Peer-to-Peer Networking and Applications (2022) 15:666–688
https://doi.org/10.1007/s12083-021-01259-3

An optimal load balancing strategy for P2P network using chicken swarm optimization

Dharmendra Kumar[1] · **Mayank Pandey**[2]

Received: 27 May 2021 / Accepted: 28 October 2021 / Published online: 8 January 2022

Abstract

Peer-to-Peer (P2P) networks are less expensive, simple to use, and do not require the traditional client–server model. It has particular advantages in data sharing and resource utilization, so it is recommended to use it for various applications. P2P networks have been used in many applications, especially in data sharing and resource utilization. Load balancing and security is an essential task to improve the performance of P2P networks. Hence, in this paper, probability-based load balancing control and security enhancement is developed in P2P networks. The probability of peer can be computed with chicken swarm optimization (CSO), which selects the best peer in P2P networks to achieve load balancing and resource utilization. The proposed method is developed to attain two main objective functions: load balancing control and security enhancement. A probability-based CSO algorithm is used to control load balancing. The security is achieved with Enhanced Rumour Riding protocol (ERR) and SXOR (Split XOR) operation. The proposed method is implemented in the NS2 platform, and the performance of the proposed method is analysed with performance metrics such as delay, delivery ratio, packet loss, encryption time, decryption time, and throughput. The proposed method is compared with existing methods such as Biased Contribution Index based Rumour Riding protocol (BCIRR), Ant Colony Optimization (ACO), and Catching Algorithms (CA). The proposed technique achieves a 98.75% packet delivery ratio, with a minimum 3.8 s delay. Ultimately the performance suggests that the proposed system can perform better for load balancing and security in the P2P network.

Keywords Probability · Security · Load balancing control · P2P networks · CSO

Abbreviations

P	Peer
nr	Best fitness
nc	Worst fitness
f	Fitness value
X	Position
k	Rooster index
ε	Small constant value
R2 and *R1*	Index value of chicken and hens groupmate
Random	Random number over [0, 1]
$X^T_{s,b}$	Position of chick's mother
fl	Random value between 0 and 2

1 Introduction

Recently, the P2P network has emerged as a solution for resource sharing and resource discovery in the network [1]. Many P2P networks have been used to meet a wide range of applications, including "Skype, BitTorrent and Gnutella". The typical client-server model has a centralized model that includes service requests and service providers. The server plays a crucial role in processing requests from a customer and managing shared resources, and all resources are centralized on the server [2]. Different network structures have been developed to share social networks, sensor networks and the Internet of Things (IoT). P2P networks are the most significant to store and search large amounts of data [3]. Users can identify media files such as music, books, games, and movies utilizing the P2P file-sharing software programme. The software program's job is to probe for other computers linked to a

✉ Dharmendra Kumar
kumar.dharmendra@rediffmail.com

Mayank Pandey
mayankpandey@mnnit.ac.in

[1] Department of Computer Science and Engineering, Dr. A.P. J Abdul Kalam Technical University, Jankipuram, Lucknow, Uttar Pradesh 226031, India

[2] Department of Computer Science & Engineering, Motilal Nehru National Institute of Technology, Teliarganj, Prayagraj, Uttar Pradesh 211004, India

peer-to-peer network to find the required content. End-utilizer computer systems that are cumulated through the Internet are the peers or nodes of such networks. In the P2P network, each partner acts as a client and server model for communication or data sharing by different peers. The P2P network can be considered advanced because it offers greater access and scalability than the traditional client–server model [4].

A load accumulates at a specific node in a P2P network system when a virus or a specific data event occurs. Having too much load on a particular node of a P2P network can reduce response time and network performance. Dynamic load conditions occur during data transfer, download and upload, which reduces system performance and response time [5]. There is no special tip in P2P to balance the load between peers. Therefore, each node focuses on reducing loads to increase performance [6, 7]. Each node seeks to reduce their load to load their own by avoiding loads from other nodes. Due to this trend, low participation rates in the load balance are a problem [8]. Similarly, security in the P2P network is a significant concern because they can resist attacks such as service denial. After the attack, some of the incorrect participants may have flooded the P2P network, thus blocking proper traffic [9, 10].

The researcher has developed various techniques to improve load balance and safety improvement in P2P networks [11]. The most basic load balancing strategy is to relocate severely laden nodes to less loaded nodes and then redistribute the load across the nodes. In a peer-to-peer system, however, this load balancing strategy is far from simple. In safety improvement, the authentication system verifies the identity or other qualities to a network usage to assure data security during its broadcast from the source node to the endpoint. Previously developed techniques have been taken into account to identify performance drives for analysing performance, balancing loads, and increasing security levels in the P2P network [12, 13]. Researchers focus only on the security of load balance or P2P networks. Both objectives are considered a vital issue in the P2P network, which is to improve system performance. A unique approach previously developed is a stimulus mechanism used to balance loads of other peers [14]. However, this stimulus mechanism encouraged the nodes to contribute the load balancing rather than the forced balance. The dependent contribution table (PCI) [15] was developed to balance loads on P2P networks, which works efficiently, but it is challenging to balance loads between peers if they have many peers.

Random choice algorithms are scalable, with few control messages and data structures required. They work in P2P networks with churn, which is a circumstance in which a substantial percentage of nodes join and leave often, resulting in capricious network size. Similarly, each technique previously developed does not provide efficient load balance control and does not focus on the security level in the P2P network configuration. The concentration of load balance control and safety enhancement is considered an essential task in the P2P network. A new method has been developed to achieve load balance control and security improvement in P2P networks, providing better results.

1.1 Motivation of the work

P2P network is utilized for several applications like sharing files and resource utilization. Many peers are connected to compose a P2P network. A single peer sends a query to another peer for amassing requisite resources. Due to the immensely colossal number of load in a single peer, the replication time and resource utilization time are gradual. Moreover, malevolent peer and assail are additionally possible in this network. To eschew these quandaries, need to amend the security and balance the load in each peer. Sundry techniques are developed to amend security and load balancing individually, yet, the quandary is both load balancing and security enhancement. So an advanced method is preferred to amend both load balancing and security in a P2P network that controls the load in each peer while relocating to another peer or balance the loads, as well as improve the security to recognize the attack node and malicious peer.

1.2 Main contribution and organisation of the paper

This paper has developed probability-based load balance control and safety enhancement in P2P networks via the CSO algorithm. The main contribution of the paper is given as follows:

- Achieving load balancing and high security in a peer-to-peer network using CSO algorithm and ERR, and SXOR operation respectively.
- P2P network is used for various applications like data sharing and resource utilization. A large number of load in a particular node of the P2P network reduce the response time and network performance so a load of each node in P2P network should be balanced as well as improve the security.
- Based on probability conditions, the proposed CSO algorithm is utilizing to control the load by selecting the best peer in a P2P network.
- In addition, security is enhanced via ERR and SXOR operations. ERR find the attacker nodes, and SXOR store the resources securely. These security systems analyse the node very well as well as avoid the unauthorised person access data from a data store.
- The proposed method performance can be analysed with performance metrics such as delay, packet loss, throughput, delivery ratio, encryption time and decryption time. The proposed method is compared with existing methods such as BCIRR, ACO and CA methods.

The remaining part of the article is organised as follows, related works based on load balancing control and security enhancement are reviewed in Sect. 2. The proposed architecture is explained in Sect. 3, which consists of a detailed description of the proposed load balancing control and security methods. A detailed description of probability-based load balancing control and CSO algorithm progress is given in Sect. 4. ERR and SXOR operation procedures are also presented in this section. The proposed method performance evaluation is analysed, and simulation results are presented in Sect. 4. Finally, Sect. 5 provides the paper conclusion part.

2 Literature survey

In P2P networks, load balancing and security improvement are seen as critical factors for improving resource utilization and system performance. Many researchers focus on improving load balance control and safety individually, but no one focuses on both needs. Some of the latest research on load balancing control or security enhancement of P2P networks is reviewed in this section.

2.1 Survey-based on structured P2P networks

In structured P2P networks, the data is placed on adequately defined platforms, and the distributed routing table shows the mapping between the data and their locations. Singh et al. [16] have developed a finger forwarding mechanism to minimise the search path length of resources. The finger forwarding mechanism reduces the load balancing problem. Ref. [17] proposed a hierarchical structure for capacity management in the P2P network. The hierarchical structure includes tasks and dedicated roles. In Ref. [18], some noticeable problems in 5G networks like large-scale mobile terminals, dynamic topology and resource management were rectified by developing a network load balancing. However, the method does not fully solve the edge computation problem like the problem of 5G network server and service allocation between edge computing, the efficiency of server-side tasks and the efficiency of the communication of edge computing tasks.

Rahmani and Benchaiba [19] have developed a Multihop Proximity aware Clustering Technique (PCSM) to improve security conditions in Mobile Peer to Peer (MP2P) systems. This technique has been shown to reduce peer physical proximity behavior and mismatch in the P2P overlay and display in the network layer. Cluster head, cluster size, and several physical fire hops were considered three critical factors for effectively joining a new client clustering. This technique has been implemented to achieve maintenance in peer movement. The PCSM was tested by topology fits in MANET, which applied to manage underlay, which provides better results from the existing cluster-based P2P overlay. Yet, the data in secondary CH is difficult to maintain up-to-date.

2.2 Survey-based on unstructured P2P networks

In this network type, the overlay links are usually established arbitrarily. In Ref. [20], a hierarchically distributed peer-to-peer architecture was proposed to overcome all significant problems encountered due to virtuality and heterogeneity in grid computing. In Ref. [21], the authors have presented a systematic literature review. In which the authors revised the so far literature for load balancing and listed its advantages and disadvantages. In Ref. [22], a software-defined network has been proposed to solve the network management issue and minimise the load balancing problem. To reduce the traffic cost of Content Delivery Network (CDN) providers, P2P -assisted DASH system was proposed in Ref. [23], and a peer selection algorithm that maximally reduces CDN provides' traffic cost was investigated for this system. Authors in Ref. [24] highlighted few drawbacks in P2P networks based on the review conducted on four groups such as structured, unstructured, super-peer, and hybrid networks.

Shen et al. [25] have developed two theoretical Locality Sensitive Hashing (LSH) based data delivery models to improve the load balancing scenario in P2P networks. Heterogeneous and homogeneous datasets were utilized to implement this method. The method's thrust was previously that only a single hash table-based load balancing control had done so, and multiple hash tables-based load balancing control was concentrated in this research. An index mapping technique was developed with a standard distribution indexing method to promote load balance control in P2P networks with the aggregate distribution. The standard indexing system has been transformed into robust and practical, with the developed technique incorporating the virtual node mechanism of P2P networks. The developed method has been tested with natural and synthetic datasets, and it yields better results for load balancing in the P2P network.

2.3 Survey-based on other P2P networks

Qi et al. [26] have developed a balanced replication technique to improve fault tolerance under high churn in P2P networks. The balanced replication technique was operating related to Zones partition-BRBZs containing three mechanisms: consistency mechanism, query mechanism, and copy distribution mechanism. The mechanism mentioned above was a familiar path used in various P2P networks. Different routing algorithms in DHT models were utilized to evaluate this method. The developed approach conflicted with the existing replication techniques, which have good scalability, low search failure rate, high query efficiency and excellent data availability.

Rguibi and Moussa [27] have developed a hybrid trust model to improve the security of B2B networks by reducing passive worm. Two different types of trust models have been developed to mitigate the worm spread, such as the peer-based trust model and the file-based trust model. The requested peer initially checks the searched file's security criteria in the developed hybrid model, defined as the trust file. After that, the most distinguished peer was selected from the entire peer who owned the file to download the progress, defined as peer-based trust. The developed approach was able to find infected downloaded files and infected companions on P2P networks. The developed system was often adapted to mitigate the spread of worms in P2P networks that stop malicious peer activity.

Chuang and Li [28] has developed a trustworthy and churn-resilient academic distribution and retrieval system (TCR) to improve the security of the P2P network. The developed system focuses on four different objective functions. The first objective was to improve that information should not be centralized through a central network administrator. The second objective was used to classify nodes based on the confidence score equations for finding the node, which improves the efficiency and accuracy of each node to send the message. The third objective was to provide a trust management system that would increase reliability, avoid misconduct and improve performance. Finally, a file was able to retrieve the required file in networks. The TCR was implemented and contradicts the existing methodology, which increases search efficiency, enhanced problem recovery, malicious node, higher competitive rate and lower message cost. The existing methods are concentrating only on resource sharing or security enhancement processes. The best process must be developed to attain load balancing and security enhancement in P2P networks.

Elrotub et al. [29] had suggested a virtual machine (VM) to reduce the number of serving requests based on its capacity. The method measured VM's capacity in percentage and map the user request in a group format to active suitable VM. The method gives an improved outcome, yet the method is not fit for a big data system.

Manasrah and Gupta [30] had suggested an optimized service broker routing depend on a differential evolution algorithm to achieve the least processing time and least response time. The different evolutionary process seeks the optimal solution from the possible solution location, so the overall cost, processing time, and response time were minimized. But, the task is how to select the best data centre from a large number of data centres.

Ahuja et al. [31] had suggested a comparison of Amazon Web Services Elastic Compute Cluster (EC2) and Google Cloud Platform (GCP) through three benchmarks. The method was conducted by cluster with an increase in the nodes from one to eight. Both EC2 and GCP offers well reliability and scalability for bandwidth. Yet, the compute instance of latency and bandwidth is within a similar region.

Chui et al. [32] had suggested a handling process of data heterogeneity in electricity load via OCEEMD–WPT. The method contains a power line noise transformation method based on OCEEMD–WPT that merges the ELD datasets. This method effectively improves the signal to noise ratio, but the merging process is not fit for a high-frequency system and large dataset.

2.4 Problem statement

The literature survey listed in this section is broadly categorised into three based on the network type. P2P network has high demand in recent trend due to its advantage in execution. The review listed above shows that the P2P network is highly affected by the load balancing problem and data security.

Load balancing problem Various loads in the P2P network are storage, access and message forwarding among participating peers. In general, the P2P network doesn't have any server or network management system to handle the load flow. So, the unstructured nature of the data flows overloading in any particular node. So, it should be avoided for better network service. Hence our system is planned to balance the load based on a probability-based method. In the proposed mechanism, four parameters like Response time, Price, Availability, and Reputation of the P2P nodes are considered. Then using these parameters, the probability value with QoS is calculated with is given in Eq. (18). The probability value is considered as the maximisation function. So, an optimization algorithm is used to solve this problem.

Security problem Data security is also an essential factor in every network. In the case of P2P, it is again a big task; a simple and effective cryptography mechanism is essential to ensure data security in the P2P network. Thus, in our system, an XOR-based cryptography scheme is employed to ensure data security in the P2P network.

3 Proposed architecture of P2P networks

The P2P network is a system that sanctions numerous people to apportion data and is more efficient than centralised algorithms. The P2P network, which links to the cyber world and operates as a self-regulatory system, stores more resources and information. In comparison to a peer-to-peer network, a conventional network setup includes equal capabilities for each pier and equal opportunities to give accommodations to users. Each utilizer's requests are processed by a central server in the classic server setup. Because each peer connects directly with others without relying on a central server, the P2P network architecture can be run without one [33]. Each pier works as a client or server during data exchange or conversations, which is a mundane benefit of the P2P network.

The most prevalent utilizations of the P2P network are cycle glomming, authentic-time data streaming, and data sharing. The fundamental obstruction in solving this quandary from the perspective of a P2P network is resource revelation. The key arduousness for ameliorating the system's performance in the P2P network is resource finding. To ameliorate the system's efficiency, we require to increment resource utilization right now. Better resource utilisation is achieved by balancing load among peers. As a result, load balancing might be considered a paramount goal of this study. When peers share data, assailers can integrate and access data that users require, averting them from offering a better accommodation. As a result, security might be considered the task's second goal. The proposed solution is intended to avail with load balancing and security. To balance the load among peers, the suggested Probability load balancing control is employed. Furthermore, the suggested ERR with SXOR operation is being developed to secure data sharing in peer-to-peer (P2P) networks. Initially, a single peer is considered for the requestor, who sent a query to other peers, the remaining peer are trying to answer the requestor's requisite resources. For increasing resource utilization and reducing wait time, P2P networks need to act as consistent behavior. Control of load balancing is each node is done via CSO algorithm. CSO analyse each node load at a time and selected the best peer to access the data. In addition, the security of the network is improved via ERR and SXOR. ERR identifying the attacker's node and the malicious peer in a P2P network. As well, SXOR encrypt the data from each peer to store securely and developed the key which is only known for data owners in the peer network. The proposed architecture of the P2P network with load balancing control and security enhancement is illustrated in Fig. 1.

P2P network load balancing and security enhancements ameliorate system efficiency, abbreviate waiting time, and expedite execution. A proposed load balancing controller can avail P2P networks to make better utilization of their resources. Furthermore, security enhancements might amend the functionality of a system during a malignant attack by unauthorised individuals. P2P has become a well-kenned method of sharing data across a wide group of peers, and it has several advantages over traditional client-server setups [34]. The process of peer-to-peer data exchange can be summarised as follows:

- **Definition of peer:** A peer, also known as a node, is a member of the group who serves as a hub for data sharing. In the P2P network, Peer also serves as a server, storing files.
- **Definition of Query:** Any peer requires a resource that sends queries to other peers to obtain the data they

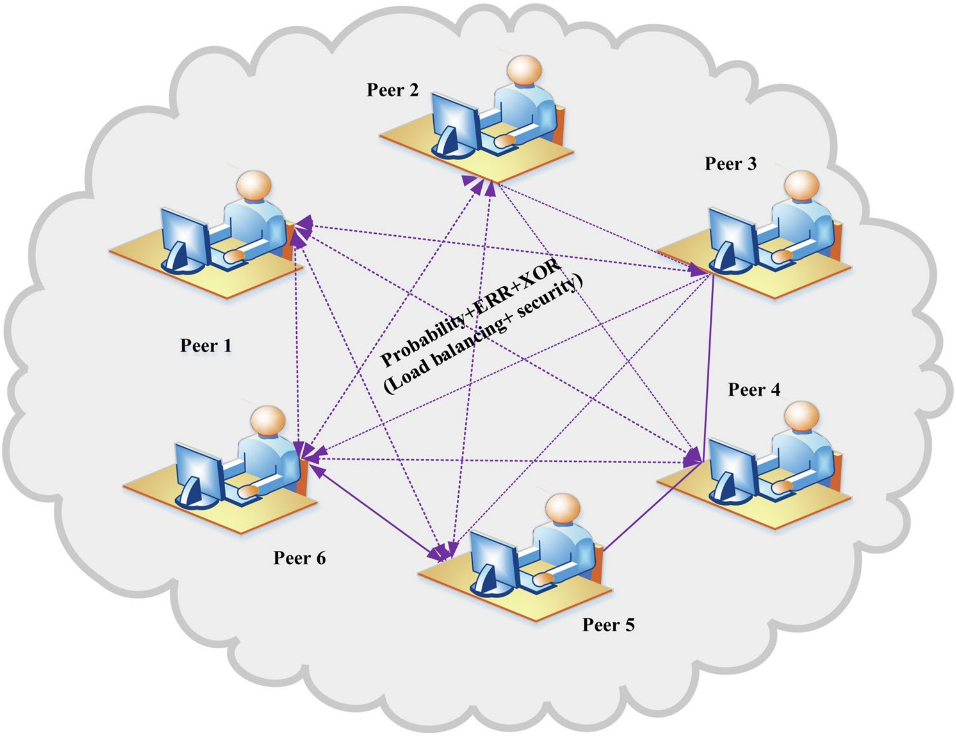


Fig. 1 Proposed Architecture of P2P network

require. After receiving a query from a peer, the remaining colleagues attempt to react to the inquiry message.

- **Definition of Requester:** On the P2P network, each peer has a lot of resources. However, the peer's resources would be insufficient to meet its requirements. It then decides to send queries to nearby peers to obtain resources. By responding to a query, neighbours can be satisfied with the requisite resources of peers. Because one or more peers would answer the request, the query sender must wait until it obtains resources from neighbours.

When each peer's high load is reduced, the P2P network structure's waiting time and processing time are reduced automatically. For the analysis of the proposed P2P network, six peers are evaluated, as shown in Fig. 1. The P2P network's probability-based load balancing is improved, with probability value neighbour peers picked for data exchange among peers. With the use of the CSO algorithm, locate the best neighbour peer for data sharing progress, the probability load balancing control is increased. To achieve great security, ERR with SXOR operation is applied. The next section contains a full discussion of the suggested methodology.

3.1 Probability load balancing control with resource discovery

In a peer-to-peer network, a single peer is designated as the requestor, who transmits the query to other peers, who then seek to provide the requestor with the resources he or she requires [35]. On the peer-to-peer network, consistent conduct yields superior results in terms of resource utilisation and wait times. CSO is used to perform load balancing and to take into account the likely scenario for each peer in a P2P network. Only the probability value of each peer should be estimated using this, and the P2P network's peer load balance is achieved. Figure 2 depicts the process structure in further detail.

The steps should be followed to attain load balancing control in the P2P network:

- **Step 1:** Initially, the query can be thought of as a message delivered by the requester peer to its neighbours. The peers are connected by the direct edge. With the remaining peers in the P2P network, the requester's peer requirements must be met.
- **Step 2:** The requester peer sends a message to the neighbour peers, who are ready to answer. Neighboring peers also send an acknowledgement message to the requester peer. The acknowledgement message includes information on the requester's peer, as well as detailed information about the peer's conditions and the scenario's supplied resources.
- **Step 3:** Then, based on probability conditions, CSO optimization can be used to pick peers. The peers chosen by the requester have the requisite resources. Based on the likelihood criterion, the requester chooses a neighbour peer to move.
- **Step 4:** The first chicken family will go to the selected peer. The placements of the chickens are modified based on their updating behaviour.

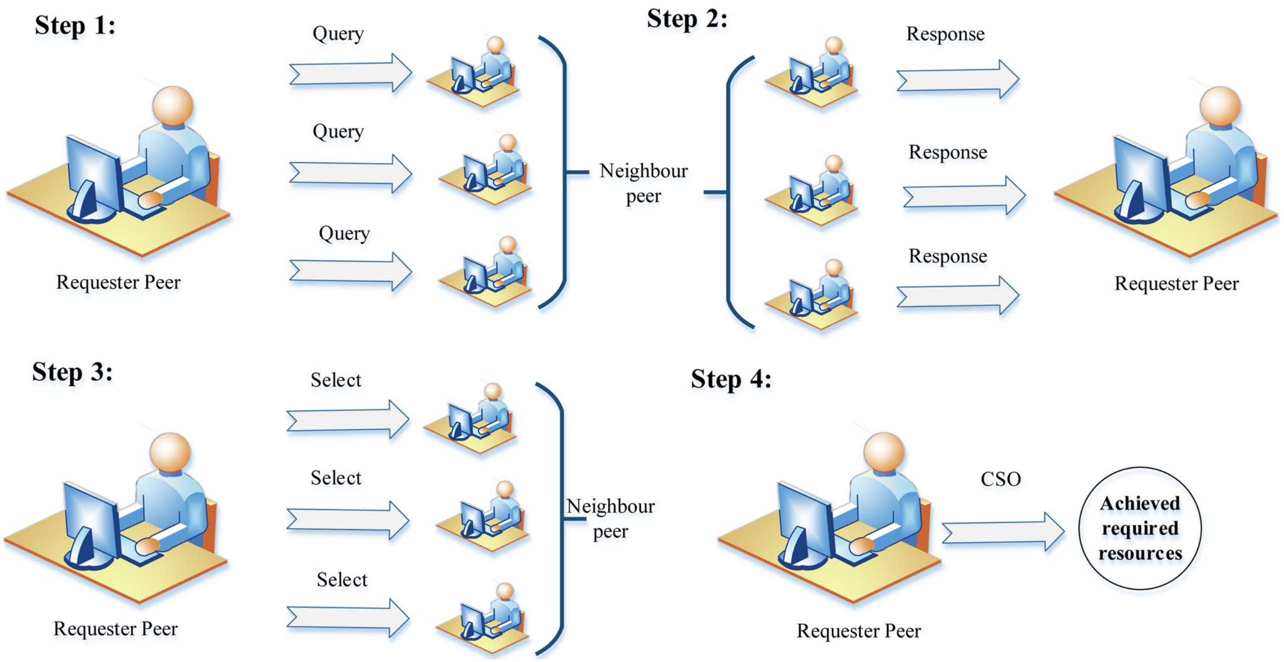


Fig. 2 Steps of proposed load balancing control with CSO

Numerical representation can be used to analyse the performance of load balancing control. Some of the assumptions used to analyse load balancing control in the P2P network system model are as follows:

In the P2P network, six peers are randomly designed, which represented as Peer 1 (P1), Peer 2 (P2), Peer 3 (P3), Peer 4 (P4), Peer 5 (P5) and Peer 6 (p6), which are presented as follows,

$$Peers = \left\{P^1, P^2, P^3, P^4, P^5, P^6\right\} \tag{1}$$

Each peer of the P2P network have different resources, which are described as follows,

$$Resources = \left\{R^1, R^2, R^3, R^4, R^5, R^6\right\} \tag{2}$$

The resources are not containing any one of the peers, and each peer have a different combination of resources which presented as follows,

$$P^1 = \left\{R^1, R^6\right\} \tag{3}$$

$$P^2 = \left\{R^2, R^3\right\} \tag{4}$$

$$P^3 = \left\{R^2, R^4\right\} \tag{5}$$

$$P^4 = \left\{R^2, R^4, R^5\right\} \tag{6}$$

$$P^5 = \left\{R^1, R^4\right\} \tag{7}$$

$$P^6 = \left\{R^1, R^3, R^5\right\} \tag{8}$$

The quality parameters also considered processing load balancing control in the P2P network, such as response time, price, availability, and reputation presented in Table 1.

N=6 can also be assumed for six different chicken families. Peer 2 can be regarded as a requester in this numerical calculation. The six chicken families begin their search for the ideal peer to meet peer 2's needs. When Peer 2's requirements aren't met by ourselves, it sends an inquiry. To satisfy the required resources, the next peer would be picked from among the remaining peers [36]. The random procedure of peer selection results in a high load burden, which increases processing time and waiting time. To meet the required resources to solve the problem, the CSO method with probability-based best peer-selected among peers is used. The CSO method is used in the suggested load balancing control to discover the optimal peers, which are completely related to the probability scenario condition.

The final solution of i^{th} chicken family in the best peer selection to satisfy resources of peer 2, which is denoted by $FS(i)$. The initial condition of the solution can be denoted as follows,

$$FS(i) = \{\} \tag{9}$$

The chicken family can be selected as a peer to move. The selected peer is added to the solution. For example, $FS(i) = \{\boldsymbol{P}^3\}$. Similarly, the required resources of a peer are denoted as $FR(i)$ in the i^{th} chicken family which is considered in the initial condition is described as follows,

$$FR(i) = \{R^4, R^5\} \tag{10}$$

When providing resources to requester peer, the required resources changes to zero in the above equation. Already discussed, $\boldsymbol{P}^2$ can be considered as requested peer which required resources of $\{R^4, R^5\}$.

A chicken family chooses the best peer from the CSO to collect resources from the remaining peers in a network based on the probability value. The peer with the highest likelihood of giving the requisite resources to peer 2 can be chosen. Each chicken family changes their place and exits the system after selecting high likelihood peers. Similarly, leftover peer-required resources provide a minimal amount of energy and less waiting time in the long run. Quality characteristics are also taken into account when choosing a peer from whom to obtain resources, as shown in Table 1. By computing the best peer in the P2P network, the suggested probability load-balancing control follows stages to achieve the load-balancing aim. The probability condition can identify and select the best peer from among a group of neighbours. The CSO is a key player in the data sharing process. The process of the CSO algorithm is described in the following section. Furthermore, to improve the system's performance, security is an important component of the P2P network. The next section provides a full description of the proposed security control.

3.2 Proposed chicken swarm algorithm for optimal load balancing

The CSO method is used to improve load-balancing behaviour in P2P network structures, allowing for better resource

Table 1 Quality Parameters in the P2P network

Parameters	P^1	P^2	P^3	P^4	P^5	P^6
Response time	80	75	100	90	70	80
Price	3	4.5	2	2.5	4	3.5
Availability	0.98	0.96	1	0.99	0.97	0.97
Reputation	0.8	0.8	0.9	0.9	0.7	0.8

use and faster response times. To meet the requirements, the requestor peer sends the query together with the required resources to surrounding peers. The probability level of each peer in the P2P network can be used to accomplish load balance. CSO is a bio-inspired optimization method for finding the optimal peer in a peer-to-peer (P2P) network to enhance load balance. This system is activated in response to the behaviours and hierarchy of chicken swarms [37, 38]. The chicken swarm family can be classified into various categories. One rooster, many chickens, and several chicks make up each group. Distinct chickens in the same family may have different operating norms. CSO is used to choose the best peer for each peer's needs, resulting in better load balancing behaviour in a P2P network. Chicken habits are best serviced with the following principles for simplicity's sake:

- The chicken swarm family is divided into several subgroups. Each group contains a cock, a pair of chickens, and a large number of chicks.
- Because it is the head cock in the group, roosters from the poultry family have the finest fitness. The chickens with the lowest fitness value are considered chicks, and the remaining birds are chosen at random. Chickens and chicks are set up at random with their mother-child resemblance.
- The head cock and the mother-child bond in the chicken family are unaffected. It would be updated regularly.
- Because they may prevent others from consuming their food, chickens follow their head roosters in quest of nourishment. Chicks locate nourishment by following their mother hen.

In the chicken family, the number of roosters, the number of hens, mother hen and the number of chicks is denoted by nr, nh, mh, nc. From that, the best fitness is considered as nr and the worst fitness is considered as nc. The remaining are considered as hens. The position of chickens is denoted by follows,

$$X_{a,b}^{T}, (a\epsilon[1, \dots, N], (b\epsilon[1, \dots, D)) \tag{11}$$

where T can be described as iteration, and D denotes dimensional step. The movement of chickens are presented below,

3.3 Movement of chickens

The best fitness value of roosters has the priority of accessing the food rather than the rooster with poor fitness value [39]. This situation can be divided as follows,

$$X_{a,b}^{T+1} = X_{a,b}^{T} * (1 + Random(0, \sigma^2)) \tag{12}$$

$$\sigma^2 = \begin{cases} 1, \ if\ F^a \le F^k, & k\epsilon[1, N], k \ne a \\ exp\left(\frac{(F^a \le F^k)}{|F^a|+\epsilon}\right), & otherwise \end{cases} \tag{13}$$

where f can be described as fitness value of the related X, k can be described as a rooster index which selected from rooster group randomly, ε is a small constant value for avoiding zero division error, $Random(0, \sigma^2)$ can be described as Gaussian distribution with a standard deviation σ^2 and mean value 0.

The more dominant chickens have the advantage of competing for food than the more submissive ones. This situation can be presented as follows,

$$\begin{aligned} X_{a,b}^{T+1} = X_{a,b}^{T} &+ M1 * Random * \left(X_{R1,b}^{T} - X_{a,b}^{T}\right) \\ &+ M2 * Random * \left(X_{R2,b}^{T} - X_{a,b}^{T}\right) \end{aligned} \tag{14}$$

$$M1 = exp\left((f^a - f^{R1})/(abs(f^a) + \varepsilon)\right) \tag{15}$$

$$M2 = \exp((f^a - f^{R2})) \tag{16}$$

where, $R2$ and $R1$ are index values of chicken and hens groupmate, which limits are $R2\epsilon[1, \dots, N]$ and $R1\epsilon[1, \dots, N]$. The index values should not be the same for hens and chickens, randomly selected from the swarm. *Random* Can be described as a random number over [0, 1].

Perceptibly, $f^a > f^{R1}$, $f^a < f^{R2}$ thus $M2 < 1 < M1$. For assumption, $M1 = 0$, then ith chicken would continue to feed the other chickens. The big fitness values and the narrow gap between the little $M2$ and the two chickens distinguish two chickens. As a result, chickens are less likely to steal food from other hens. M1's formulaic form differs from $M2$ in that it contains matches in a group. The fitness value of the chickens is related to the fitness value of the chickens, and rivalry amongst the chickens in a group is simulated. Assume $M2 = 0$, and the chicken is hunting for food based on his actions. The fitness value of the service is unique to a certain group of families. As a result of the limited spacing between the placements of the ith chicken and its group mate's cock $M1 = 1$ and the tiny ith hen fitness value $M1 = 1$. As a result, dominant chickens are more likely to be dominant than submissive chickens. Chicks in the poultry family follow their moms, who are shaped like this:

$$X_{a,b}^{T+1} = X_{a,b}^{T} + fl * \left(X_{s,b}^{T} - X_{a,b}^{T}\right) \tag{17}$$

where the position of ith chick's mother can be denoted by $X_{s,b}^{T}$ which range $(s\epsilon[1, N])$.= and chicks can follow their mothers to eat food which denoted by parameter $fl(fl\epsilon(0,2))$. Based on the individual difference, fl of each chick can be selected randomly among 0 and 2. The step by step procedure of the proposed CSO algorithm is presented as follows:

Step 1: Initialisation phase

Six chicken swarm families can be created during this phase. A request query of resources was also used to initialise the resources of each peer.

Step 2: Fitness evaluation

Fitness is defined as a peer probability calculated using the peer's requester resources. The entire peer structure has the resources necessary to meet the requestor peer's needs. The probability is computed based on the below formulation,

$$Probability\ function = QoS(t) * \frac{P}{N} \tag{18}$$

$$QoS(t) = \left|P_b \cap FS(i)\right| \tag{19}$$

$$P = Availability + reputation \tag{20}$$

$$N = Price + response\ time \tag{21}$$

The best neighbour peer can be chosen based on probability functions to satisfy the requestor peer's requirements. Load balancing is performed effectively in a P2P network when resources are used to their full potential.

Step 3: Updating process

The fitness function is used to update the positions of the hens, roosters, and chicks.

Step 4: Checking the termination condition

When the procedure reaches the termination condition, the results are saved in this phase.

Figure 3 depicts the overall process of suggested CSO-based best neighbour peer selection to meet requestor requirements.

3.4 Security enhancement based on ERR with SXOR operation

To enable proper data exchange and resource usage in a P2P network, security development is a critical challenge. The P2P network has the capability of connecting and disconnecting users. Any malicious or attack peer could be added to the network as a result of this behaviour, causing the system's performance to be completely disrupted. If a malicious peer is introduced to the system, data can be retrieved or viewed while the data sharing process is in place. Before stealing or accessing data from the network, data must be safely passed across peers, and the malicious peer or attack node must be detected. In a P2P network, an ERR can be used to detect rogue nodes and hostile peers. Based on the likely scenario and the CSO algorithm's assistance, the recommended load balance control is the best peer to supply the requested resources to the requestor. The selected peer may behave as an attacker, causing the data sharing process and procedure in P2P networks to entirely break down. The chosen peer should examine the situation in order to identify the attacker and malicious nodes. The node condition is tested using the ERR approach. In the P2P network where ERR and SXOR operate, the dual security analysis is taken into account. The P2P network has a variety of resources that are protected by encrypted creation, which improves security.

3.4.1 ERR technique

The initiator node transmits an encrypted inquiry and core message via the Rumor Riding protocol. When an Intermediate node receives this query and the key messages, it utilizes the key to decode the message. The node then integrates an IP address to the query message afore sending it to the other nodes in the network. A node can accommodate a Responder, containing the desired file. As a result, it will utilize the Intermediate node to relay a replication to the initiator. Possible attacks of ERR contain start-up node, Intermediate node and respondent node.

When a malicious node poses as a start-up node, it sends out a bogus request and launches bogus search resources [39]. The chosen peer (neighbouring peer) could be a rogue node, slowing down the system. The responder node could be a malicious node that sends the initiator node a duplicate answer. Figures 4, 5, and 6 depict the various attack conditions. The initial attack causes full system failure by delivering a duplicate response to the remaining peers' network of names as the asking peer. The ERR approach can be used to identify the attacker [40]. Each peer encrypts and stores the resources. It is impossible to assign a key to a user for them to access resources without first confirming peer behaviour. Different queries can be used to verify the process described in the ERR. The distinct conditions of ERR and SXOR operation are used to identify the resources and malicious nodes. This section briefly describes the proposed ERR and SXOR operations in detail.

A demanding query in the ERR approach can be found in the P2P network by the attacker peer. The initiator peer or requester peer sends a query message with the challenge questions stated below: IP address, success rate, and location. With inquiries, three queries are sent to neighbours. The attacker doesn’t know the answers to

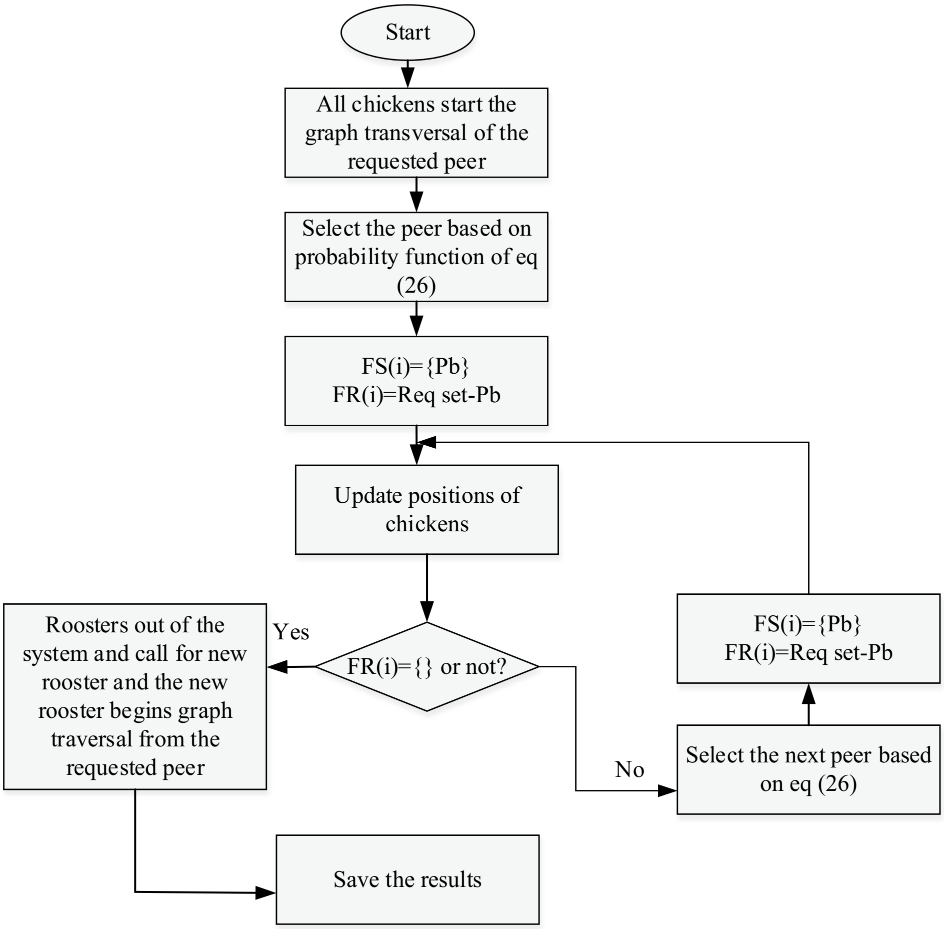

Start
All chickens start the graph transversal of the requested peer
Select the peer based on probability function of eq (26)
FS(i)={Pb}
FR(i)=Req set-Pb
Update positions of chickens
FR(i)={} or not?
Yes
Roosters out of the system and call for new rooster and the new rooster begins graph traversal from the requested peer
No
Select the next peer based on eq (26)
FS(i)={Pb}
FR(i)=Req set-Pb
Save the results

Fig. 3 Process of CSO to load balancing control

such queries; only the solutions are known to the secure peer. Each peer's status can be obtained depending on the situation, and the required resources are collected from the P2P network's neighbour. In the initial state, each peer's IP address, location, and success rate can be assigned based on previous transactions, which can help identify each peer's status [41]. The ERR approach in the P2P network identifies the attacker node, which

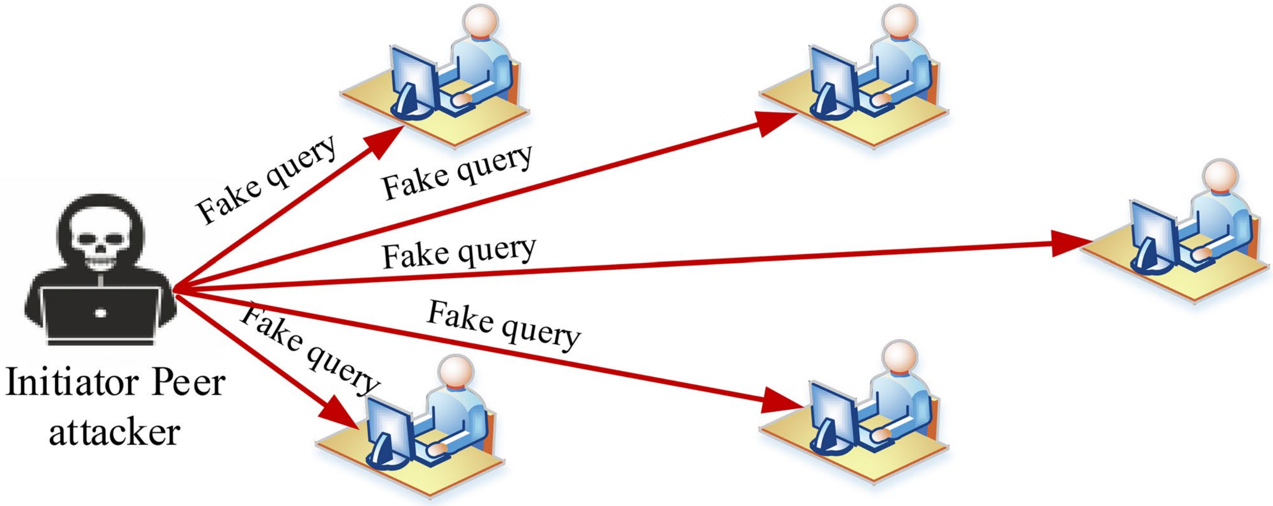


Fig. 4 Initiator or requester peer attacker

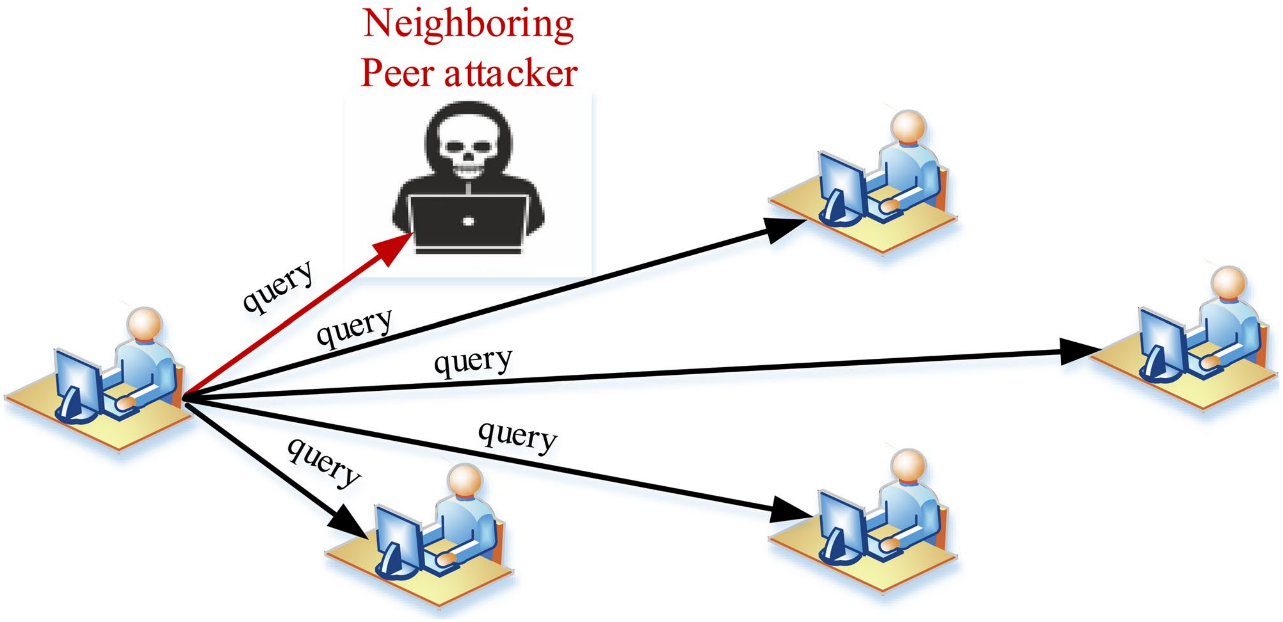


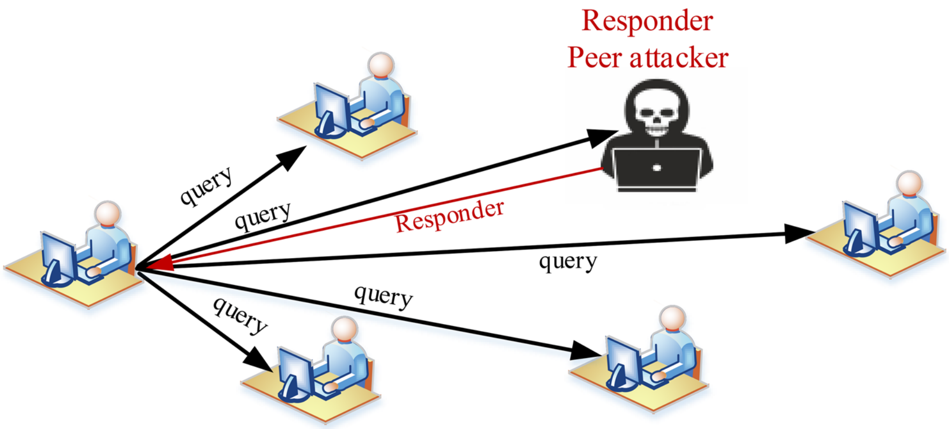
Fig. 5 Neighboring or intermediate peer attacker

improves the system's performance. The neighbour node is chosen based on its probability value, which is also used as an initial selection of peers for compensating required resources in the P2P network to improve load balancing control. The CSO algorithm is used to calculate the peer probability value, which can aid in the selection of the peer by the requester peer to obtain the required resources. The attacker's peer or safe node scenario can then be identified using the ERR approach on the selected peer. In a peer-to-peer network, the term "safe node" refers to a node that promptly begins the transaction process. SXOR was created to protect data resources in each peer. The SXOR procedure is used to protect resource storage in each peer. The operation of SXOR is described in-depth in the section below.

3.4.2 SXOR (Split-XOR) operation

Additionally, the SXOR introduced encryption to secure data resources in each peer when resources are required for access, which is done by decryption. The key, which only knows the data saver in peers, was also developed. Each peer in a peer-to-peer network has data resources that must be stored securely. If security measures are not maintained, any unauthorised individual will be able to access data resources in peers. Secure storage is accomplished by the use of the SXOR operation [42]. The SXOR operation is a high-performance security procedure that sends data to peers before retrieving it. Before sending data resources to requester peers, the data resources are stored in responder peers. Initially, SXOR

Fig. 6 Responder peer attacker



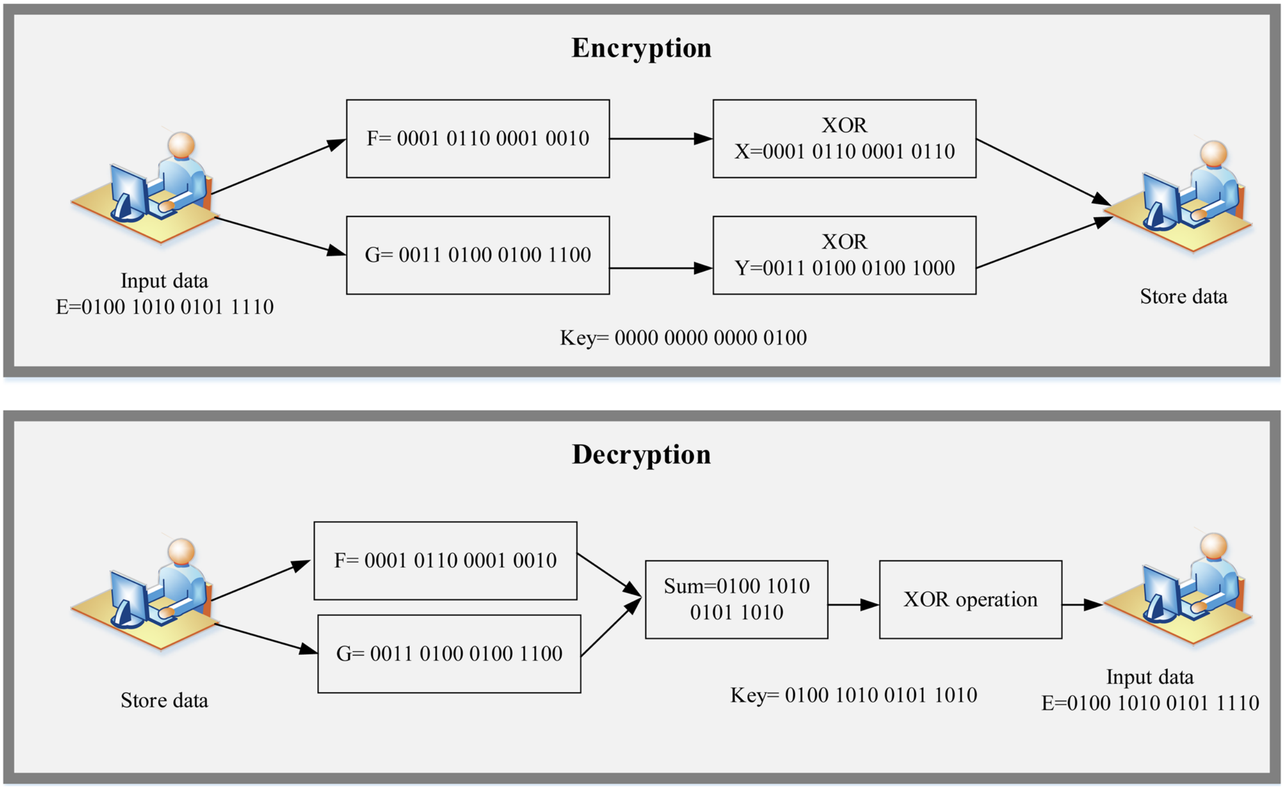


Fig. 7 Process of SXOR operation

encryption stored the input data. Encryption is the process of creating a high-security key, which is known only to the data saver in peers [43]. Use the key to access any data from the data store. If an unauthorized person sends the request, the CSO analyzes the network to identify and block it. The process of XOR operation can be illustrated in Fig. 7.

The random key is used to generate the key rated as 0100 and perform XOR operation with E and F's split elements. The XOR function can obtain using the function given below;

$$X = F\ XOR\ key \tag{22}$$

$$Y = G\ XOR\ key \tag{23}$$

where, X and Y values are $X = 0001011000010110$ and $Y = 0011010001001000$. On the split elements stored in the peers, after performing the XOR operation. Once the peer status has been verified, this divided data can be distributed to colleagues together with the key. The SXOR function can split input resources into two halves, which abusers and attackers can't do because the key value and data elements are impossible to predict [42]. The requestor obtains data resources by attempting to retrieve the data through the encryption process. First, the split data performs the XOR function with the key separately. From the SXOR function, achieve two different values such as $F = 0001011000010010$ and $G = 0011010001001100$. After that, these two values can be summed, and $E = 0100101001011010$. Send final data to do the XOR function to achieve original input data. The final output of the encryption process is given below,

$$E = 0100101001011110 \tag{24}$$

The SXOR function is capable of effectively storing user data. The crucial value is created at random, and no content information is carried in the split data. On each peer, the SXOR function is employed to store data resources, which provides great security. With the help of the SXOR operation, resources are handled safely in a P2P network.

The CSO method is primarily used to choose a peer-based on likelihood to satisfy the requestor's resource requirements. Similarly, the use of ERR and SXOR in peer networks increased security. The goal of the CSO algorithm is to choose the best peer to provide the requested resources to the requestor. After then, the ERR approach, which provides status, can be used to determine the node state. The resources are transferred and required resources are delivered to requestor peers in the P2P network once a safe node has been found. The data resources are encrypted using the SXOR procedure, which ensures network security. The key and encrypted data can be transferred to a requestor peer in the network, who can then use the key to decode resources via the XOR process. Finally, load balancing and security enhancement are achieved in the P2P network topology by utilising CSO, ERR, and SXOR operations. The proposed control architecture is implemented in the P2P network, and the simulation results are reported in the following sections.

4 Experimental evaluation

In the P2P networks, load balancing and security are attained with CSO, ERR, and SXOR operations. The proposed methodology performance can be analysed and verified with the help of simulation environments. The proposed method is implemented in the CPU speed of 2.20 GHz with 8 GB of RAM using the NS2 platform with LINUX. The proposed method is analysed with performance metrics such as Average end to end delay, delivery ratio, packet loss, security, encryption time, decryption time, and throughput. The proposed method's efficiency is proved based on a comparison with existing BCIRR, CA, and ACO methods in performance metrics. The implementation parameters of the proposed method are presented in Table 2. The simulation generates 100 peers with resources and defined parameters of availability, reputation, response time, and price.

The initial node creation of P2P networks is presented in Fig. 8. In the P2P networks, 100 nodes are created with different IP addresses and individual properties. Each peer has the resources and can send queries to other peers to attain the required resources from other neighbor peers. Based on the query, the best neighbour peer can be selected by checking probability conditions checked through constraints. Each peer has the constraints of availability, price, reputation, and response time, which compute probability based on service function quality. The CSO is developed to identify the best peer for compensating requirements of resource query of requester with quality. The proposed method also enhances security by ERR technique and SXOR operation. In the P2P network, each peer's resources are stored encrypted with SXOR operation's help to avoid unauthorised person access. Initial communication of P2P networks, which is illustrated in Fig. 9. The ERR technique is used to find out the attacker node by asking questions to remaining neighboring peers, such as IP address, success rate, and location. If a peer provides an appropriate solution for these questions, it would be selected as a safe node. With the ERR technique's help, the attacker node is identified, such as the initiator peer attack, neighbour peer attack, and responder node attack. The attacker identification related to three attacks is illustrated in Figs. 8, 9, 10, 11, and 12.

To analysis, the proposed method in P2P networks, the communication of peers initialised in P2P networks by sending queries to other neighboring peers. From P2P networks, one peer has started the communication by sending the query to other neighboring peers. The best neighbour peers must be selected to achieve load balancing in P2P networks, which improve resource utilization of networks. The requestor peer also gets the required resources from the other one within less time consumption. The selected peer should be secure, which can be achieved with the ERR technique and SXOR operation. The attack may happen in the initiator node, neighboring node, and responder node, identified by the ERR technique. Based on the proposed method, load balancing control and security can be achieved through probability conditions. The performance of the proposed methodology is presented in the below section.

Table 2 Implementation parameters for BCIRR method

S.No	Description	Value
1	Simulator	NS2
2	Number of nodes	100
3	Simulation area	1630 m X 750 m
4	Packet size	500 bytes
5	Node type	Static
6	Time of simulation	50 s
7	MAC TYPE	MAC / 802_11
8	Propagation	Two Ray Ground
9	Antenna	Omni Antenna
10	Traffic Source	CBR
11	Initial population	100
12	Total Iteration	100

4.1 Performance analysis

The performance of the proposed method is analysed with performance metrics. Based on the load balancing control and security control of the proposed method, the performance of the P2P network is developed under the attacking condition. Performance metrics such as delay, packet loss, delivery ratio, encryption time, decryption time, and throughput are considered to evaluate the proposed method's effectiveness. Delay can be defined as the time engaged aimed at a packet to be communicated the network diagonally from one peer to another destination

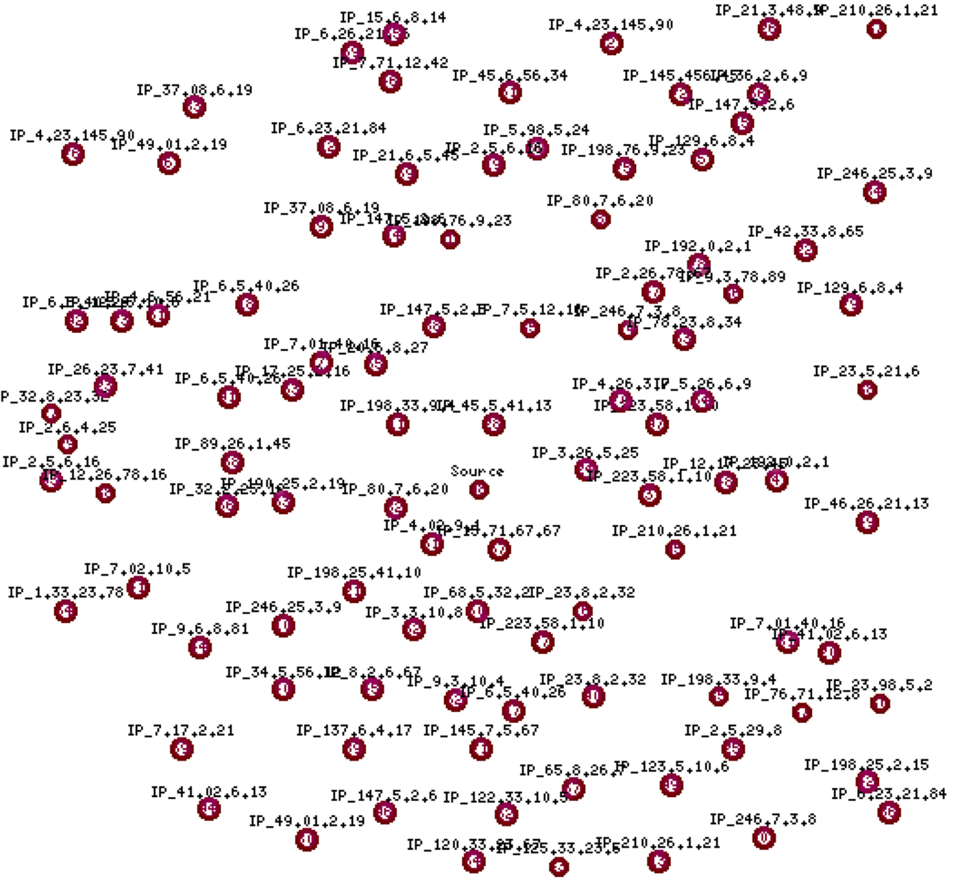


Fig. 8 Initialisation of nodes in P2P networks

peer. The packet loss is derived from solitary or additional communicated packets to send the destination peer from the sender. The failure packet data can be considered packet loss. A delivery ratio can be derived as the percentage of packets gathered to the overall sending packet amount. Throughput is a measurement of how many units of packets can be processed in a given amount of time. The computation of performance metrics is presented in the given formula.

$$Delay = \frac{TD}{PR} = \frac{Total\ delivery\ time}{Packets\ received} \tag{25}$$

$$packet\ loss = Packet\ received(PR) - packet\ send(PS) \tag{26}$$

$$Delivery\ ratio\ DR = \frac{PR}{PS} \times 100\% = \frac{Packet\ ratio}{packet\ send} \times 100\% \tag{27}$$

$$Throughput = \frac{PR}{SE} \times SZ = \frac{Packet\ received}{Simulation\ end\ time} \times packet\ size \tag{28}$$

Based on the performance metrics, the verification and analysis of the proposed method are done. The delay and packet loss of the proposed method should be below, which is only considered the best performance system. The throughput and delivery ratio should be high, which is only considered as the best performance system. The proposed method satisfies the above conditions related to performance metrics. The delay, packet loss, delivery ratio, and throughput of the proposed method are simulated and illustrated in Figs. 13, 14, 15, and 16.

The performance metric of delay is illustrated in Fig. 13, which must attain in low value. The proposed method has a minimum delay value of 5 ms while comparing it with the other techniques like BCIRR, ACO and CA. The simulation is executed for 50 s, so the value is noted for every 5 s. At the time of 5 ms of the simulation, the proposed system's delay is 0.25 s, whereas the other techniques, BCIRR, ACO and CA, have the delay of 0.5, 076, and 0.99, respectively. Even though the proposed system has the optimization for load balancing, the delay is minimum because of the usage of CSO. Due to the better convergence performance of CSO, the overall delay get

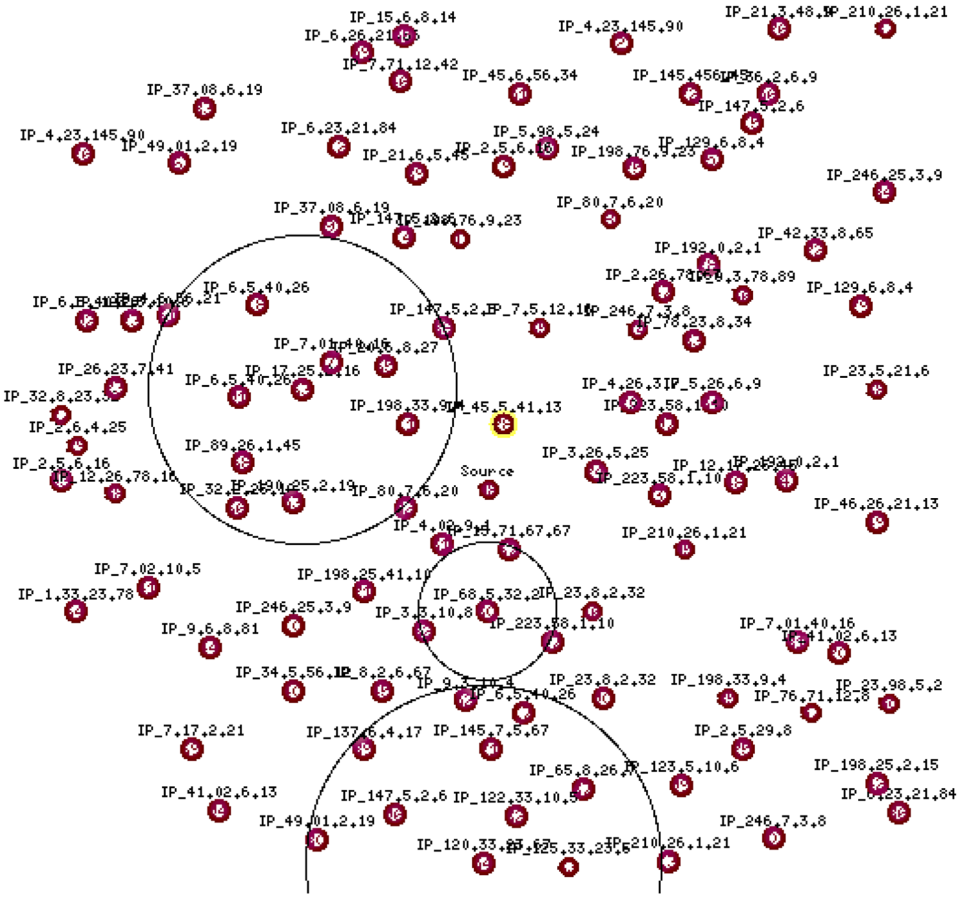


Fig. 9 Initialisation of communication in P2P networks

reduced in the proposed system. Similar to the delay evaluation, the packet delivery ratio performance is evaluated for the period of 50 s execution. The proposed delivery ratio's performance metric is illustrated in Fig. 14, which must attain a high value. The proposed delivery ratio minimum value is 94 at 5 ms, and the maximum delivery ratio is 98.75% at 50 ms.

The performance metric of packet loss is illustrated in Fig. 15, which must be attained at.

a low value. The proposed method has a packet loss value of 200 at 5 ms. Similarly, packet loss values are changed to 300 at 10 ms. The minimum limit of packet loss is 200 at 5 ms, and the maximum limit of delay is 680 at 50 ms. The performance metric of throughput is illustrated in Fig. 16, which must be attained in high value. The proposed method attain a throughput value is 5 at 50 ms.

Similarly, throughput values are changed to 5 ms. The minimum limit of throughput is at 5, and the maximum limit of throughput is at 50 ms. The resources of each peer in a P2P network can be stored securely with SXOR operation. The SXOR operation is working based on encryption and decryption methods. The specific time of encryption and decryption is presented in Fig. 17. Based on the analysis of the results, it is clear that the proposed system provided better performance than the other techniques. Table 3 present the time comparison of the encryption and decryption process.

4.2 Comparison analysis

The comparison analysis is an essential part of the research paper to prove the proposed method's efficiency. The proposed method of probability-based load balancing control of P2P networks is achieved with the CSO algorithm's help. The peer can send a query request to other neighboring peers. The request is sent to each neighboring peer, which checks them resources with request resources. From neighboring peer who has same resources related to request resources that one ready to send requestor resources. The query sending and get resources in P2P networks to consume significant waiting times. To reduce the waiting time, load balancing control and security enhancements, the proposed method

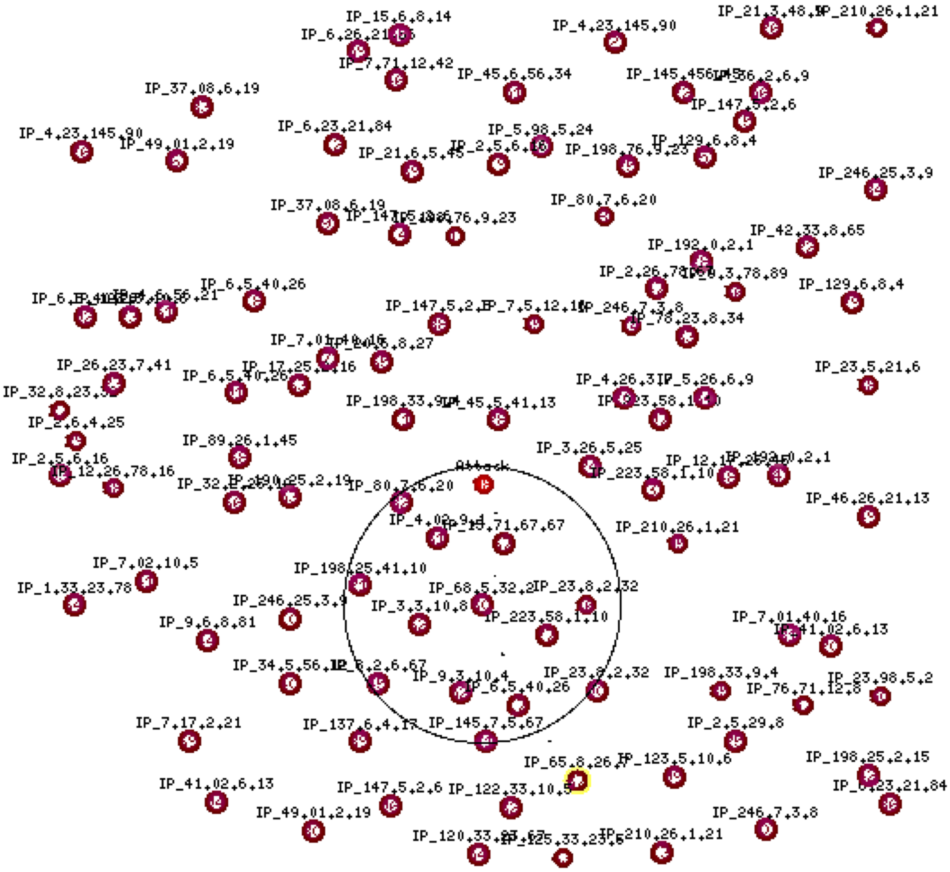


Fig. 10 Identification of initiator peer attack

is developed. The proposed method is compared with existing methods such as BCIRR, ACO, and CA methods. The proposed method's comparison analysis is analysed with performance metrics such as delay, delivery ratio, throughput, and packet loss.

The comparison analysis of the proposed method performance metrics is illustrated in Figs. 18, 19, 20, and 21. Figure 18 describes the delay values of proposed and existing methods. From the figure, the proposed method has a delay is 0.20×10^{-4}; BCIRR have the delay is 1.2×10^{-4}; ACO have the delay is 2.1×10^{-4}; CA have the delay is 2.6×10^{-4}. The proposed method has a low delay value from the analysis than the BCIRR, ACO and CA methods. Figure 19 describes the delivery ratio values of proposed and existing methods. From the figure, the proposed method has a delivery ratio of 0.96; BCIRR has a delay of 0.94; ACO has a delivery ratio of 0.92; CA has a delivery ratio of 0.90. The proposed method has a high delivery ratio value compared with the BCIRR, ACO and CA methods from the analysis. Figure 20 describes the packet loss values of proposed and existing methods. The proposed method has a packet loss of 1.97; BCIRR has the packet loss of 2; ACO has the packet loss of 7; CA has the packet loss values of 9. The proposed method has a low packet loss compared with the BCIRR, ACO and CA methods from the analysis. Figure 21 describes the throughput values of proposed and existing methods. From the figure, the proposed method has a throughput of 2.6×10^{10}; BCIRR have the throughput of 2.5×10^{10}; ACO have the throughput of 2.3×10^{10}; CA have the throughput of 2.1×10^{10}. From the analysis, the proposed method has a high throughput value compared with the BCIRR, ACO and CA methods. Based on the comparison analysis, we can conclude, the proposed technique delivers the optimal solutions efficiently (Table 4).

The execution times of the proposed and existing methods were also investigated. The approach's operational time refers to the time it takes to detect a network attack and load balancing. Within 350 ms, the proposed method detects the attack and balance the load. The existing techniques BCIRR, ACO and CA take 398 ms, 430 ms and 500 ms, respectively.

The proposed method offers higher performance value and lower delay time compared to the load balancing method of cluster-based replication architecture [44]. The existing method has a throughput is 1, yet the novel CSO have 2.6×10^{10} Mbps. There is no packet loss in the existing method but the proposed CSO have 1.97% packet loss (Table 5).

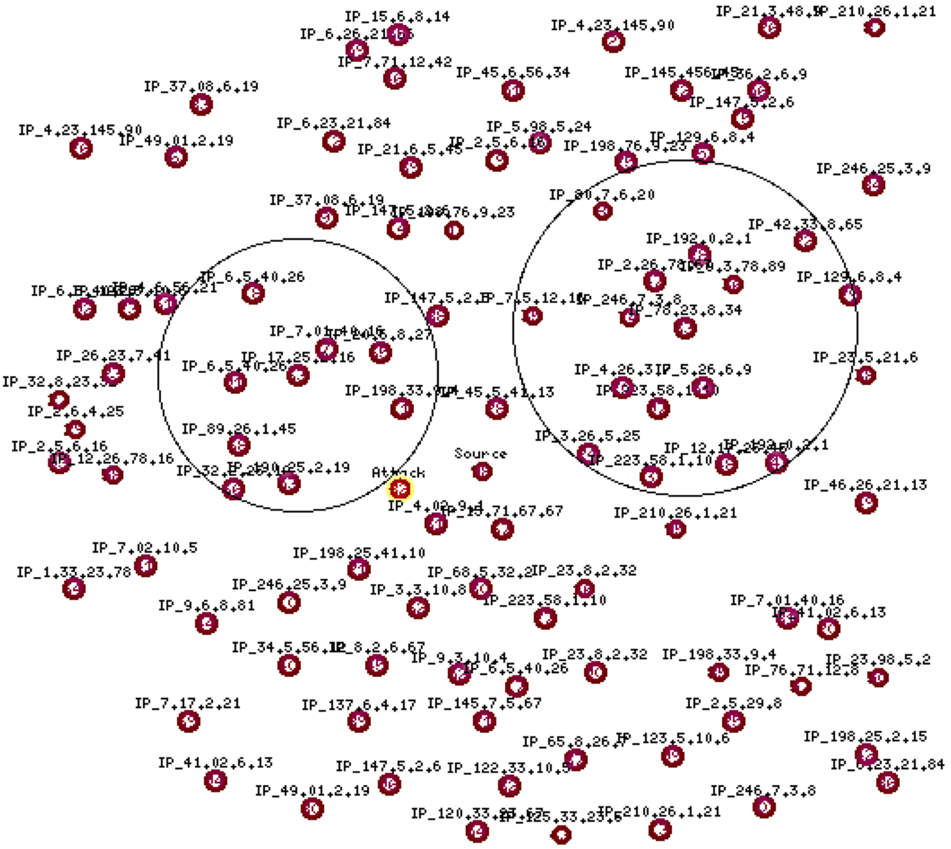


Fig. 11 Identification of neighbour peer attack

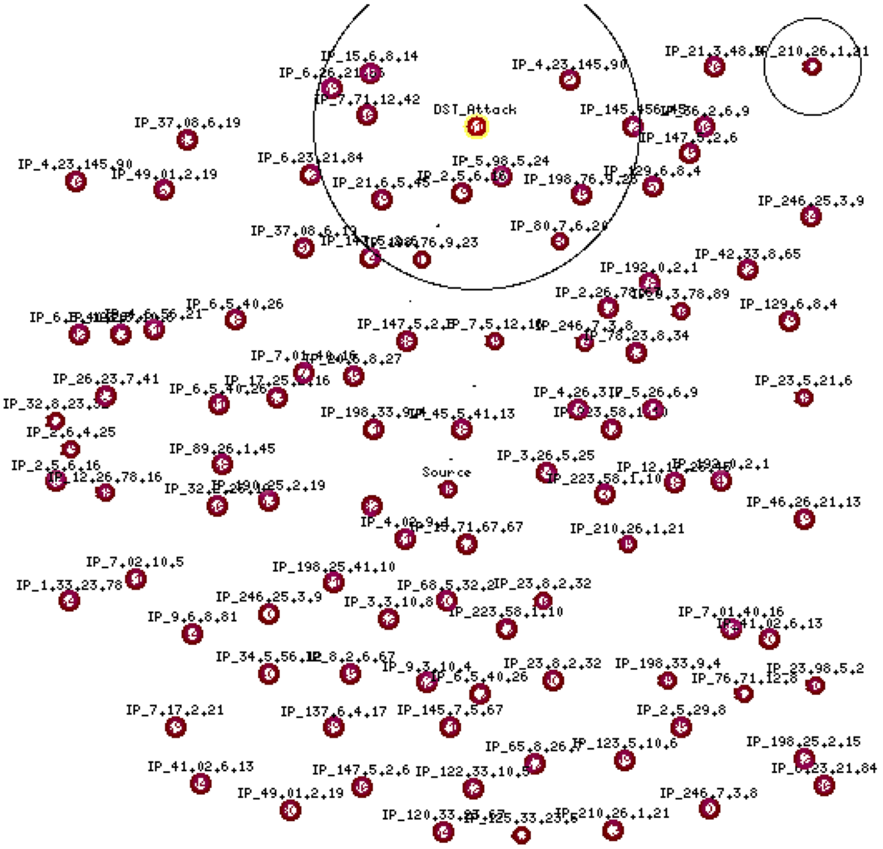


Fig. 12 Identification of responder peer attack

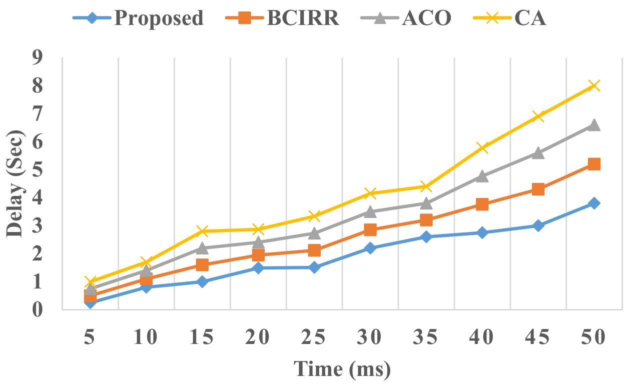


Fig. 13 Evaluation of Delay in the proposed method

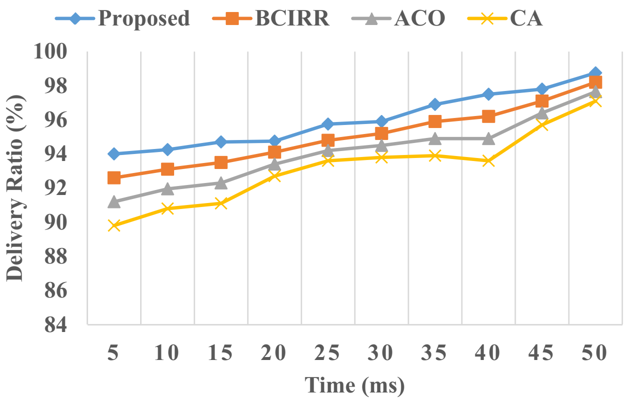


Fig. 14 Evaluation of Delivery ratio in the proposed method

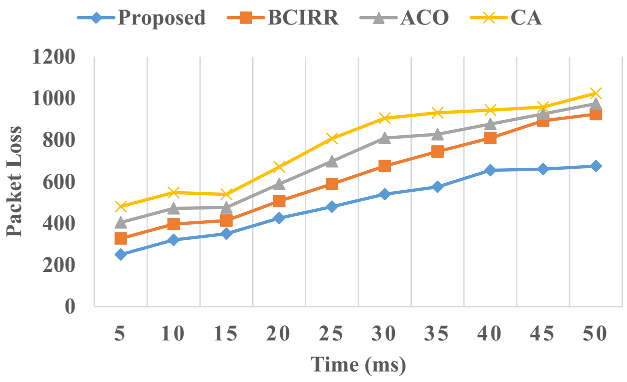


Fig. 15 Evaluation of Packet loss in the proposed method

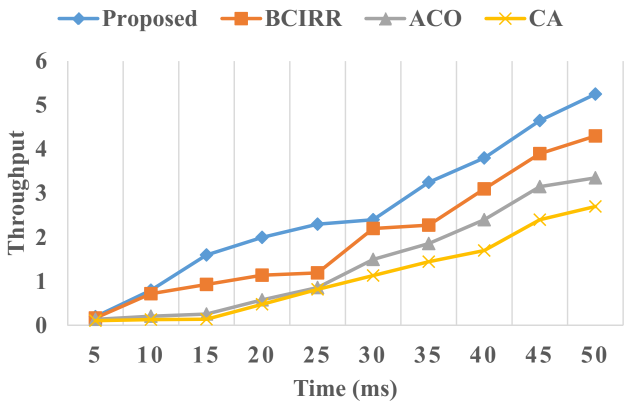


Fig. 16 Evaluation of Throughput in the proposed method

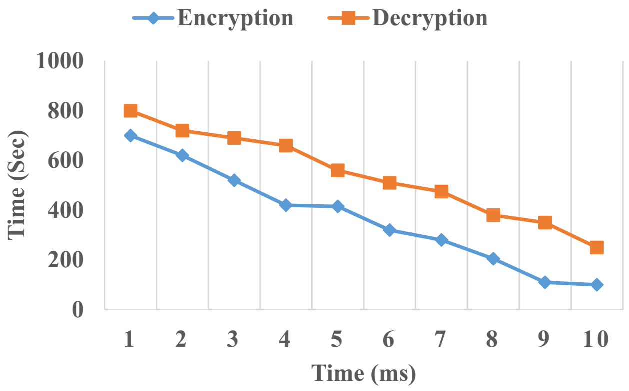


Fig. 17 Evaluation of Encryption and decryption time in the proposed method

Table 3 Encryption and decryption comparison

Time	Proposed CSO (seconds)		DNA approach [42] (seconds)		Symmetric method [43] (seconds)	
	Encryption	Decryption	Encryption	Decryption	Encryption	Decryption
2	600	720	760	860	820	890
4	400	650	440	678	530	735
6	350	540	380	590	420	640
8	270	400	310	470	330	490

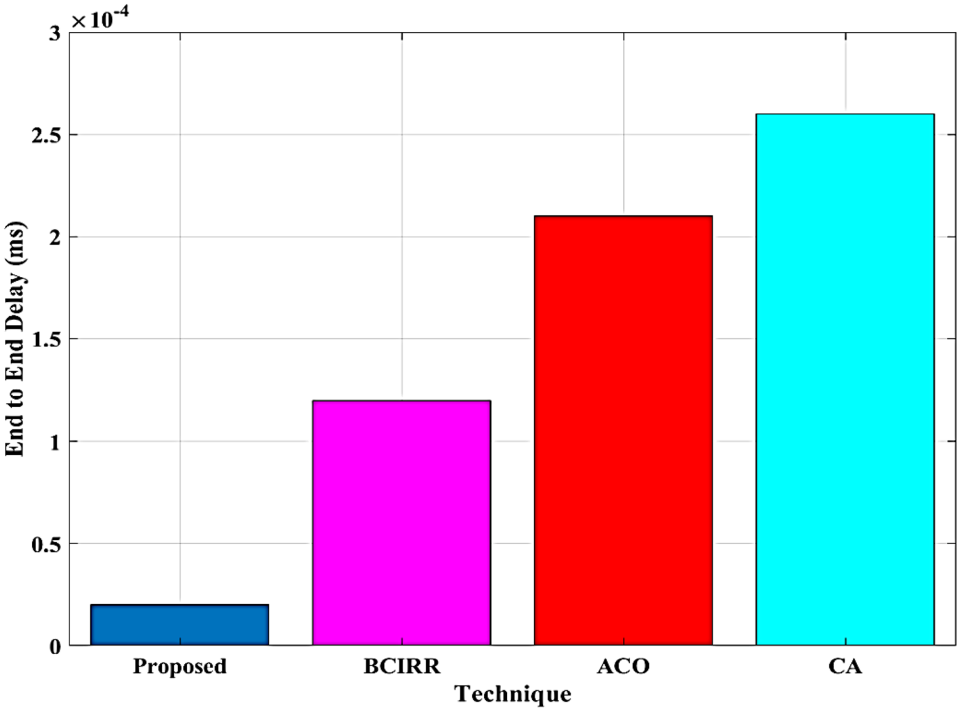


Fig. 18 Comparison of delay in the proposed method

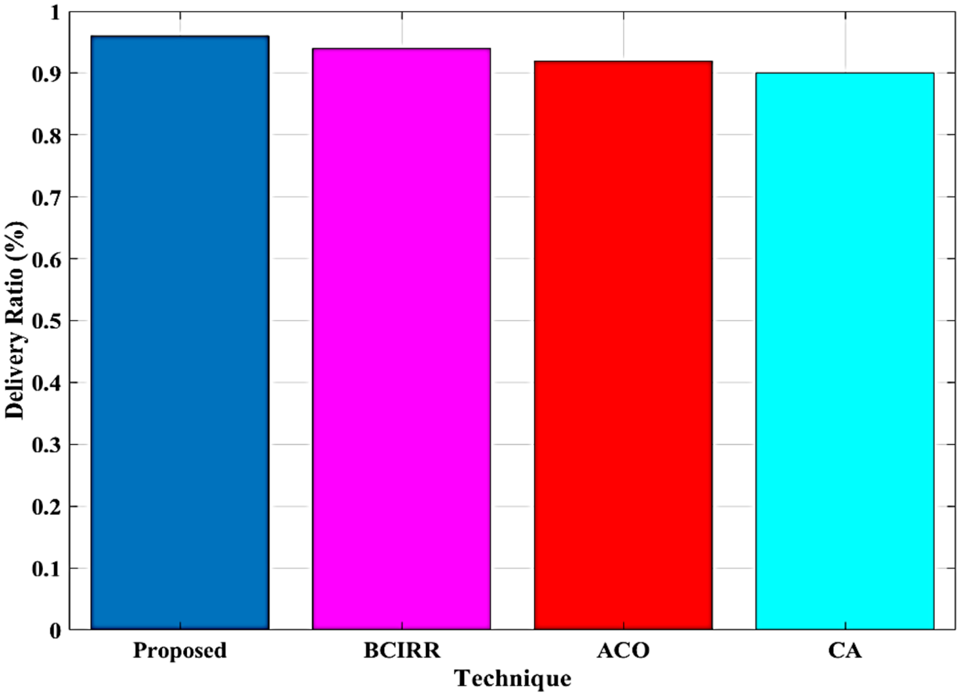


Fig. 19 Comparison of Delivery ratio in the proposed method

Fig. 20 Comparison of Packet loss in the proposed method

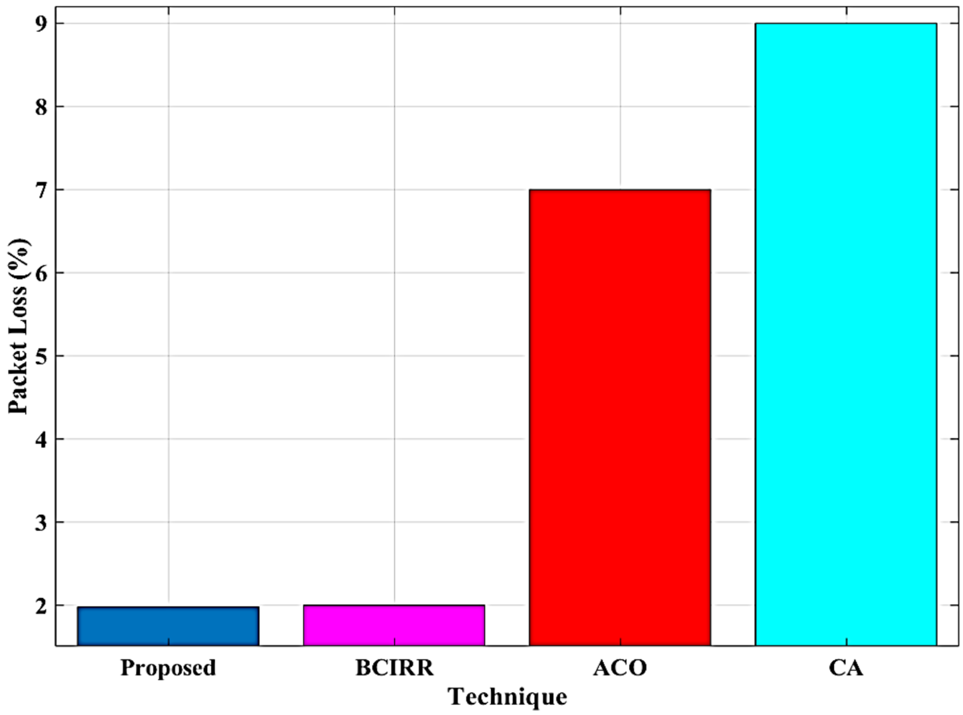


Fig. 21 Comparison of Throughput in the proposed method

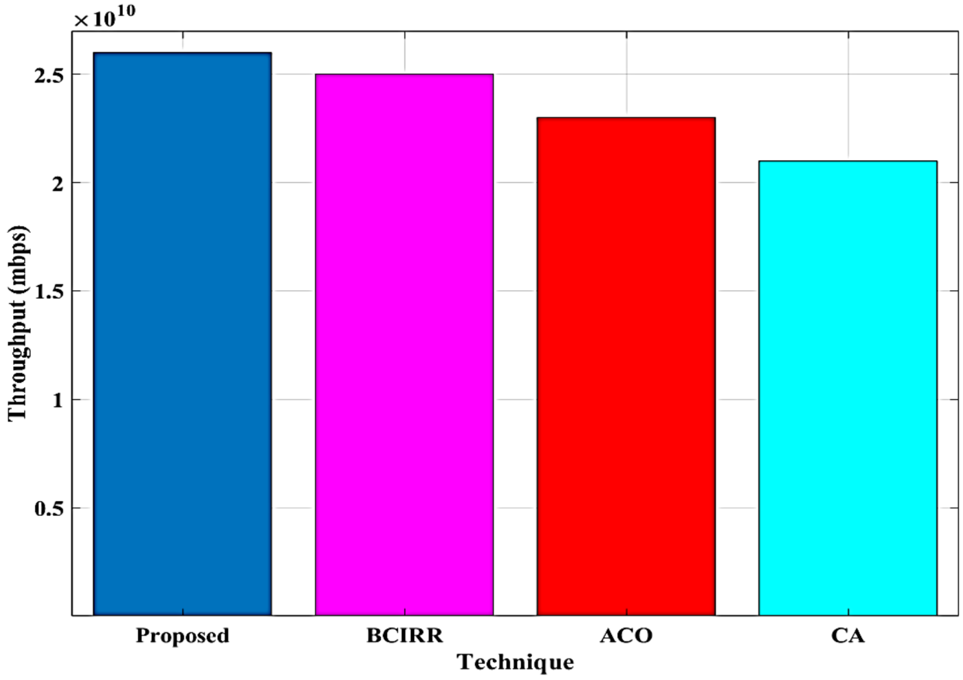


Table 4 Execution time of proposed and existing approaches

Techniques	Execution time
Proposed CSO	350 ms
BCIRR	398 ms
ACO	430 ms
CA	500 ms

Table 5 Comparison of proposed and existing method performance

Performance	Proposed CSO	Cluster-based replication architecture
Throughput	2.6×10^{10} Mbps	1 Mbps
Delay	0.20×10^{-4} sec	0.008 s
Packet loss	1.97	0

5 Conclusion

In this paper, probability-based load balancing control and security are developed in P2P networks. The load balancing control is enhanced with the utilization of CSO by probability condition. The proposed method works based on two main objective functions: load balancing control and security enhancement. The load balancing and security control are processed with four sections as query sends to neighboring peers, a response from neighbour peers, best peer-selected based on probability with CSO, finally, security and load balancing are achieved. Here, the CSO algorithm selects the best peer by a probability function, which contains constraints of price, response time, availability, and reputation. The selected peer is checked with the ERR technique to check the attacker or not. The resources of each peer can be securely stored with the help of the SXOR operation. The proposed method is implemented in the NS2 platform, and performances are analysed with performance metrics such as delay, delivery ratio, throughput, packet loss, encryption time, and decryption time. The proposed technique reaches a 98.75% packet delivery rate, with a delay of at least 3.8 s. The proposed method is compared with existing methods such as BCIRR, ACO, and CA methods. From the comparison analysis, the proposed method has the best performance in terms of performance metrics which most suitable approach for increase resource utilization and security enhancement in P2P networks. Parallelism is not taken into account during the evaluation. As a result, employing peers or a similar simulation toolkit for parallelism computation will be highly interesting in the future. Additionally, using meta-heuristic algorithms for load balancing, waiting time minimization, and execution time reduction will be fascinating.

Author contribution The corresponding author claims the major contribution of the paper including formulation, analysis and editing. The Second author guides to verify the analysis result and manuscript editing.

Funding This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest There is no conflict of Interest between the authors regarding the paper preparation and submission of a manuscript.

References

1. Dhungana A, Bulut E (2020) Peer-to-peer energy sharing in mobile networks: Applications, challenges, and open problems. Ad Hoc Netw 97:102029
2. Masinde N, Graffi K (2020) Peer-to-peer based social networks: A comprehensive survey. Preprint at arXiv:2001.02611
3. Djellabi B, Younis M, Amad M (2020) Effective peer-to-peer design for supporting range query in Internet of Things applications. Comput Commun 150:506–518
4. He H (2020) A reliable peer-to-peer storage framework based on the virtual peers model. Int J Networking Virtual Organ 22(2):129–146
5. Hou WJ, Jiang Y, Lei W, Xu A, Wen H, Chen S (2020) A P2P network-based edge computing intelligent grid model for efficient resources coordination. Peer-to-peer Netw App 1–12
6. Khalid R, Javaid N, Almogren A, Javed MU, Javaid S, Zubair M (2020) A blockchain based load balancing in decentralised hybrid P2P energy trading market in smart grid. IEEE Access
7. Sina M, Dehghan M, Rahmani AM, Reshadi M (2020) WidePLive: A coupled low-delay overlay construction mechanism and peer-chunk priority-based chunk scheduling P2P live video streaming. IET Commun 14(6):937–947
8. Premarathne US, Rajasingham S (2020) Trust based multi-agent cooperative load balancing system (TCLBS). Futur Gener Comput Syst
9. Özsu MT, Valduriez P (2020) Peer-to-peer data management. In: Principles of Distributed Database Systems. Springer, Cham, pp 395–448
10. Berenjian S, Hajizadeh S, Hatamian M, Atani RE (2019) An incentive security model to provide fairness for peer-to-peer networks. Preprint at arXiv:1906.09355
11. Goswami A, Gupta R, Parashari GS (2017) Reputation-based resource allocation in P2P systems: A game theoretic perspective. IEEE Commun Lett 21(6):1273–1276
12. Li S, Sun W (2016) A mechanism for resource pricing and fairness in peer-to-peer networks. Electron Commer Res 16(4):425–451
13. Alhussain A, Kurdi H (2018) EERP: An enhanced eigen trust algorithm for reputation management in peer-to-peer networks. Procedia Computer Science 141:490–495
14. Jamal AA, Teahan WJ (2017) Alpha multipliers breadth-first search technique for resource discovery in unstructured peer-to-peer networks. Int J Adv Sci Eng Inf Technol 7(4):1403–1412
15. Han Q, Wen H, Feng G, Wu B, Ren M (2016) Self-nominating trust model based on hierarchical fuzzy systems for peer-to-peer networks. Peer-to-Peer Netw App 9(6):1020–1030

16. Singh M, Kumar C, Nath P (2020) Finger forwarding scheme to reduce lookup cost in structured P2P networks. Wireless Pers Commun 114:2263–2281
17. Masinde N, Bischoff S, Graffi K (2020) Capacity management protocol for a structured P2P-based online social network. In: 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE, pp 1–8
18. Shu Y, Zhu F (2020) An edge computing offloading mechanism for mobile peer sensing and network load weak balancing in 5G network. J Ambient Intell Humaniz Comput 11(2):503–510
19. Rahmani M, Benchaïba M (2019) PCSM: an efficient multihop proximity aware clustering scheme for mobile peer-to-peer systems. J Ambient Intell Humaniz Comput 10(11):4243–4260
20. Gomathi S, Manimegalai D (2019) Hierarchically distributed peer-to-peer architecture for load balancing and effective dynamic group scheduling in grid computing. Int J Bus Inform Syst 32(3):312–323
21. Milani AS, Navimipour NJ (2016) Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. J Netw Comput Appl 71:86–98
22. Neghabi AA, Navimipour NJ, Hosseinzadeh M, Rezaee A (2019) Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network. Int J Commun Syst 32(4):e3875
23. Seo JH, Kim YH (2020) A peer load balancing method for P2P-assisted DASH Systems. J Broadcast Eng 25(1):94–104
24. Mohammadi B, Navimipour NJ (2019) Data replication mechanisms in the peer-to-peer networks. Int J Commun Syst 32(14):e3996
25. Shen L, Wu J, Wang Y, Liu L (2018) Towards load balancing for LSH-based distributed similarity indexing in high-dimensional space. In 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, pp 384–391
26. Qi X, Qiang M, Liu L (2020) A balanced strategy to improve data invulnerability in structured P2P system. Peer-to-Peer Netw and App 13(1):368–387
27. Rguibi MA, Moussa N (2018) Hybrid trust model for worm mitigation in P2P networks. J Inform Secur App 43:21–36
28. Chuang Y-T, Li F-W (2020) TCR: a trustworthy and churn-resilient academic distribution and retrieval system in P2P networks. J Supercomput 1–33
29. Elrotub M, Bali A, Gherbi A (2021) Sharing VM resources with using prediction of future user requests for an efficient load balancing in cloud computing environment. Int J Softw Sci Comput Intel (IJSSCI) 13(2):37–64
30. Manasrah AM, Gupta BB (2019) An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. Clust Comput 22(1):1639–1653
31. Ahuja SP, Czarnecki E, Willison S (2020) Multi-factor performance comparison of amazon web services elastic compute cluster and google cloud platform compute engine. Int J Cloud App Comput (IJCAC) 10(3):1–16
32. Chui KT, Gupta BB, Liu RW, Vasant P (2021) Handling data heterogeneity in electricity load disaggregation via optimized complete ensemble empirical mode decomposition and wavelet packet transform. Sensors 21(9):3133
33. Ali S, Banerjea S, Pandey M, Tyagi N (2019) Towards DHT-Based P2P Resource Sharing Over Hybrid Infrastructure of Wireless Mesh Network and Mobile Ad hoc Networks. In Computing and Network Sustainability, Springer, Singapore, pp 147–155
34. Asghari S, Navimipour NJ (2019) Cloud service composition using an inverted ant colony optimisation algorithm. Int J Bio-Inspired Comput 13(4):257–268
35. Souri A, Navimipour NJ (2014) Behavioral modeling and formal verification of a resource discovery approach in Grid computing. Expert Syst Appl 41(8):3831–3849
36. Asghari S, Navimipour NJ (2019) Resource discovery in the peer to peer networks using an inverted ant colony optimisation algorithm. Peer-to-Peer Netw App 12(1):129–142
37. Wang B, Li W, Chen X, Chen H (2019) Improved chicken swarm algorithms based on chaos theory and its application in wind power interval prediction. Math Prob Eng 2019
38. Deb S, Gao X-Z, Tammi K, Kalita K, Mahanta P (2020) A new teaching–learning-based chicken swarm optimization algorithm. Soft Comput 24(7):5313-d5331
39. Christo MS, Meenakshi S (2016) Reliable and authenticated rumor riding protocol for unstructured peer-to-peer network. Indian J Sci Technol 9(21):1–9
40. Christo MS, Meenakshi S (2018) Enhancing rumor riding protocol in P2P network with cryptographic puzzle through challenge question method. Comput Electr Eng 65:122–138
41. Li Y, Gai K, Qiu L, Qiu M, Zhao H (2017) Intelligent cryptography approach for secure distributed big data storage in cloud computing. Inf Sci 387:103–115
42. Siddaramappa V, Ramesh KB (2019) DNA-Based XOR operation (DNAX) for data security using DNA as a storage medium. In Integrated Intelligent Computing, Communication and Security, Springer, Singapore, pp 343–351
43. Anupriya E, Soni S, Agnihotri A, Babelay S (2011) Encryption using XOR based extended key for information security–a novel approach. Int J Comput Sci Eng (IJCSE) 3(1):146–154
44. Ayyasamy S, Sivanandam SN (2010) A cluster based replication architecture for load balancing in peer-to-peer content distribution. Preprint at arXiv:1009.4563

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dharmendra Kumar received his M.Sc. in Computer Science from the University of Allahabad and his M.Tech. in Computer Science from Motilal Nehru National Institute of Technology Allahabad, Prayagraj. He is an associate professor in Computer Science department at the United College of Engineering and Research, Prayagraj. He is doing a Ph.D. from Dr. A.P.J. Abdul Kalam Technical University, Lucknow. His research areas are Peer-to-Peer Computing, Cloud Computing, and Machine Learning. His early research work focused on the development of resource discovery algorithms in Mobile P2P network.

Mayank Pandey obtained his PhD degree from Motilal Nehru National Institute of Technology Allahabad in 2012. He is currently working as Associate Professor in the Department of CSE, Motilal Nehru National Institute of Technology Allahabad, 211004, India. At present, he is guiding five PhD students in the field of software defined networking (SDN), ad-hoc peer to peer networks, IP mobility, fog computing, internet of things (IoT), and agent-based modeling. He is having the membership of IEEE and ACM. His areas of expertise are SDN, P2P networks, distributed computing, wireless/mobile networks and formal methods. He has been awarded with Young Faculty Research Fellowship by Government of India.