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

Optimal Data Scheduling in Hybrid CDN-P2P Video on Demand System

Deepali Tatyarao Biradar *[1], Sudhir N. Dhage[2]

Submitted: 26/01/2024 **Revised:** 04/03/2024 **Accepted:** 12/03/2024

Abstract: The proliferation of various video applications like video telephony, video conferencing and VoD (Video on Demand) has led to a heterogeneous user base on the Internet. VoD systems empower users with efficient control over video access, enabling them to stream content at their convenience. A Hybrid Peer-to-peer CDN based video on demand systems is an attractive solution for VoD service providers. It combines a traditional CDN with a P2P overlay for exchanging video chunks, reducing the cost of the CDN. However, P2P overlay in VoD systems face challenges, including issues like increase in start-up delay for playing video, underutilization of peer resources, and excessive server load, significantly impacting viewers' quality-of-service (QoS). Efficient data scheduling emerges as a pivotal concern among these challenges, focusing on the effective transmission and dispatching of data segments within the system. This paper introduces a data scheduling approach using swarm intelligence optimization algorithm -Crow Search Algorithm (CSA). The proposed approach first establishes priorities for chunk requests to ensure prompt handling of high-priority requests. By leveraging the CSA, the optimal peer among the neighbour which caches the requested chunks is selected as the parent for the chunk. The effectiveness of the proposed approach is assessed through simulations, revealing superior performance compared to existing methods and resulting in enhanced system efficiency.

Keywords: *crow search algorithm, data scheduling,* hybrid *CDN P2P, parent selection, Video on Demand.*

1. Introduction

In recent years, there has been a substantial surge in the delivery of video content through the Internet infrastructure. Initially, video content was captured and transmitted in the analog form. However, with advancements in Internet technology and devices, there has been a notable expansion in multimedia services and applications. The escalating demand for multimedia data over the Internet can be attributed to its widespread use in various sectors such as communication, education, entertainment, and public safety [1]. Video streaming has emerged as a widely adopted method of communication, including live video streaming and video on demand. In case of live video streaming, users can only enjoy the content at a specific transmission schedule, while video on demand allows users to access and enjoy the content at their convenience [2]. The past few years have seen a significant surge in the demand for media content, driven by the accessibility of platforms like YouTube NetTV, Netflix, and iflix. These platforms are prominent players in the Video on Demand Market. Video-on-Demand (VoD) stands out as a widely embraced application with a substantial viewership, distinguishing itself from video live streaming. Video-on-Demand (VoD) applications have a broader range of data compared to live streaming platforms. In live streaming, nodes focus on a specific playback point, catering to users watching simultaneously. However, VoD systems experience nodes requesting videos at various times, creating diverse playback points. This difference in data patterns distinguishes the two systems. Consequently, interactive requests such as Back-Ward (BW) or Fast-Forward (FF) are absent, rendering them more akin to a broadcast service, which is trivial [3]. The significant content size and widespread usage of VoD applications emerge as primary challenges in the VoD content distribution problem [4]. A VoD system usually consists of media server and distributed clients over the entire network. Server stores pre-recorded videos and client can access that by request. The VoD systems need to be able to handle several viewers watch the same video asynchronously in an efficient manner. At any given time, users may be viewing different parts of the same video. The design of VoD system necessitates substantial server resources and significant bandwidth to support its user base. In the centralized architecture, the server serves as the sole source capable of meeting the demands of each user. The server's upload capacity is limited, and as the number of users' increases, upload capacity decreases, leading to a decline in the overall network performance. With the centralized client-server architecture used for video streaming, the complexity and cost of the server side expand exponentially with the number of users. To cope up with rising demand distributed architecture based VoD systems evolved. Users expects high QoS (Quality of Experience) in terms of uninterrupted and quality video delivery, so there is a lot of load on the servers to meet the requirements with the available resources such as network bandwidth and storage capacity. Various studies in recent years have focused on optimizing Video-on-Demand content distribution while minimizing server load. One approach is to move away from the traditional single-server system, which may struggle to meet VoD demands. An effective solution proposed is the use of Content Delivery Networks (CDN) with a multicast

1 Research Scholar, Rajiv Gandhi Institute of Technology, Mumbai, Maharashtra, India.
ORCID ID: 0009-0003-5708-9640
Email: deepaliktpatil@gmail.com
2 P Professor, Sardar Patel Institute of Technology, Mumbai, Maharashtra, India.
ORCID ID: 0000-0001-8100-8941
** Corresponding Author Email: deepaliktpatil@gmail.com*

schema. This helps distribute the server load by employing multiple servers, addressing the challenge of efficiently serving VoD clients. Content Delivery Network (CDN) technology [5] [6] emphasizing on service quality. CDN technology reduce the content and user management by distributing among various servers in hierarchy, yet it has challenges such as the user base for downloads grows, network encounters challenges related to expansion and increased costs.

In P2P-based VoD systems, each user has the ability to download and watch videos from either other peers or server. Additionally, they can share the video segments they have downloaded with other users. Unlike traditional systems that heavily rely on the server, P2P networks utilize the upload capacity of individual users. This means each user functions as both a server and a receiver, boosting the system's reliability, service capabilities, and reducing server load and network costs. Thus users can upload and download the data simultaneously and enhances the overall performance of the network [7]. In a P2P network, the processes of adding and removing peers are relatively straightforward compared to a client server architecture. Unlike in a client server setup where the addition or deletion of a new users (peers) can impact the supply for other users (peers), in a P2P system, these actions do not have adverse effects on the server's performance. A hybrid architecture that combines elements of both client-server and P2P networks has been proposed in [8] [9] aiming to take advantage of both architectures. There exists a complementary relationship between the effective scalability and service efficiency provided by P2P technology and CDN technology within the computational model [10]. In [11] authors showed that the integration of these two types of architectures can enhance the coordination between network transmission and bandwidth resource consumption, addressing the challenges associated with these aspects more effectively.

Two most challenging tasks in a P2P network which shows effectiveness of P2P video streaming system are Overlay construction and data scheduling. In the initial phase, when peers enters into the system, an overlay network is created by arranging peers in some structure. This overlay allows all the peers to collaboratively share video resources. The overlay network architecture determines how video data is distributed among peers. Typically overlay networks are categorized into- tree based and mesh-based overlay. In P2P tree-based systems, peers are structured in a tree format to facilitate the delivery of streams. In this setup, each peer gets required video data from its parent and transmits the data to its children. However, a downside to this setup is that if the parent peer encounters a problem, it prevents all its child peers from receiving the video parts [12] . In mesh based overlay of P2P systems, peers operate on an equal level without a hierarchical structure. Each peer connects with several neighbours, limiting its data exchanges to these connections. This mesh overlay method addresses the constraints found in tree-based systems, offering a more flexible and robust approach to data distribution. A mesh overlay network can be full or partial. In a full mesh, each peer connects with every other peer in the network. In a partial mesh, peers connect to a selected number of other peers as needed. This structure enhances reliability, especially when dealing with peer departures (churn), because if one connection drops, a peer can still access data from other connected neighbours. Mesh overlays are favoured for their scalability, straightforward setup, resilience to changes in peer presence, and effective use of bandwidth. [13] [14] [15].

The second essential aspect in P2P overlay of VoD system is data scheduling, where it involves establishing the sequence for requesting and downloading video chunks from peers. This mechanism forms a significant part in the video streaming process. Effectiveness of this data scheduling greatly influences the performance of the entire system. Data scheduling methods in P2P overlay network is broadly classified into push-based and pull-based systems.. Tree-based systems are commonly structured with a push-based approach [16] [17] [18], wherein video chunks are transmitted from the parent to children along the tree architecture. In contrast, the pull mechanism employed in fully mesh or partially mesh-based systems relies on the availability of chunks at neighbouring peers. This mechanism determines which chunks are accessible from specific neighbours. Thus, a receiving node is tasked with identifying the missing chunks and requesting them from the relevant nodes [19] [20]. The pull mechanism is thought to be a very straightforward and appropriate method since it enables the receiver to handle the two primary issues of removing redundant chunks and recovering from lost chunks. But because node has to select the right chunks to be chosen from the right neighbour, it complicates things on the receiving end. In the context of video on demand, scheduling becomes challenging because chunks received beyond their playback date result in lengthy playback delays and lower service quality.

Efficient data scheduling is crucial to ensure seamless playback and optimal utilization of network resources in P2P based systems. As use of P2P in VoD system continues to evolve and gain popularity, further advancements in data scheduling techniques are necessary to address the challenges of scalability, heterogeneity, and dynamic network conditions, ultimately enhancing the streaming experience for users in P2P based VoD systems.

In this work, our focus is to provide an optimal data scheduling approach in hybrid CDN based P2P VoD systems. We present our contribution for data scheduling using crow search algorithm. Results of simulation show that our algorithm is effective and improves the QoS of VoD system.

2. Related Work

Data scheduling in P2P VoD plays a vital role in ensuring efficient content delivery, minimizing latency, and maximizing resource utilization. In the context of a P2P mesh network, the overlay construction strategy fulfils specific quality requirements, such as minimizing delay, maximizing throughput, or ensuring resiliency. However, the overlay construction strategy does not impose how to utilize the overlay. In pure P2P environment, each peer can function as both a server and a client simultaneously and thus there is no dedicated server. The peer's buffer stores available chunks, facilitating their exchange with other peers. The receiving peer must obtain unavailable chunks before playback deadlines, aiming to reduce start-up buffering delays. So it is very important while selecting parent peer for chunks and there are various factors to be consider as not only single factor contributes to maximize the system performance. Like for maximizing throughput, a node might possess parents with abundant upload bandwidth. However, if these parents lack sufficient or suitable content means chunks to send and thus due to improper selection of parent causes chunks to arrive beyond their playback deadline thus the actual observed throughput may fall below anticipated levels. Additionally, when a receiving node predominantly requests most chunks from a single parent, its sensitivity to the departure or failure of this particular parent increases, potentially leading to significant quality degradation. Consequently, the system exhibits weak resiliency in such scenarios.

The data scheduling algorithm determines which chunks to request from which peers based on different criteria, such as peer bandwidth, chunk availability, and playback buffer occupancy. It aims to optimize the scheduling order to maximize the utilization of network resources, reduce redundant data transfers, and ensure smooth video playback without significant buffering or interruptions. There have been various studies on data scheduling for improving the performance of video streaming system. In a study outlined in [21], the authors proposed an approach that accomplishes the scheduling of video sharing among peers based on a scheme of dynamic buffering progress. In this approach, a receiver selectively receives chunks exclusively from sender peers with comparable playback progress. The aim is to achieve optimal peer bandwidth utilization while minimizing system maintenance costs. However they considered that leeches will be a part of system until the downloading is complete but in reality they may leave the system in-between.

To address the parent peer selection problem, a distinct peer selection algorithm, called "Closest Playback Point First," is introduced in [22].This innovative approach involves sending requests to peers with playback points closest to the requester, fostering the grouping of peers with similar available content. This strategy aims to achieve a more balanced and collaborative exchange of pieces among peers. Similarly, Liu P. et al. [23] proposed data scheduling approach called an event driven high-priority first to address load balance and contention of bandwidth issues resulting from peer churn. However, these scheduling schemes lack specific bandwidth allocation methods to fully optimize the use of peers' available bandwidth. In [24], the smallest queue size strategy is employed to choose nodes with the smallest load for sending requests, aiming to prevent overloading issues. However, this strategy lacks consideration of the urgent request and number of sources available for requests. As priority for urgent requests are not given, which results in broken playback. Addressing this limitation, a load balancing strategy is introduced in [25]. In their work, the authors presented a load balancing technique termed Request Queue Load Balancing for scheduling in a P2P-VoD system. Their approach involves determining the priority of requests by considering urgency and scarcity factors. This ensures that high-priority requests are addressed promptly. Additionally, peer selection is performed based on considerations such as upload bandwidth, reliability, and load degree. Another method which is proposed in [26] focuses on dynamically switching between two strategies instead of using single static rules for load allocation. This approach optimally adapt to observe and predicted shifts in content popularity and based on it switches the load allocation strategy so that upload bandwidth of all peers will be utilized. Another study[27]aims to adapt the Bit-Torrent protocol for use in multimedia on-demand streaming systems emphasis lies on peer selection policies. In this research author proposed three different policies for neighbor selection namely, Select Balanced Neighbour Policy, Select Regular Neighbour Policy, and Select Optimistic Neighbour Policy with the objective of enhancing the scalability of the system. In [3], the Least Loaded Peer (LLP) policy is introduced for selection of peer, aim was to optimize the utilization of peers upload capacity. Another research contribution to the neighbor selection was by Huang et. al. [28], they proposed a gossip protocol for peer selection based on value of performance function of a peer. Additionally, [29] proposes a distributed peer selection algorithm with the goal of balancing traffic load among peers, thereby enhancing playback quality in P2P streaming networks. A max-flow-based optimal data scheduling technique was presented by Q. Yu et al. [30] for P2P VoD system. Reducing server load and improving playback continuity quality were the goals. The authors used streaming data scheduling techniques based on polynomial max-flow to optimize data scheduling in order to achieve this. On the other hand, system throughput and scheduling time still have space for improvement. An effective replication technique of chunk for improving QoS in heterogeneous P2P VoD networks of both fixed mobile and peers was presented by the authors in [31], where peer ranking factor (PRF), local demand based chunk replication (LDCR) and local demand-based chunk download (LDCD) are the three primary algorithms that make up the enhanced chunk regulation algorithm (ECRA).To serve clients based on chunk availability, the local demand-based chunk download algorithm manages selection and prefetching of chunk in the sliding window. In essence, a request for a regular chunk download triggers LDCD. The PRF algorithm uses each peer's PRF value to determine which peers should download video chunks. Therefore, only peers with strong signals, ample energy, and sufficient bandwidth are added to the list of source peers from whom video chunks can be downloaded. The LDCR method, on the other hand, determines which piece needs to be copied next. The most demanded first" rule is applied in this process to determine which portion is appropriate for replication. However, the concern of source server downloads and missed chunks increments with the increase in mobile peers joining the network was not considered.

Thibaud Rohmer et al.'s proposed algorithm [19] used Evidence theory for dynamic strategy selection which is less sophisticated learning method to improve the P2P VoD system performance. Nonetheless, there is a requirement for further enhancement in the efficiency of the system, especially under heavy demand.

Nature-inspired meta-heuristics algorithms aim to achieve the global optimal solution by investigating the most suitable places in the search space domain using natural mechanisms. These approaches differ depending on the type of algorithm, as some algorithms are suitable for tackling certain types of issues but not for others. This is because certain NP-Hard problems are solved using stochastic or randomized techniques. Numerous earlier studies have put forth a variety of meta-heuristic-based strategies for scheduling workflow applications. In [33], an efficient data scheduling algorithm based on Artificial Bee Colony (ABC) is proposed. The approach involves determining the priority of segments, and then using the ABC algorithm to select an optimal peer as the parent peer. After determining the priority of a chunk, the prioritized chunk is requested from the peer selected using the Artificial Bee Colony algorithm. Ajitha Gladis [34] suggested a data scheduling solution for hierarchically formed P2P VoD systems using physics based algorithm- gravitational search algorithm which is inspired by Newton’s law of gravity and the law of motion, is utilized for the best parent selection for the selected chunk request, providing an alternative nature-inspired algorithm for data scheduling.

The authors achieved their goal by presenting various data scheduling approaches from the literature that was previously evaluated. Yet the system still has to be made even more efficient. So, in order to improve performance, optimal peer selection using CSA algorithm is proposed in this paper. Based on the peers' capabilities, this algorithm effectively chooses the best peer for chunk request.

3. Proposed System

In this section, we describe the hybrid CDN-P2P Video on Demand System model used in our work and the proposed crow search based data scheduling algorithm in detail.

3.1. System Model

Before exploring the proposed work on the data scheduling strategy, it is important to introduce the Video on Demand (VOD) system used in this work. To address the cost challenges associated with deploying CDN servers and dynamic issues in P2P, a CDN-P2P hybrid system is adopted. This hybrid system combines the advantages of both CDN and P2P technologies, offering benefits such as high-quality transmission, efficient handling of extensive streaming, and a reduction in the costs associated with CDN deployment. The top-level architecture utilizes a CDN to provide content closer to users, thereby minimizing delays. Simultaneously, a P2P overlay is employed in the lower layers of the architecture to ensure low latency during streaming. This dual approach not only enhances the user experience by reducing delays but also alleviates the load on the CDN, leading to cost savings.

The design of our system defines a hybrid CDN P2P based VOD network model by configuring the network in three different layers as shown in figure 1. The first level of system is source media Server layer. The media server connects with the edge servers which forms the second layer. The third level specifies the terminal layer of end users or peers. These nodes in the system forms a mesh type of overlay to utilize the upload bandwidth efficiently and improve the system performance.

Thus the hybrid CDN P2P based VOD System comprises main entities: Source Media Server, Edge Servers with tracker and end nodes (Peers).

- Source Media Server:

  The Source Media Server serves as the origin of content, housing a permanent copy of a set of videos. It serves as the foundation for content availability, delivering videos to edge servers and requesting end nodes. All videos are first compressed and divided into small units called chunks and stored on the Media Server layer. The media server connects with the edge-server which forms the second layer in the system. Media server provides the content to edge servers over the network and also can serve end users requests.
- Edge Servers:

  The edge servers placed at different strategic locations to serve the users with minimum access delay. Each edge server contains subset of source media files due to storage constraints and serves user requests. In case of requests for videos that are not available with it are obtained from the source media server and serves the corresponding peers in his region. Tracker is associated with edge server which contains the record of all peers served by that edge server. This record includes information about peers like identifier, joining time, playback point, upload bandwidth and download bandwidth and its location.
- End Nodes (Peers):

  End nodes requests are served by the edge server to which they are connected or peers in the overlay. Peers establish mutual connections to form partner relationships and construct an overlay. During connection establishment phase peer determines the distance with all its neighbors in terms of number of hops and keep records. In our system, peers are organized in a mesh overlay, where peers watching the same video are grouped together. The mesh-based overlay comprises two key components: the local buffer and the media player, facilitating video playback. Each peer fetches chunks from the server or other peers and stores them in a local buffer. The local buffer then feeds the fetched video chunks into the media player, which decodes and plays the video on the end-user side.

When a peer enters into the system to watch video, based on its closeness it will be directed to nearest edge server then first it interacts with the trackers with its details and request to identify its neighbours to form an overlay. The overlay structure in our system model falls into the category of mesh-based overlay. When peers exits the system, it frees from its existing neighbours and in case of jump operation, it abandons its current playback point and abandons its current set of neighbours. Subsequently, the peer changes its playback point and request tracker for new set of neighbours. Also in case of less number of neighbours, the peer periodically communicates with the trackers to acquire new neighbours. If a peer is unable to obtain chunks from neighbour then it reaches out to the edge servers for the necessary video data.

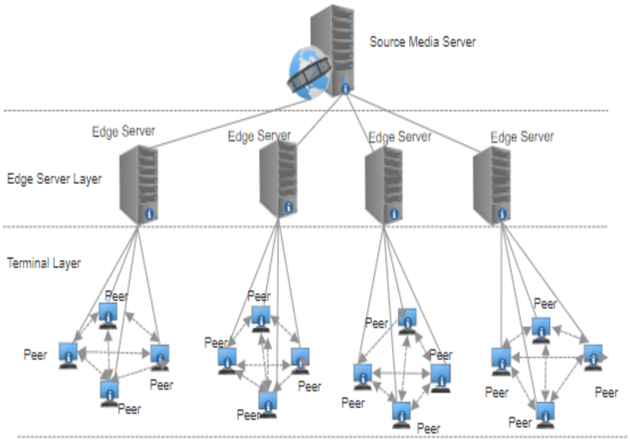

Fig 1: Hybrid CDN P2P Based VOD System Architecture

3.2. Overview of Crow Search Optimization (CSA) Algorithm

The interest among researchers has grown in the development of methodologies capable of addressing a diverse array of problems in a generic manner. This has led to the emergence of meta-heuristics, a specialized category of intelligent optimization techniques[35]. Meta-heuristics leverage a set of generic techniques and abstract concepts to iteratively enhance a set of potential solutions. Consequently, these approaches are often praised for their ability to converge towards optimal solutions for a majority of problems, irrespective of their specific design or characteristics. Achieving a well-balanced blend of diversification and intensification is crucial for the effectiveness of meta-heuristic algorithms. These two components, diversification, and intensification, constitute integral aspects of any meta-heuristic algorithm. The capacity of the algorithm to generate a range of solutions by globally scanning the search field is known as diversification. On the other hand, intensification focuses on directing the search activity inside the local space, knowing that this is where the solution is more likely to be located. A higher rate of convergence is achieved when diversity and intensification are used in a balanced manner to guarantee the achievement of the optimal solution and global optima. Effectively implementing a meta-heuristic algorithm requires careful adjustment of its parameters. However, parameter tuning is a time-consuming process. Consequently, in various optimization scenarios, algorithms with a limited number of parameters prove to be more straightforward to implement. CSA (Crow Search Algorithm) stands out as a recent addition to the meta-heuristic algorithms. A comparative analysis conducted by authors [36] has thoroughly examined CSA in contrast to several other meta-heuristic algorithms. Notable algorithms included in this analysis are

Particle Swarm Optimization, Grey Wolf Optimization, Sine Cosine Algorithm and Bat Algorithm. Crows are widely acknowledged as the most intelligent birds, possessing a remarkably large brain relative to their body size. Crow intelligence has been demonstrated by numerous evidences. Askarzadeh has introduced a ground breaking meta-heuristic optimization algorithm- CSA algorithm inspired by the social behavior of crows [37] . The concept of CSA is inspired by how crows store extra food in secret spots and retrieve it when needed. Crows are clever—they watch other birds hide food, steal it when they're not around, and then hide to avoid being caught later. CSA is based on four main ideas:

(a) Crows live in flocks

(b) They remember where they hide their food

(c) They follow each other to steal and

(d) They use a probability to protect their hidden food from being stolen.

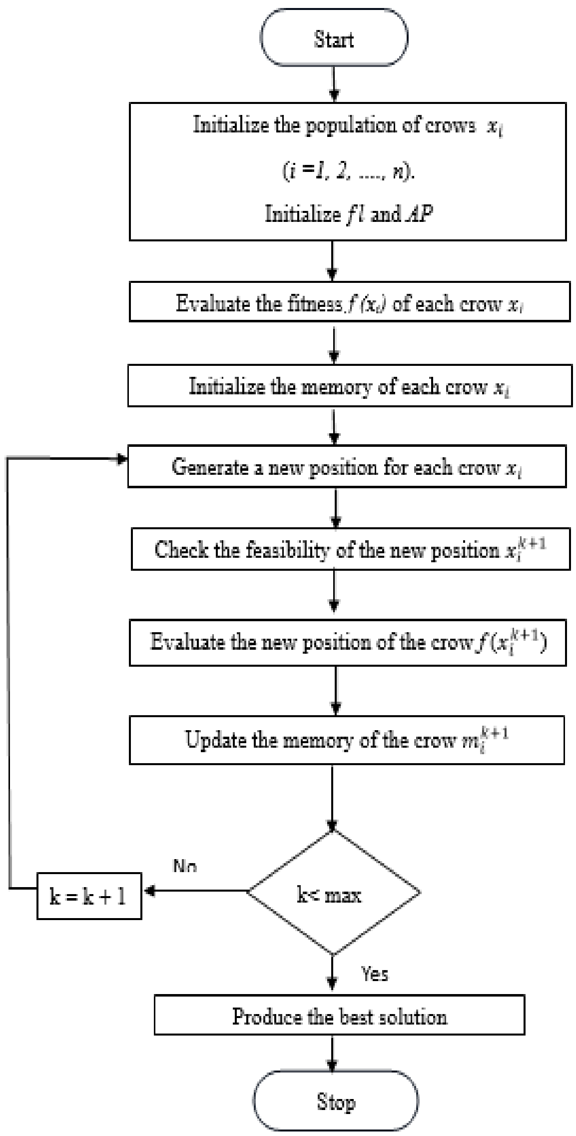

Fig 2: Flowchart of Crow Search Algorithm

Assuming that there are N crows in the flock, and that at iteration k, each crow *i* has a position indicated by x_i^k. The crows remember the hiding place of the food they are following thus the location where crow *i* hides its food is memorized. These crows move across the search plane, actively searching for the best food source, defined as - m_i^k.

Two likely possibilities arise from the searching strategy in CSA. The first is that the thief crow follows the owner crow j of food source m_j^k without the owner crow's knowledge, and as a result, the thief crow finds its hiding spot.

The CSA searching approach involves two potential scenarios. In the first scenario, the owner crow *j* of the food source m_j^k is unaware that the thief crow *i* is following it, allowing the thief crow to reach the hiding place of the owner crow. Equation (1) shows the updating process for the crow thief position.

$$x_i^{k+1} = x_i^k + r_i \times fl_i^k \times (m_i^k - x_i^k) \qquad (1)$$

Where,

r_i - Random number in range [0, 1] called approximation probability

fl_i^k - Flight length of crow *i* at iteration *k.*

m_i^k - Hiding place of food (memory) of cow *i.*at iteration *k.*

In the second likely situation, the owner crow j may know that the thief crow *i* is following it; as a result, the owner crow *j* may trick crow i by moving to a different location in the search space. A random position is used to update the position of Crow *i.* The two scenarios can be summarized as shown in equation (2)

The two scenarios can be summarized as shown in equation (2)

$$x_i^{k+1} = \begin{cases} x_i^k + r_i \times fl_i^k \times (m_i^k - x_i^k); & if\ r_i \geq AP_i^k \\ a\ random\ position; & Otherwise \end{cases} \qquad (2)$$

In the search space, the CSA algorithm proves to be highly efficient at identifying optimal solutions. Comparing CSA to most other widely used meta-heuristic techniques, it has the advantage that awareness probability and flight length these two parameters need to adjust. The awareness probability parameter in CSA primarily regulates the balance between intensification and diversification. Meanwhile, the second parameter- flight length plays an important role in determining the searching capability of crows, where larger values facilitate global search, and smaller values prompt local search.

3.3 Receiver-driven Data scheduling using Crow Search Optimization

To watch video peers need to get the chunk, and this action involves the data scheduling. Data scheduling can be further divided into two parts: chunk selection and parent peer selection.

When the peer decides which chunk to be requested, it will go to the next phases – source (parent) selection. A good source peer selection can dramatically accelerate the speed of chunk download, and minimize the networks resource consumption. In order to simultaneously improve the quality of service and the sender nodes resource consumption, our aim in this work is to approximate the ideal schedule. In the following, we describe our proposed approach in more detail.

When a peer enters into the system, it is connected to nearest edge server for serving its request then based on the video request id tracker finds the status of all nodes connected to the edge server and based on that request will be either served by edge server or other peer nodes. If incoming request is the first request for the video then node will be directly connected to edge server and served by it. In case of having some peers watching same video in the system then tracker will assign new incoming node to swarm. To serve request by other peers it needs neighbors so tracker introduce some peers watching the same video and playback point greater than playback point of incoming request as probable neighbors. Then peer send neighbor request to these nodes and establishes the connection and keeps record of neighbors such as joining time of peer, distance with it in terms of hops. The peer then sends its playback point, buffer map and upload bandwidth status to tracker periodically so tracker update the status of peers in the system. After joining overlay the peer begins to regularly receive video chunks from other peers after joining the swarm and offers its own upload capacity to broadcast the chunks that its neighbors want. Active partners are those that supply chunks, and passive partners are those that download chunks. The active partners of a peer sends chunks to requested peers. The peers may differ in their upload and download speeds. As a result, each peer may have a different amount of active and passive partners.

In our system the efficient parent selection for P2P data scheduling operates through a three -step process: Initially, peers across the network exchange buffer maps. Subsequently, chunks are prioritized and scheduled based on their estimated importance. Next, the efficiency of each neighboring peer is assessed to evaluate their potential as senders and one peer with the highest efficiency is chosen to transmit the requested chunk to the receiving peer which are detailed below.

Our system assumes that the video is divided into equal size segments known as chunks and coded with a constant bit rate (CBR). For first step of process-buffer map exchange - each peer manages a Buffer Map (BM) which is an array representing the availability of chunks in its local buffer. The BM is consisted of some bits with 0's and 1's and each bit is equivalent with a chunk position in buffer. Using equation (3) each peer finds missing chunks from sliding window.

$$BM\ (pi,k) = \begin{cases} 1 & chunk\ j\ is\ possessed\ by\ peer\ pi \\ 0 & else \end{cases} \qquad (3)$$

If *BM (pi, k)* =1, indicates that chunk k is possessed by peer *i*; if not, peer *pi* lacks chunk k.

Periodically, peers exchange their available upload bandwidth information, playback point and BMs with neighboring peers, so it knows which data chunks are available with neighbor peers.

In second step of process- chunk prioritization and scheduling- each peer keeps a sliding window of video chunks that periodically progresses as peer plays chunks from buffer. The chunk selection process works on a sliding window. A peer stores the already received video chunks and from this sliding window determines the presence and absence of video chunks. In this way, a peer can discover the availability of video chunks at its neighbors that are missing in its window. Subsequently, based on the received BMs from neighbors and considering chunk priorities for missing chunks, the peer determines which chunks to request for the continuous playback of video.

Peers employ a sliding window mechanism to manage data chunks, which periodically progresses. For smooth playback of video all chunks should be present in sliding window. Within this framework, peers retain video chunks they have already received, and monitors both the availability and absence of video chunks in their windows. This approach enables peers to identify missing data chunks which are essential to receive before their playing deadline and determines the chunk schedule for smooth playback of video.

Let W represent the set of data chunks within the current sliding window for peer p_i, and M - a subset of W, includes the data chunks missing from W that peer p_i can seek from its neighbors. Peer p_i aims to request these missing chunks in a manner that optimally enhances its utility from them. Peer p_i determines the priority of each missing chunks from sliding window.

Thus after prioritizing the missing chunks, peers decides the order for which chunks to request to ensure uninterrupted video playback.

Urgency and scarcity of chunks are important factors to determine the chunk schedule. Urgency shows the playing position of chunk with respect to current playback point, thus the chunk closer to playback point are considered as urgent chunks. While scarcity denotes the number of peers caching a chunk. To determine the chunk request schedule the priority of chunk is determined by using equation (4). Then the chunks with higher priority are scheduled first. Thus priority of chunk *k* where k ≤ M in peer *pi*

$$CP_{pi,k} = \frac{1}{tr_{pi,k} - tp_{pi,k}} + \alpha * \left(\frac{1}{N_b\left(C_{pi,k}\right)}\right) + ran(0,1) * \delta \qquad (4)$$

Where,

$CP_{pi,k}$ - Priority of chunk *k* in peer *pi.*

$tr_{pi,k}$- Location of chunk k of peer *pi* during playback along the time axis.

$tp_{pi,k}$ - The position that peer *pi* is currently playing along the time axis.

$N_b\left(C_{pi,k}\right)$ - Total neighbors of the peer p*i* which hold the chunk k

α - The local rarest first strategy's coefficient helps to precisely balance the importance of giving urgent and rare chunks equal weight.

δ - When two chunks have the same rarest priority, the shaking coefficient is utilized to change the trembling range.

The priority of missing chunks from sliding window are calculated and stored in descending order and scheduled sequentially.

Once the priority of chunks is calculated, the next step is parent peer selection for sending chunk request as there are multiple peers with selected chunk. For this step receiving peer first evaluates the Buffer Maps (BMs) received from neighbors and finds neighbor list with available chunks and each peer is evaluated for optimal selection of parent.

In this system, the best parent peer for request is preferred with the help of crow search optimization algorithm. We used a fitness function considering the service capability of the peer.

The uplink bandwidth, online time of the peer and distance between peers are the features considered for calculation of service capability. Minimizing delay is crucial for enhancing on-demand system performance. Directly impacting transmission delay, returning the nearest neighbours in the network topology to the downloading peer is essential. Given the autonomous nature of peers in a P2P network, with the freedom to join or leave at any time, a strategy to lower the likelihood of connecting with unstable peer involves prioritizing those with longer online durations. Along with this if peer with sufficient upload bandwidth should be considered which increase the probability of connecting peers which will serve the request. So considering all these important factors we used for calculation of service capability of each neighbour. Thus, we define the fitness function based on capability of node for serving the request of the neighbour peer *j* of peer *i* and is described by following equation (5).

$$Capa_{pi}^{pj} = \frac{\left\lfloor T_{ipi}^{pj} * \left(UBw_{pi}^{pj}\right)^2 \right\rfloor}{D_{pi}^{pj}} \qquad (5)$$

Where,

$Capa_{pi}^{pj}$ - Service capacity of the neighbour *pj* of peer p*i*,

T_{pi}^{pj} - The duration for which neighbour *pj* of peer pi has been online in the system

UBw_{pi}^{pj} - Upload bandwidth of the neighbour *pj* of peer *pi*

D_{pi}^{pj} - Distance between peer *pi* and peer *pj* in number of hops.

Initially, the service capability of each neighbouring peers are identified considering parameters as age, upload bandwidth and distance. Thus objective function is defined by equation (5) and goal is to maximize the fitness function as shown in equation (6). Crow search algorithm is used for finding optimal peer from all available neighbours with needed chunk such that peer with maximal fitness will be selected and improve the QoS. Thus the chunk request will be sent to the optimal parent peer which is derived by CSA.

$$Fitness = \max\left(\sum_{N_b(C_{pij,k})} Capa_{pi}^{pj}\right) \qquad (6)$$

s.t:

$$\sum_{j=1}^{N_b(C_{pi,k})} pj \leq 1,\ \forall\, k \in M \qquad (a)$$

$$\sum_{k\in M} sz(k)\ \leq UBw(pj)\quad t, \forall\, pj \in\ N_b(pi) \qquad (b)$$

Where,

Fitness− It is the optimal peer having maximum service capability among all the neighbor $N_b(C_{ij})$ of peer p*i*.

$sz(k)$ − Size of missing chunk which is to request.

$UBw(pj)$ − Upload bandwidth of peer sender *pj* at request time t

$N_b(pi)$ – Neighbors of peer *pi*.

τ is the request period.

Constraint (a) - ensures that each chunk should not be scheduled with only one sender.

Constraint (b) - ensures that chunks should be scheduled only when sufficient upload bandwidth is available with sender.

3.3. Steps for Crow Search Based Data Scheduling Approach

The efficient parent selection we propose operates through the following steps:

1. Initially, the hybrid CDN-P2P network is designed with source server, edge servers and nodes will be added to each edge server as time evolves.
2. When new node enters into the system it gets connected with respective edge server and other peers who are watching same video based on its playback point and forms mesh overlay.
3. Each peer in the overlay caches the video chunks that it receives from other peers and serve to other peers. Peers across the network exchange buffer maps and finds missing chunks from buffer using equation (3).
4. Then missing chunks gets prioritized using the equation (4) and high priority chunks are selected first for request phase.
5. Peer checks the availability of needed chunk with neighbours using their buffer maps. If peer finds the neighbours from neighbour list with selected chunk then go to step 6 else go to step 7
6. Crow search algorithm finds the optimal source peer as parent for chunk request using the steps as shown in fig (2) where all neighbouring peer forms flock and each peer acts as crow. Initial memory for each peer is set based on decision variable values is initialized. Each peer is evaluated by calculating fitness for each neighbouring peers using equation (5) and gives the optimal peer as output which maximize the equation (6). Then peer requests the chunk from selected optimal peer. Parent peer sends chunks to requested peer.
7. Peer request chunk from the edge server directly.

4. Performance Evaluation

In this part, we assess the efficiency of the data scheduling strategy. Initially, we present the simulation environment used for evaluation. Following that, we compare the performance of the proposed data scheduling algorithm against traditional methods such as LLP and Random, to gauge its effectiveness.

4.1. Simulation Setup

OMNET++ [38] and its extension modules INET [39] and OverSim [40] make up the simulation platform. OMNET is an open-architecture, component-based discrete temporal event simulator. INET is a framework based on OMNeT++ that includes many application models, IPv4, IPv6, TCP, UDP, and many more protocol implementations. Sitting atop the INET framework is OverSim, an open-source overlay network simulation framework. Using Georgia Tech Internet Topology Model (GT-ITM) physical topology is generated with 28 AS (Autonomous System) which forms the backbone router and each AS with 28 access routers operating in top-down mode [41]. For simulation, we developed a hybrid CDN-P2P VoD system having four edge servers. The system uses tracker associated with edge server to arrange nodes into mesh structure. New joining peer assigned to the respective mesh overlay based on playback point. Peers follow a Poisson distribution for their arrival process. Peers enters into the system with 2 seconds of a mean inter-arrival time. Each peer independently chooses its upload bandwidth between 350 Kbps to 2 Mbps with uniform distribution and its download bandwidth from 1 to 4 Mbps. Server upload and download bandwidth is set to 10 Mbps.The MPEG-4 stream content under consideration is I, P and B frames. A video is composed of chunks and chunk size in our system is considered of one frame. This is a receiver-based scheduling where peer initiates the process by requesting chunks from its neighboring peers. During the startup buffering phase, the chunk request rate is set to as high as three times the average video bitrate to minimize startup delay. While during normal playback, the rate of chunk request is controlled to maximum double of average video bitrate. Due to P2P networks' inherent randomness, all simulations for the MPEG-4 video coded Verbose_ARD Talk sample video trace file are repeated four times. Table 1 shows the other parameters used in simulation.

Table 1. Simulation Parameters

Parameters	Value
Video Codec	MPEG-4
Total frames in GOP	12
Video Frame Rate	25 frames/sec
Start-up Buffer Size	12 sec
Average Video Bit Rate	512 Kbps
Number of Peers	50, 100, 150, 200
Simulation Cycle	600 sec
Buffer Map Exchange period	1 sec.

It is important to set the controlling parameters- flight length and awareness possibility for CSA which has effect on the optimal solution. The performance of the CSA is notably influenced by the flight length, where a shorter length tends to result in local optimum solutions, while a longer length aids in achieving global optimum solutions. Observing the responses of the CSA at various awareness probability (AP) values, higher AP values lead to the algorithm struggling to capture the global optimum solution and instead converging towards local optima. We tested various flight length values - 2, 3 & 4. Also approximation probability (AP) values ranging from 0.1 to 0.3. After various trial the system gave the optimal solution for *fl=2* and *AP=0.3* with 80 iterations maximum.

4.2. Evaluation metrics

We evaluate the proposed scheduling with other different scheduling strategies using following performance metrics:

- Start-up Delay: The interval between joining to the network and commencing video playback is termed as startup delay.
- Continuity Index (CI): It is the ratio of chunks that reach before their playback deadline to the total number of chunks during video playback. It conveys how smoothly the video is received. An improved viewing experience is provided for the audience with a higher continuity index score. It is expressed using equation (7).

$$C = \frac{(\text{Total no. of chunks} - \text{Total missing chunks})}{(\text{Total no. of chunks})} \qquad (7)$$

- Server Load: This measure illustrates how much upload bandwidth the server needs to process requests from every peer that is linked to it.

4.3. Results and discussions

We use the following peer parent selection policies for comparison with our proposed system.

- Random peer selection: In this approach peer sends the chunk request to a randomly selected neighbor who holds the necessary chunk.
- Least Loaded Peer (LLP) selection: The basic idea of this approach is that all the peers required to periodically report the size of their request queues to their neighbors. Then, the chunk request is sent to the smallest queue size peer means less loaded peer who have the needed chunk.

The selection of the random and LLP strategies is based on two main considerations. Firstly, the random strategy is widely recognized and has been implemented in numerous real-world P2P networks. Secondly, the LLP approach is designed to factor in the workload of individual peers.

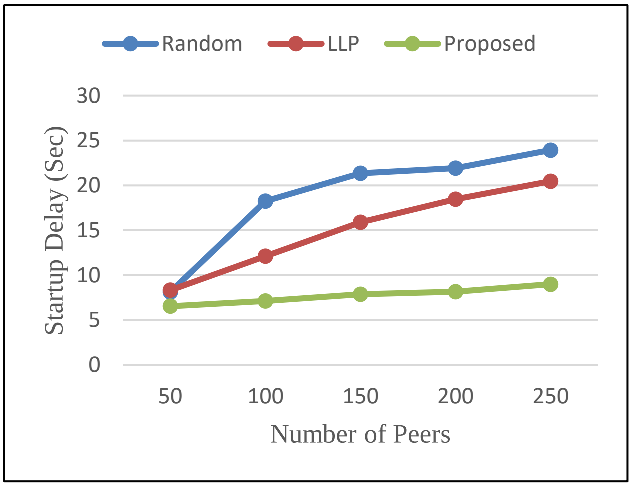

Fig 3. Average Start-up Delay

Figure 3 illustrates the initial startup delay experienced by all algorithms across different peer counts. Initially, all methods perform similarly, leading to a minimal variation in startup delays among the various approaches. As the count of peer nodes increases, it is common to see an increase in delay. Consequently, the startup delay associated with the three evaluated policies intensifies with the expansion of peer numbers. However, our proposed system demonstrates a more modest growth in delay, especially when contrasted with the increments observed in both the random and LLP methodologies.The reason for the low start-up delay in our proposed system is that for each peer neighboring nodes will be allocated which are closer to playback point of peer and selects parent which has sufficient upload bandwidth, close to requesting peer and is stable in the system. Hence, chunks will be delivered faster and start-up delay is reduced. Thus when the peer count in the network increases from 100 to 250, the start-up delay increases rapidly in case of random and LLP method but not in our proposed method.

As illustrated in Fig. 4, the media playing continuity index improves as the network scale grows because a greater playing continuity and a faster download rate are associated with more peers in the network. When there are 250 peers, our CSA-based approach's continuity index approaches 0.98 due to the fastest optimal parent peer found in a short amount of time, which takes into account the node's stability, distance, and available upload bandwidth. As a result, the majority of chunk requests result in successful chunk arrivals. However, we can see that the viewing quality which is indicated by CI under CSA based approach is much better than that of Random and LLP method. In particular, the mean LLP CI is approximately 0.88, whereas the value is approximately 0.69 for Random. This is because the upload bandwidth limit and the node's age, which indicates stability, are not taken into account when choosing a parent. As a result, data may not be sent to the requesting peer in a timely manner before the deadline for video playback, which could cause users to experience buffer waiting or even lose data.

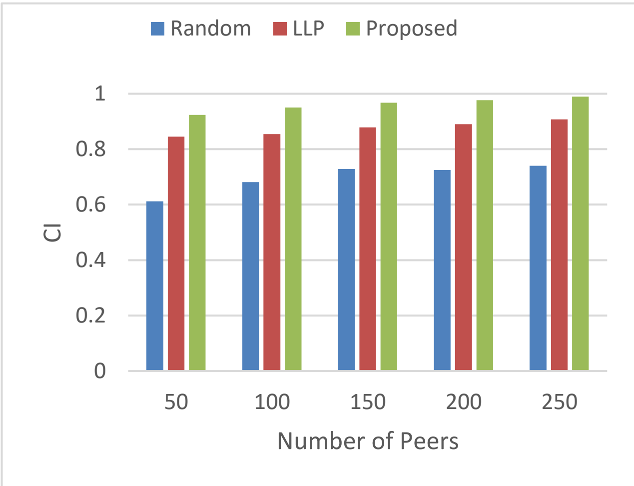

Fig 4. Continuity Index for Different Number of Peers

The comparison of server load is shown in Fig. 5. In the first phase, we see an overlap of the three curves, which suggests that when few peers join the network, the server bandwidth consumption of various algorithms is roughly equal. All peers often retrieve pieces from the server because there are very few peers and thus few chunks are available in the neighbourhood. As the network expands and peers start to serve other peers, the server load eventually drops. Our CSO-based method performs significantly better than the other two methods. Approximately 69% and 58% of server upload capacity are conserved as compared to the Random and LLP techniques, respectively. The substantial reduction in server load is primarily attributed to the peer selection strategy by using crow search algorithm, which prioritizes peers with close proximity, sufficient upload bandwidth and stable peer as well as similar content availability within our system. Because LLP uses a better load balancing strategy than the random based request peer selection process, it can also minimize server load. It is not as efficient as our CSO-based solution, though. Peers in LLP are perhaps more dependent on the server because LLP does not differentiate between the server and regular peers.

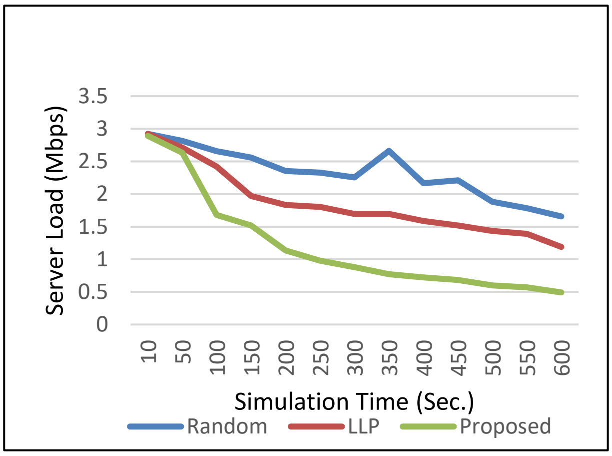

Fig 5. Server Load with Simulation Time

5. Conclusion

A novel scheduling technique based on the crow search algorithm (CSA) is proposed in this paper. Our proposal improve the quality of service and efficiency of a hybrid CDN P2P based video-on-demand system. Simulations using OMNET++ with the INET and Oversim frameworks are used to assess performance. First, the network has been developed and peers are arranged in mesh overlay. Each peer caches chunks in its buffer before playing and maintains it to serve other peers. These chunks are prioritized using the priority function and optimal peers which cache the requested chunks has been selected. This new parent selection technique chooses peers with the least distance and whose up time is longer, in addition to utilizing the upload bandwidth of peer. Peer service capability is assessed using the fitness function, and the CSA algorithm finds the best peers based on peer service capability. According to simulation results, it showed that our strategy performs better than conventional methods in terms of startup latency, playback continuity, and also reduce server load.

In future we will focus on overlay management strategy to make the system more efficient and improve the user experience.

References:

[1] K. Feng and Y. Xudong, "A Study on Grid-based VOD System in the E-Learning," *Inf. Technol. Appl. Int. Forum On*, vol. 1, pp. 407–410, May 2009, doi: 10.1109/IFITA.2009.52.

[2] S. N. Dhage, S. K. Patil, and B. B. Meshram, "Survey on: Interactive Video-on-Demand (VoD) systems," in *2014 International Conference on Circuits, Systems, Communication and Information Technology Applications (CSCITA)*, Apr. 2014, pp. 435–440. doi: 10.1109/CSCITA.2014.6839300.

[3] Y. Yang, A. L. H. Chow, L. Golubchik, and D. Bragg, "Improving QoS in BitTorrent-like VoD Systems," in *2010 Proceedings IEEE INFOCOM*, Mar. 2010, pp. 1–9. doi: 10.1109/INFCOM.2010.5462029.

[4] M. Haddad *et al.*, "A survey on YouTube streaming service," in *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, in VALUETOOLS '11. Brussels, BEL: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), May 2011, pp. 300–305.

[5] N. Ramzan, H. Park, and E. Izquierdo, "Video streaming over P2P networks: Challenges and opportunities," *Signal Process. Image Commun.*, vol. 27, no. 5, pp. 401–411, May 2012, doi: 10.1016/j.image.2012.02.004.

[6] H. He, Y. Feng, Z. Li, Z. Zhu, W. Zhang, and A. Cheng, "Dynamic Load Balancing Technology for Cloud-oriented CDN," *Comput. Sci. Inf. Syst.*, vol. 12, pp. 765–786, Jul. 2015, doi: 10.2298/CSIS141104025H.

[7] A. Passarella, "A survey on content-centric technologies for the current Internet: CDN and P2P solutions," *Comput. Commun.*, vol. 35, no. 1, pp. 1–32, Jan. 2012, doi: 10.1016/j.comcom.2011.10.005.

[8] G. Zhang, W. Liu, X. Hei, and W. Cheng, "Unreeling Xunlei Kankan: Understanding Hybrid CDN-P2P Video-on-Demand Streaming," *IEEE Trans. Multimed.*, vol. 17, no. 2, pp. 229–242, Feb. 2015, doi: 10.1109/TMM.2014.2383617.

[9] Y. Ding, Z. Wu, and L. Xie, "Enabling Manageable and Secure Hybrid P2P-CDN Video-on-Demand Streaming Services Through Coordinating Blockchain and Zero Knowledge," *IEEE Multimed.*, vol. 30, no. 01, pp. 36–51, Jan. 2023, doi: 10.1109/MMUL.2022.3191680.

[10] Z. Ma, K. Xu, and Y. Zhong, "Exploring the policy selection of P2P VoD system — A simulation based research," in *2012 IEEE 20th International Workshop on Quality of Service*, Jun. 2012, pp. 1–4. doi: 10.1109/IWQoS.2012.6245991.

[11] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," *Proc. IEEE*, vol. 96, no. 1, pp. 11–24, Jan. 2008, doi: 10.1109/JPROC.2007.909921.

[12] N. Magharei and R. Rejaie, "Understanding mesh-based peer-to-peer streaming," in *Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video*, in NOSSDAV '06. New York, NY, USA: Association for Computing Machinery, May 2006, pp. 1–6. doi: 10.1145/1378191.1378204.

[13] X. Hei, Y. Liu, and K. W. Ross, "IPTV over P2P streaming networks: the mesh-pull approach," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 86–92, Feb. 2008, doi: 10.1109/MCOM.2008.4473088.

[14] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *IEEE Trans. Multimed.*, vol. 9, no. 8, pp. 1672–1687, Dec. 2007, doi: 10.1109/TMM.2007.907451.

[15] X. Wei, P. Ding, L. Zhou, and Y. Qian, "QoE Oriented Chunk Scheduling in P2P-VoD Streaming System," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8012–8025, Aug. 2019, doi: 10.1109/TVT.2019.2922273.

[16] Y. Cui, B. Li, and K. Nahrstedt, "oStream: asynchronous streaming multicast in application-layer overlay networks," *IEEE J. Sel.*

Areas Commun., vol. 22, no. 1, pp. 91–106, Jan. 2004, doi: 10.1109/JSAC.2003.818799.

[17] T. T. Do, K. A. Hua, and M. A. Tantaoui, "P2VoD: providing fault tolerant video-on-demand streaming in peer-to-peer environment," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, Jun. 2004, pp. 1467-1472 Vol.3. doi: 10.1109/ICC.2004.1312755.

[18] M. Fouda, T. Taleb, M. Guizani, Y. Nemoto, and N. Kato, "On Supporting P2P-Based VoD Services over Mesh Overlay Networks," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov. 2009, pp. 1–6. doi: 10.1109/GLOCOM.2009.5425840.

[19] M. Hanada and H. Kanemitsu, "P2P Streaming Method Based on Playback Deadline Using Linear Programming," *J. Signal Process.*, vol. 22, pp. 47–62, Mar. 2018, doi: 10.2299/jsp.22.47.

[20] C. Feng, B. Li, and B. Li, "Understanding the Performance Gap Between Pull-Based Mesh Streaming Protocols and Fundamental Limits," in *IEEE INFOCOM 2009*, Apr. 2009, pp. 891–899. doi: 10.1109/INFCOM.2009.5061999.

[21] C. Liang, Z. Fu, Y. Liu, and C. W. Wu, "Incentivized Peer-Assisted Streaming for On-Demand Services," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 9, pp. 1354–1367, Sep. 2010, doi: 10.1109/TPDS.2009.167.

[22] Z. Wen, N. Liu, K. L. Yeung, and Z. Lei, "Closest Playback-Point First: A New Peer Selection Algorithm for P2P VoD Systems," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, Dec. 2011, pp. 1–5. doi: 10.1109/GLOCOM.2011.6134116.

[23] P. Liu, G. Huang, S. Feng, and J. Fan, "Event-Driven High-Priority First Data Scheduling Scheme for P2P VoD Streaming," *Comput. J.*, vol. 56, no. 2, pp. 239–257, Feb. 2013, doi: 10.1093/comjnl/bxs127.

[24] S. Yang, Y. Shen, W. Qu, and K. Li, "A Novel On-Demand Streaming Service Based on Improved BitTorrent," in *2010 Fifth International Conference on Frontier of Computer Science and Technology*, Aug. 2010, pp. 46–50. doi: 10.1109/FCST.2010.111.

[25] X. Wei, P. Ding, F. Zho, J. Lou, and Y. Gao, "A Load Balancing Strategy based on Request Queue for P2P-VoD System," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Jun. 2019, pp. 668–673. doi: 10.1109/IWCMC.2019.8766536.

[26] T. Rohmer, A. Nakib, and A. Nafaa, "Optimal Peer Selection Strategy in P2P-VoD Systems Using Dynamic Evolution Strategy," in *2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum*, May 2013, pp. 474–481. doi: 10.1109/IPDPSW.2013.92.

[27] C. Rodrigues, "Analyzing Peer Selection Policies for BitTorrent Multimedia On-Demand Streaming Systems in Internet," *Int. J. Comput. Netw. Commun.*, vol. 6, pp. 203–221, Jan. 2014, doi: 10.5121/ijcnc.2014.6114.

[28] G. Huang, P. Liu, and X. Gong, "A novel peer selection strategy in P2P VoD system using biased gossip," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, Jun. 2015, pp. 372–377. doi: 10.1109/ICCSN.2015.7296186.

[29] J. Zhou, Z. Chen, and G. Feng, "Game theoretical bandwidth request allocation strategy in P2P streaming systems," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2013, pp. 1657–1662. doi: 10.1109/GLOCOM.2013.6831311.

[30] Q. Yu, B. Ye, S. Lu, and D. Chen, "Optimal data scheduling for P2P video-on-demand streaming systems," *IET Commun.*, vol. 6, no. 12, pp. 1625–1631, Aug. 2012, doi: 10.1049/iet-com.2011.0190.

[31] M. Alshayeji, D. Dias, and M. D. Samrajesh, "Enhanced Chunk Regulation Algorithm for Superior QoS in Heterogeneous P2P Video on Demand," *J. Netw.*, vol. 10, Jan. 2016, doi: 10.4304/jnw.10.10.567-578.

[32] T. Rohmer, A. Nakib, and A. Nafaa, "Dynamic Strategy Selection based on Evidence Theory. Application on a P2P-VoD system," *IFAC-Pap.*, vol. 49, no. 12, pp. 775–780, Jan. 2016, doi: 10.1016/j.ifacol.2016.07.868.

[33] G. Sivakumar and V. Venkatachalam, "Artificial Bee Colony Based Data Scheduling for Peer to Peer Network Video on Demand System," *J. Comput. Theor. Nanosci.*, vol. 14, pp. 5907–5914, Dec. 2017, doi: 10.1166/jctn.2017.7034.

[34] K. AjithaGladis, "Gravitational search algorithm based data scheduling for peer to peer video on demand system," *Multimed. Tools Appl.*, vol. 78, Oct. 2019, doi: 10.1007/s11042-019-7644-y.

[35] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. 2010.

[36] A. G. Hussien *et al.*, "Crow Search Algorithm: Theory, Recent Advances, and Applications," *IEEE Access*, vol. 8, pp. 173548–173565, 2020, doi: 10.1109/ACCESS.2020.3024108.

[37] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016, doi: 10.1016/j.compstruc.2016.03.001.

[38] https://omnetpp.org/

[39] https://inet.omnetpp.org/

[40] http://www.oversim.org/.

[41] https://omnetpp.org/download-items/GTITM-OverSim-Mlab.html