shares its virtualized resources with another node is still alive by checking the Dynamic Time (DT) of the neighboring peer node. This gives RAWDT a better performance rate than the other algorithms.

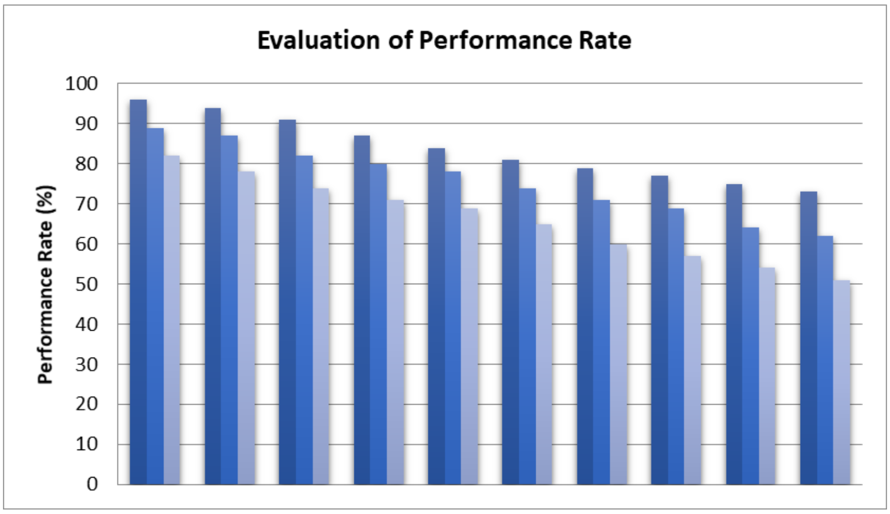

Fig 5. Comarison of Performance Rate with Existing Algorithms

6. Conclusion:

An unstructured P2P network with time-varying peers was shown utilizing RAWDT and virtual servers in this paper. The use of probability distributions to depict the system condition sets the proposed system apart. Unlike prior systems, which often rely on complete system knowledge, our proposed solution has each peer independently estimate the cumulative load of stranded virtual machines and the probability distributions function of neighbor contributions. Every node is able to tell if it is overloaded or under loaded thanks to the estimated probabilistic, and subsequently rebalances the loads among itself as needed. Test findings show that the proposed system surpasses the state-of-the-art Load Balancing methods in Unstructured P2P Systems, both in terms of the rate at which load is distributed across virtual servers and the overall performance of the system.

References:

[1] S. Rantnasamy,P. Francis,M. Handley,R. Karp and S.Shenker. A Scalable Content-Addressable Network. In *ACMSIGCOMM*,2016.

[2] I.Stoica,R.Morris, D.Karger, M.F.Kaashoek and H.Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup ServiceforInternetApplications.In*ACMSIGCOMM*, 2016.

[3] A.RowstronandP.Druschel. Pastry:Scalable,Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems. In proceedings of the 2nd IFIP/ACM International Middleware Conference, 2017.

[4] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph and J. Kubiatowicz. Tapestry: An Infrastructure for Fault Tol- erant Wide-area Location and Routing. IEEE Journal on Selected Areas in Communications, 12(1):41-53, 2014.

[5] F. Kaashoek and D. Karger. Koorde: A simple degree-optimal distributed hash table. In proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS), 2018.

[6] H. Shen, C. Xu, and G. Chen. Cycloid: A Constant-Degree and Lookup-Efficient P2P Overlay Network. In proceedings of the 18th IEEE International Parallel and Distributed Pro- cessing Symposium (IPDPS), 2018.

[7] S. Saroiu, P. K. Gummadi and S. D. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. In proceedings of 9th Multimedia Computing and Networking (MMCN), 2018.

[8] Saurabh Tewari, Performance Study of Peer-to-Peer File Sharing, Ph.D Thesis, University of California, Los Angeles, 2017.

[9] Kim and L. Hoffman, “Napster and other Internet peer-to-peer applications,” George Washington University, available: citeseer.ist.psu.edu/kim01pricing.html,2015.

[10] Stoica, Morris, R. Karger, D. Kaashoek, M and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” Proc. of SIGCOMM 2017.

[11] Rowston and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” Proc. of IFIP / ACM Middleware, Heidelberg, Germany, 2011.

[12] S. Ratnasamy, P. Francis, M. Handley and R .Karp, “A scalable content-addressable network,” In Proc. Of SIGCOMM, 2011.

[13] Ganesh, Davanam, Thummala Pavan Kumar, and Malchi Sunil Kumar. "Optimised Levenshtein centroid cross-layer defence for multi-hop cognitive radio networks." *IET Communications* 15.2 (2021): 245-256.

[14] Davanam, Ganesh, T. Pavan Kumar, and M. Sunil Kumar. "Novel Defense Framework for Cross-layer Attacks in Cognitive Radio Networks." *International Conference on Intelligent and Smart Computing in Data Analytics: ISCDA 2020*. Springer Singapore, 2021.

[15] Davanam, Ganesh, T. Pavan Kumar, and M. Sunil Kumar. "Efficient energy management for reducing cross layer attacks in cognitive radio networks." Journal of Green Engineering 11 (2021): 1412-1426.

[16] Davanam, Ganesh, et al. "Multi-Controller Model for Improving the Performance of IoT Networks." *Energies* 15.22 (2022): 8738.

[17] Stephanos Androutsellis-Theotokis, “White Paper-A Survey of Peer-to-Peer File Sharing Technologies,”

[18] Ganesh, Davanam, T. Pavan Kumar, and M. Sunil Kumar. "A Dynamic and adaptive learning mechanism to reduce cross layer attacks in cogntive networks." *Materials Today: Proceedings* (2020).

[19] Kumar, M. Sunil, and D. Harshitha. "Process innovation methods on business process

reengineering." *Int. J. Innov. Technol. Explor. Eng* (2019).

[20] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and replication in unstructured peer-to-peer networks,” in Proc. 16th ACM Conf. Supercomput., 2017, pp. 84–95.

[21] Balaji, K., P. Sai Kiran, and M. Sunil Kumar. "Resource aware virtual machine placement in IaaS cloud using bio-inspired firefly algorithm." *Journal of Green Engineering* 10 (2020): 9315-9327.

[22] A. D. Sarma, A. R. Molla, and G. Pandurangan, “Efficient random walk sampling in distributed networks,” J. Parallel Distrib. Comput., vol. 77, pp. 84–94, 2018.

[23] D. Tsoumakos and N. Roussopoulos, “Adaptive probabilistic search for peer-to-peer networkss,” in Proc. 3rd Int. Conf. Peer-to-Peer Comput., 2013, pp. 102–109.

[24] R. L. Pereira, T.Vaza¨o, and R. Rodrigues, “Adaptive search radius— Using hop count to reduce P2P traffic,” Comput. Netw., vol. 56, no. 2, pp. 642–660, 2017.

[25] A. Kurve, C. Griffin, D. J. Miller, and G. Kesidis, “Optimizing cluster formation in super-peer networks via local incentive design,” Peer-to-Peer Netw. Appl., vol. 8, no. 1, pp. 1–21, 2015.

[26] J.-S. Li and C.-H. Chao, “An efficient super-peer overlay construction and broadcasting scheme based on perfect difference graph,” IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 5, pp. 594–606, May 2018.

[27] G. J. Fakasa and B. Karakostas, “An efficient super-peer overlay construc- tion and broadcasting scheme based on perfect difference graph,” Inform. Softw. Technol., vol. 46, no. 6, pp. 423–431, 2014.

[28] Z. Xu and L. N. Bhuyan, “Effective load balancing in P2P systems,” in Proc. IEEE Int. Symp. Cluster Comput. Grid, 2006, pp. 81–88.

[29] G. Sreenu, P. M. Dhanya, and S. M. Thampi, “Enhancement of BATER- CAST using reinforcement learning to effectively manage free riders,” in Advances in Computing and Communications. New York, NY, USA: Springer, 2011, pp. 126–136.

[30] L. R. Merino, A. F. Anta, L. Lo´pze, and V. Cholvi, “Self-managed topolo- gies in P2P networks,” Comput. Netw., vol. 53, no. 10, pp. 1722–1736, 2009

[31] E. Pournaras, G. Exarchakos, and N. Antonopoulos, “Load-driven neighbourhood reconfiguration of Gnutella overlay,” Comput. Commun., vol. 31, no. 13, pp. 3030–3039, 2018

[32] R. Morales and I. Gupta, “Avmon: Optimal and scalable discovery of consistent availability monitoring overlays for distributed systems,” IEEE Trans. Parallel Distrib. Syst., vol. 20, no. 4, pp. 446–459, Apr. 2019.

[33] Burada, S., Swamy, B. E., & Kumar, M. S. (2022). Computer-Aided Diagnosis Mechanism for Melanoma Skin Cancer Detection Using Radial Basis Function Network. In Proceedings of the International Conference on Cognitive and Intelligent Computing (pp. 619-628). Springer, Singapore.

[34] Prasad, T. G., Turukmane, A. V., Kumar, M. S., Madhavi, N. B., Sushama, C., & Neelima, P. (2022). Cnn Based Pathway Control To Prevent Covid Spread Using Face Mask And Body Temperature Detection. Journal of Pharmaceutical Negative Results, 1374-1381.

[35] AnanthaNatarajan, V., M. Sunil Kumar, and V. Tamizhazhagan. "Forecasting of Wind Power using LSTM Recurrent Neural Network." Journal of Green Engineering 10 (2020).

[36] H. Mashayekhi and J. Habibi, “Combining search and trust models in unstructured peer-to-peer networks,” J. Supercomput., vol. 53, no. 1, pp. 65–85, 2018.

[37] J. A. Torkestani, “A distributed resource discovery algorithm for P2P grids,” J. Netw. Comput. Appl., vol. 35, no. 6, pp. 2028–2036, 2017.

[38] S. Merugu, S. Srinivasan, and E. Zegura, “Adding structure to unstructured peer-to-peer networks: The use of small-world graphs,” J. Parallel Distrib. Comput., vol. 65, no. 2, pp. 142–153, 2015.

[39] K. Wu, C. Wu, and L. Liu, “State-based search strategy in unstructured P2P,” in Proc. IEEE 13th Int. Symp. Object/Component/Service-Oriented Real-Time Distrib. Comput., 2017, pp. 381–386.

[40] Natarajan, V. A., Kumar, M. S., Tamizhazhagan, V., & Chevdumoi, R. M. (2022). PREDICTION OF SOIL PH FROM REMOTE SENSING DATA USING GRADIENT BOOSTED REGRESSION ANALYSIS. Journal of Pharmaceutical Negative Results, 29-36.

[41] Sunil Kumar, M., & Rama Mohan Reddy, A. (2016). An Efficient Approach for Evolution of Functional Requirements to Improve the Quality of Software Architecture. In Artificial Intelligence and Evolutionary Computations in Engineering Systems

(pp. 775-792). Springer, New Delhi.

[42] Park, G. et al. (2021). Adaptive Load Balancing Mechanism for Server Cluster. In: , et al. Computational Science and Its Applications - ICCSA 2021. Lecture Notes in Computer Science, vol 3983. Springer, Berlin, Heidelberg

[43] H. Nurwasito and R. Rahmawati, "Weighted Response Time Algorithm for Web Server Load Balancing in Software Defined Networks," 2021 3rd International Conference on Electronics Representation and Algorithm (ICERA), Yogyakarta, Indonesia, 2021, pp. 143-148, doi: 10.1109/ICERA53111.2021.9538792.