Received June 3, 2021, accepted June 18, 2021, date of publication June 24, 2021, date of current version July 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3092277

Share-Ratio-Based Incentive Mechanism for File Sharing With BitTorrent Protocol

AMINU ADAMU
School of Information and Communications Technology, University of the Gambia, Serrekunda, Gambia
Department of Computer Science, Umaru Musa Yar'adua University, Katsina 2218, Nigeria

Corresponding author: Aminu Adamu (amadamum@gmail.com)

This work did not involve human subjects or animals in its research.

ABSTRACT Several P2P file-sharing networks were successfully deployed and are used by millions of Internet users globally to share files (such as movies, video clips, software systems, e-books, etc.) cooperatively over the Internet. These networks were designed with cooperation in mind, therefore, several incentive mechanisms were devised to encourage cooperation among downloaders and yet the free-riding phenomenon remains a threat to these networks since free-riders could indulge in acts such as whitewashing, Sybil, and collusion attacks to escape penalties imposed by the incentive mechanisms. This paper proposes a share-ratio-based incentive mechanism for P2P file-sharing networks where files are shared via BitTorrent protocol. The proposed incentive mechanism is not prone to Sybil and collusion attacks, and it was designed to simultaneously: (1) encourage cooperation, (2) provide fairness to new downloaders, and (3) deter free-riding while resisting whitewashing attack. The proposed incentive mechanism does not require any central entity to be realized and it was designed in existing terminologies used in P2P file-sharing networks, hence, it is easy to implement. In addition to the accomplishment of the set objectives, the proposed incentive mechanism has successfully deterred free-riding as shown by the experiments conducted.

INDEX TERMS File-sharing networks, free-riding, incentive mechanism, P2P, share ratio.

I. INTRODUCTION

File-sharing applications such as BitTorrent [1], qBittorrent [2], BitComet [3], BitLord [4], μTorrent [5], etc. are used by millions of users globally to download files over the Internet. In these systems, files are shared among users cooperatively using the Peer-to-Peer (P2P) model. Recently, P2P model has been proposed for the design of distributed and scalable social-aware network systems [6]–[8]. One of the key functional ideas of P2P systems is the motivation of users to act as clients and servers simultaneously, hence, users are called peers. In P2P systems, peers not only download data but also upload data to other peers; in this way, peers' upload bandwidth is effectively utilized to reduce the data distribution burden otherwise placed on the server, i.e. file source [9], [10]. Peers downloading and uploading a file using a P2P file-sharing protocol are said to be in a *torrent-session*.

There are two kinds of peers in P2P file-sharing networks, namely, *leeches* and *seeds*. Leeches are peers that are downloading the file (i.e. they do not possess the complete file) and they perform two actions simultaneously, i.e. downloading the file's data for their consumption and uploading the so far downloaded file's data to others; however, the term *downloader* will be adopted in the paper instead of *leech*. Seeds are peers that have completely downloaded the file and decided to stay in the network to altruistically upload the file's data to the remaining downloaders in the torrent-session, seeds perform only one action, i.e. uploading the file's data to downloaders. Unless there is a need for specification, further in the text, the term *peer* will be used as a general term to refer to both a downloader and a seed.

Since P2P networks were designed with cooperation in mind, one of their key challenges is *free-riding* [11], [12]. In P2P networks, some downloaders regulate their upload capacity to zero or to a very low level such that they can only download data from the network with little or no data upload, such downloaders are referred to as *free-riders*. The free-riding act worsens the network's performance by prolonging the average file-download time and may even lead to the system's collapse. To address the free-riding phenomenon, several incentive mechanisms were proposed which are either economy-based, reciprocity-based or their

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed M. A. Moustafa.

hybrid. In economy-based incentives, virtual currencies or credit points are used for the payment of each service received by a peer, subsequently, peers lose currency/points for each service they received (i.e. for each data downloaded) and earn for each service they have provided (i.e. for each data uploaded). In reciprocity-based incentives, peers receive services based on their contributions to the system.

To escape the punitive measures introduced by some of the existing incentive mechanisms, misbehaving peers indulge in acts such as whitewashing [11], Sybil [13], and collusion attacks [14]. Whitewashers are peers that leave and rejoin the network with new identities to continue misbehaving, and in Sybil attack, a misbehaving peer joins the network with different pseudonymous identities to maximize its benefits, while in collusion attack, misbehaving peers collude with each other and act maliciously to escape penalties imposed by the incentive mechanism or to maximize their reputations (more prominent in reputation related incentives). Additionally, some of the existing incentive mechanisms are unfair to newcomers, some produce heavy overheads in a single transaction, and some rely on central entity for realization.

Generally, P2P file-sharing communities are either public or private [15]–[17]. Public communities are open, i.e. peers are free to join without registration, while private communities are closed, i.e. a peer must register to join the community. In private communities, private trackers are used to track each peer's cumulative uploads and downloads *across all* the torrent-sessions it has so far participated in. The ratio of uploads to downloads is termed as *share-ratio*. In private communities, a threshold share-ratio is fixed, such that if a peer's cumulative share-ratio falls below the set threshold value, its account will be deleted. It is competitive to have and to maintain an account in a private community, because the number of accounts they can hold is limited, as such, peers strive to maintain their accounts by uploading higher than they download. Seeding after file-download completed is encouraged in private communities by given certain privileges to peers with high share-ratios, such as the ability to access and easily download the latest files, enhanced searching capabilities in the private sites, allocation of invitations which could be sold or given for free to friends willing to join the community, etc. [15]. Such policy used in private communities leads to the high seed-to-downloader ratio in torrent-sessions, therefore, newcomers have to seed for a long time to have their share-ratios above the threshold value, this will, in turn, increase the number of seeds in a session, consequently, users may lack the motivation to initiate a new download due to the possibility of long seeding [17], [18]. The share-ratio policy enforced in private communities is not directly involved during data exchange among peers.

Unlike the share-ratio policy enforced in private communities, this paper proposes a share-ratio-based incentive scheme that is purely synchronous within the same torrent-session, i.e. peers use their share-ratios generated only in a given torrent-session to exchange data with their neighbors only in that session. The proposed share-ratio-based incentive mechanism is meant for public communities since they are more susceptible to free-riding due to their open nature [19]–[21], however, it can still be used in private communities for data exchange among peers to complement the already existing share-ratio policy. The proposed mechanism is lightweight and decentralized, therefore, if used, the scalability and fault-tolerance of the P2P file-sharing system will not be compromised.

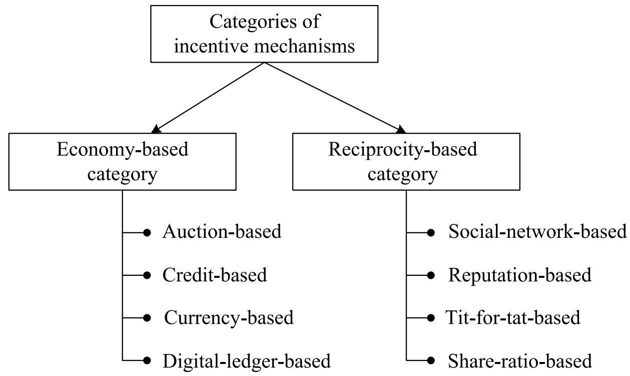

FIGURE 1. Taxonomy of incentive mechanisms.

The incentive mechanism proposed in this paper aimed at:

- preventing free-riding while resisting whitewashing attack;
- encouraging cooperation by providing download priorities to the best cooperating downloaders; and
- providing fairness to the newcomers.

The rest of this paper is organized as follows: Section II provides the related works and section III presents a brief overview of a file sharing process with BitTorrent protocol. In section IV, the share-ratio-based incentive mechanism is presented and section V concludes the paper.

II. RELATED WORKS

Several incentive schemes were proposed for P2P data sharing networks. Figure 1 shows the taxonomy of the incentive mechanisms. The existing incentive mechanisms can be categorized into two categories, i.e. economy-based category and reciprocity-based category.

A. ECONOMY-BASED CATEGORY

With incentive mechanisms in this category, peers pay via virtual currency or credit points for each service they have received and earn for each service they have provided. Examples of incentive mechanisms in this category are auction-, credit-, currency-, and digital-ledger-based incentives.

In **auction-based incentive mechanisms**, exchanges of data are done based on auctions. In these incentive schemes, peers bid for their desired data from their neighbors, and data are delivered after payments. Each peer maintains a budget and functions as a bidder and a seller simultaneously. Peers aim at maximizing their revenues to gain more purchasing power, as such, they maximally utilize their upload bandwidths to accumulate more revenue over time. Liu *et al.* [22] proposed an auction-based incentive mechanism for P2P systems, where peers bid for their required resources from their neighbors, and resources are delivered to the winning bidders after the payments. A similar auction-based incentive mechanism was proposed by Wu *et al.* [23] for P2P VoD streaming systems. In the work, the data exchange process among peers was modeled in form of dynamic and iterative auctions. Since the streaming quality depends on the availability of rarest blocks as well as the blocks with the closest playback deadline, therefore, in the scheme proposed in [23], peers that own these kinds of blocks set high prices to them, thus, peers' target is to obtain these blocks to sell at high prices so as to boost their purchasing powers, subsequently, this act maximizes the utilization of upload bandwidth as well as the circulation of these valuable blocks in the system. However, when newcomers are subsidized, these incentive mechanisms become vulnerable to Sybil and whitewashing attacks. Additionally, in these schemes, no proper and secured currency system is set for transactions.

In **credit-based incentive mechanisms**, peers are incentivized by awarding credit points for each service they have provided; and for each service they received, their points are deducted. In the credit-based schemes such as the ones proposed in [24]–[26], to provide fairness to newcomers, each new arriving downloader is allocated a fixed number of points which may either decrease with each service received or increase with each service provided by the peer, such policy made credit-based incentives prone to whitewashing and Sybil attacks. In the case of a whitewashing attack, a misbehaving peer could leave the network before it exhausts the initially awarded points and rejoins to get more points, whereas, in the case of Sybil attack, a misbehaving peer could join the network with several fake identities to accumulate multiple initial points which will give it an undue advantage over other peers.

In **currency-based incentives mechanisms**, peers pay for the services they have received and charge for the services they have provided. In such incentive mechanisms, virtual currency is used for payments after each transaction and a single authority is employed to oversee the transactions. Wang *et al.* [27], [28], Dong *et al.* [29], and Mehr and Fooladi [30] proposed currency-based incentive mechanisms where peers maintain a virtual currency account and their accounts are debited for each service they have received, likewise, their accounts are credited with the virtual currency for each service they have provided. In these schemes newcomers are subsidized for their initial transactions, however, this policy will attract whitewashing and Sybil attacks as in the case of credit-based incentives. Additionally, reliance on a central authority to oversee the transactions in currency-based incentive schemes as well as lack of proper and secured currency system could compromise the system's efficiency.

In **digital-ledger-based incentive mechanisms** such as the ones proposed in [31] and [32], a cryptocurrency is used as a mode of payment for data transactions. Peers earn money for providing services and lose money for receiving services. Because cryptocurrency has some direct financial value, therefore, due to downloading costs, free-riders will be reluctant to request and download data from a system with such an incentive mechanism. However, these incentive approaches will require a lot of computational power and will cause bandwidth overhead. In such systems, to avoid misbehaving users, additional service is provided only after the payment of the previous service is acknowledged, hence, the use of this approach to incentivize peers will prolong the data delivery latency and subsequently the overall file-download time. Additionally, the complexity of these schemes makes them difficult to implement.

B. RECIPROCITY-BASED CATEGORY

With incentive mechanisms in this category, peers receive services proportional to their contributions to the system, examples of incentive mechanisms in this category are social-network-, reputation-, tit-for-tat-, and share-ratio-based incentives.

In **social-network-based incentive mechanisms** such as [7], [12], [33], peers maintain social relationship links and provide services to their social friends based on the weight of their friendship and their mutual sharing contributions. Social network-based incentives are unfair to newcomers and are not scalable.

In **reputation-based incentive mechanisms**, peers are recognized as either good or bad based on their reputations, good peers are rewarded and bad peers are punished. In [34], a semi-distributed and application-independent reputation system was proposed for P2P networks. In such a system, QoS-related information is stored across system users and clients predict the reliability of servers based on self-experiences and feedbacks from other users. GossipTrust proposed in [35] is a reputation-based incentive scheme. GossipTrust system applies a gossip-based protocol for aggregating global reputation scores for all peers in the system. GossipTrust system leverages Bloom filter architecture for efficient score ranking. Another reputation-based incentive was proposed in [36] where a user uses a watchdog to monitor the behaviors of its neighbors, if it observed a non-cooperative neighbor, it will broadcast the information of that neighbor to all users in the network. Pouryazdan *et al.* [37] proposed a centralized reputation-based evaluation using collaborative reputation scores. In reputation-based incentives, misbehaving users could collude with each other to maximize their reputations. Reputation-based incentives apart from being unfair to newcomers; are also prone to whitewashing and Sybil attacks.

In **tit-for-tat-based incentive mechanisms**, the level of service received by peers depends on their reciprocal cooperation. Free-riding in BitTorrent is addressed via tit-for-tat and choke policy [38]. In [39], a tit-for-tat incentive scheme was proposed to discourage free-riding in P2P layered video streaming applications using the T-Chain incentive mechanism. With tit-for-tat incentive mechanisms, users periodically search for neighbors with better reciprocation by randomly sending data to other users that are not currently their partners, as such, non-cooperating peers will still receive data periodically and a misbehaving user could take the advantage of the optimistic unchoking to create multiple fake identities to receive multiple services periodically [40].

In **share-ratio-based incentive mechanisms**, peers receive services based on their share-ratios, and peers whose share-ratios fall below a defined threshold value are punished. Share-ratio policy is prominently used in private P2P communities where global share-ratios of peers are saved on private trackers and a decision on whether to allow or remove a peer from the private community is done based on its current global share-ratio, if peer's share-ratio falls below a defined threshold value it will be evicted from the community, since being a member within a private community is a privilege, therefore, to avoid eviction, peers maximally utilize their upload resources to maintain their share-ratios above the threshold [15]–[17]. In [41], Nishida and Nguyen proposed a global contribution approach to maintain fairness in P2P networks. In the work, the peer's relative contribution is computed after each transaction made, the value obtained is termed as peer's Global Contribution (GC). With the GC approach, the peer's GC value is recomputed after each transaction, and once the peer's value has changed from its previous state after the transaction, it must broadcast the value to all peers participating in the file sharing, hence, the approach is not scalable. During each transaction cycle, peers aimed at requesting and downloading data from peers with low GC values, however, this will cause excessive wasted requests since peers with low GC value could be misbehaving peers whose upload bandwidths were intentionally set low to free-ride. Additionally, with this approach, at the beginning of each transaction, peers have to estimate to see whether their GC value will increase or decrease after the transaction, then they can decide on whether to partake in the transaction or to refuse, such time taken to do the estimations and to make the decisions will prolong the file-download time.

Unlike the share-ratio policy used in private communities which is not directly used during data exchange among peers, the share-ratio-based incentive mechanism proposed in this paper is for data trading among peers in a single torrent-session. The incentive mechanism proposed in this paper prevents free-riding while resisting whitewashing attacks. The proposed mechanism is lightweight, decentralized, and therefore, scalable.

III. BRIEF OVERVIEW OF FILE-SHARING PROCESS WITH BITTORRENT PROTOCOL

The proposed incentive mechanism in this paper is aimed at preventing free-riding in P2P systems that use BitTorrent protocol for file-sharing. In such systems, a file to be shared is fragmented into *pieces* of equal size (e.g. 256KB each) and each piece is further fragmented into sub-pieces known as *blocks* [38], [42]. Peers sharing (downloading and uploading)

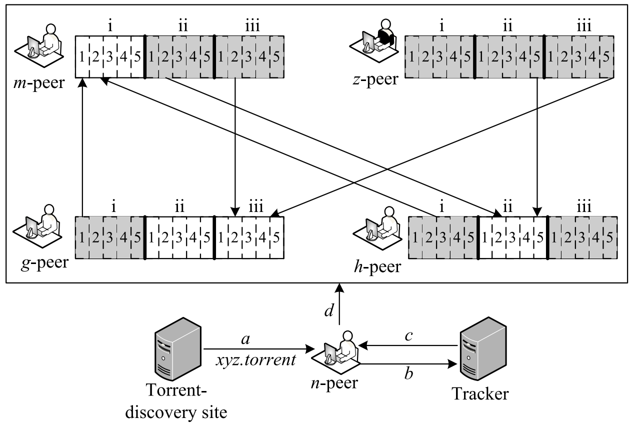

FIGURE 2. Illustration of data exchange among peers.

the same file are grouped to form a *swarm* for that file and its pieces are exchanged between the peers cooperatively. If a peer is missing a particular piece of the file, it will request and download blocks of that piece from other neighboring peers in the same swarm. For a file to be useful, peers must download all the pieces that constituted the file.

One of the key Quality of Service (QoS) parameters of P2P file-sharing systems is the *file-download time*, which is the time taken to download the entire file's pieces. The file-download time is not defined for any file (i.e. it has no limit) and it varies from peer to peer. During the download process, pieces are downloaded in any order, i.e. the order of pieces' download is not defined. Since each piece is made up of several data blocks, therefore, different blocks belonging to the same piece can be downloaded from different peers that have already obtained that piece. The already downloaded pieces are kept in the peer's buffer to serve other peers downloading the same file as illustrated in Fig. 2 where peers exchange data blocks of a file which consists of three (3) pieces each of which is made up of five (5) blocks. When a piece (say *i*-piece) is successfully downloaded by a peer, it will notify all its neighbors that it has *i*-piece in its buffer; therefore, interesting peers can forward their requests. In Fig. 2, *m*-peer, *g*-peer, and *h*-peer are downloaders, because they do not have the complete file, while *z*-peer is a seed.

A special node known as a *Tracker* is used to facilitate the download process. The Tracker helps downloaders to find other peers (downloaders and seeds) that are currently participating in a given torrent-session. To initiate a download process of the interested file (e.g. a file *xyz*), a new downloader (*n*-peer in Fig. 2) will first search and download the corresponding torrent file *xyz.torrent* from the torrent-discovery site (in Fig. 2, procedure *a*), the *torrent file* with extension *.torrent* contains the metadata about the file to be downloaded as well as the address of the Tracker that coordinates the data exchange in the swarm. Then the new downloader's *client* (a computer program which is used for file-sharing using BitTorrent Protocol) will contact the Tracker provided in the torrent file *xyz.torrent* (in Fig. 2, procedure *b*). The Tracker will respond by sending a random list of some peers (downloaders and seeds) which may have a part of the file or the whole file (in Fig. 2, procedure *c*). The random list of peers returned to the new downloader by the Tracker is called *peer-set*. The new downloader will then establish a connection with some of the peers in the peer-set as its neighbors and finds out which piece is available for download in each neighboring peer's buffer (in Fig. 2, procedure *d*). Then, the new downloader will send its request to download different blocks belonging to the same piece to different neighboring peers in its peer-set. If the number of members within the n-peer's peer-set falls below a given threshold due to churn, it will re-contact the Tracker to obtain additional members in its peer-set. A swarm can be viewed as a collection of interconnected peer-sets.

One of the key challenges of P2P file-sharing systems is free-riding. To deter free-riding act in BitTorrent as well as to maintain the balance between upload and download traffics, a choking algorithm is used. Each downloader tries to maximize its downloading rate via *tit-for-tat* (TFT) approach, where a downloader will collaborate only with partners from its peer-set that are cooperative (i.e. from which it receives mutual service). This is achieved through the choking procedure [38], [42].

Choking means temporary refusal to upload data to a non-cooperating downloader. In contrast, *unchoking* means uploading data to a downloader. Each peer is allowed to select and unchoke at once a limited number of downloaders that send them download-requests, this number is defined by the system's *UnchokeDefaultValue* (in BitTorrent, *UnchokeDefaultValue* $= 4$). Since the unchoke capacity is limited, therefore, each time, peers select and unchoke the best cooperative downloaders (in this case, seeds select and unchoke those downloaders with the best upload rates).

Each downloader is allowed to explore and find better cooperative partners from its peer-set. This is achieved through optimistic unchoking. With optimistic unchoking, each peer will randomly select and unchoke one additional downloader from which it received download request (i.e. in addition to the *UnchokeDefaultValue* allowed), in this case, each peer is allowed to upload to the maximum of *UnchokeDefaultValue* $+ 1$ downloaders. Since optimistic unchoking is used to discover the best cooperative partners, therefore, to maintain the maximum allowed unchokes, whenever a peer optimistically unchokes a random downloader, it will also select and temporarily choke the least cooperative downloader among its partners. However, with an optimistic unchoking policy, a free-rider may decide to join the network with different fake identities to obtain multiple data blocks periodically from different optimistic unchokers, and in this case, a free-rider may even download the entire file faster than cooperative downloaders [40].

IV. DATA EXCHANGE WITH THE SHARE-RATIO-BASED INCENTIVE MECHANISM

In this section, the proposed share-ratio-based incentive mechanism will be described via a simple model, to this end, consider a swarm with a single peer-set which consists of

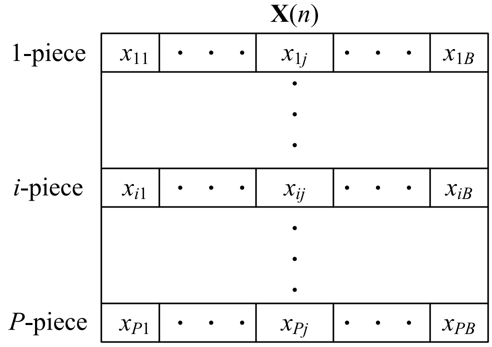

FIGURE 3. State of n-peer's buffer.

a set of peers $\mathcal{N}$ sharing a file using BitTorrent Protocol, where $\mathcal{N}_1 \in \mathcal{N}$ are downloaders and $\mathcal{N}_2 \in \mathcal{N}$ are seeds, $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$, $|\mathcal{N}| = N$, $|\mathcal{N}_1| = N_1$, $|\mathcal{N}_2| = N_2$ and $N_1 + N_2 = N$. The file being shared is fragmented into pieces of equal sizes and each piece is further fragmented into blocks of data. Denote by P the total number of pieces obtained after the file's fragmentation and each piece is further divided into B data blocks. Assume that each peer is provided with a buffer of size $P \times B$ for the storage of file's data blocks, i.e. for each i-piece, B buffer positions are reserved for the storage of its data blocks. Denote by $\mathbf{X}(n) = (x_{ij}(n))$ the state of n-peer's buffer (Fig. 3), where the vector $\mathbf{x}_i(n) = (x_{i1}, \ldots, x_{ij}, \ldots, x_{iB})$ shows the state of i-piece in n-peer's buffer, $x_{ij}(n) = 1$ if j-position is occupied with data block of i-piece, otherwise, $x_{ij}(n) = 0$. If for i-piece, $\forall j, j = 1, \ldots, B$, $x_{ij}(n) = 1$, then n-peer has fully downloaded i-piece and is ready for sharing. If $\forall i, i = 1, \ldots, P$ and $\forall j, j = 1, \ldots, B$, $x_{ij}(n) = 1$, then n-peer has successfully downloaded all the pieces and subsequently became a seed. Peers exchange buffer maps in order to indicate their available pieces for sharing.

After the fragmentation process, the file's pieces are orderly arranged and their serial numbers are used as their unique identifiers (ID), as such, each block of a piece is referenced via its parent's ID. However, during the download process, downloaders could request and download pieces randomly, i.e. regardless of their order. If the ID of i-piece is i, $1 \leq i \leq P$, then when a downloader will request for a j-block of i-piece from n-peer, the request will contain a value V defined by Eq. (1).

$$V = i + 0.01 \cdot j, \tag{1}$$

where j is the index of the requested block contained in i-piece.

For example, if a downloader will request for the 16^{th} block of the 25^{th} piece, then the request will contain a value $V = 25.16$.

Let $\mathcal{P}_{\mathrm{n}}$ be the set of pieces successfully downloaded by n-peer, $n \in \mathcal{N}_1$, then *Availability* (i.e. the number distributed file's copies) is denoted as A and expressed as

$$A = \frac{|\bigcup_{n=1}^{N_1} \mathcal{P}_{\mathrm{n}}|}{P} + N_2. \tag{2}$$

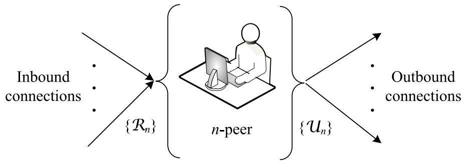

FIGURE 4. ***n*-peer with inbound and outbound connections.**

Note that even if $N_2 = 0$ (i.e. without seeds), it may be possible that $A = 1$, in this case, downloaders alone could reconstruct the file, therefore, to maintain $A \geq 1$, it is paramount to prevent free-riding and to encourage cooperation so as to maximize the replication and circulation of pieces among downloaders. A torrent-session will end when $A < 1$.

Assume that the process of data exchange among peers is slotted into time slots and during each time slot t_l, peers exchange data blocks. Each peer in the system has an *age* denoted as τ_n, $n \in \mathcal{N}$, which is the time from the moment a peer joined the session to current time slot t_l.

Furthermore, peers in the system can be categorized as either old or young, to that end, let σ_n be the *incubation period* of n-peer, which is the period elapsed between the time it joined the session to the time it has downloaded a portion of the file. To obtain the incubation period, let δ_n be the n-peer's download bandwidth and F the size of the file being shared, hence, σ_n is expressed as

$$\sigma_n = \lambda \cdot \frac{F}{\delta_n}, \tag{3}$$

where $0 < \lambda < 1$.

Equation (3) was formulated to fairly set peers' incubation periods based on their download bandwidths.

Denote by a_n the age status of n-peer, i.e.

$$a_n = \begin{cases} 1 & \text{if } \tau_n \geq \sigma_n, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Therefore, from Eq. (4), n-peer is categorized as *old* if $a_n = 1$ (i.e. its current age is greater than or equals to its incubation period) and *young* if $a_n = 0$.

During each time slot, a peer has two connections, i.e. inbound and outbound connections. The inbound connections are connections formed due to the *requests* for data blocks sent to n-peer from neighboring downloaders, while the outbound connections are connections formed while *uploading* the requested data blocks to downloaders (Fig. 4).

Let $\mathcal{R}_n$ be the set of inbound connections to n-peer, where $|\mathcal{R}_n| = r_n$ is the number of n-peer's inbound connections, and $\mathcal{U}_n$ the set outbound connections from n-peer, where $|\mathcal{U}_n| = u_n$ is the number of n-peer's outbound connections. During each time slot, the maximum number of uploads allowed per peer is U_{max}, as such, $u_n \leq U_{max}$ and $\mathcal{U}_n \subseteq \mathcal{R}_n$, $n \in \mathcal{N}$. Let $\mathcal{Q}_n$ and $\mathcal{Y}_n$ be the set of old and young downloaders respectively that have sent their requests to n-peer during time slot t_l, which are expressed by Eq. (5) and Eq. (6), i.e.

$$\mathcal{Q}_n = \{k \in \mathcal{R}_n | a_k = 1\}, \quad k \neq n, \tag{5}$$
$$\mathcal{Y}_n = \{k \in \mathcal{R}_n | a_k = 0\}, \quad k \neq n, \tag{6}$$

where $\mathcal{R}_n = \mathcal{Q}_n \cup \mathcal{Y}_n$, $|\mathcal{Q}_n| = q_n$, $|\mathcal{Y}_n| = y_n$, $n \in \mathcal{N}$, $k \in \mathcal{N}_1$ and $r_n = q_n + y_n$.

Denote by α_{max} and β_{max} the maximum number of old and young downloaders respectively, to which n-peer is allowed to upload data blocks during each time slot t_l, such that $\alpha_{max} + \beta_{max} = U_{max}$. Since it is possible that for n-peer, $n \in \mathcal{N}$, during time slot t_l to have $U_{max} < q_n + y_n$ with either $q_n < \alpha_{max}$ or $y_n < \beta_{max}$, therefore, considering these possibilities, the following auxiliary parameters α_n and β_n (defined in Eq. (7) and Eq. (8) respectively) were introduced to augment α_{max} and β_{max}, and this was done in order to fully utilize the maximum allowed uploads (i.e. U_{max}) for each peer in the network.

$$\alpha_n = \begin{cases} (\beta_{max} - y_n) + \alpha_{max} & \text{if } y_n < \beta_{max}, \\ \alpha_{max} & \text{otherwise.} \end{cases} \tag{7}$$

$$\beta_n = \begin{cases} (\alpha_{max} - q_n) + \beta_{max} & \text{if } q_n < \alpha_{max}, \\ \beta_{max} & \text{otherwise.} \end{cases} \tag{8}$$

In this case, the number of outbound connections for n-peer during each time slot is $u_n = \alpha_n + \beta_n$, where α_n and β_n define the number of uploads to old and young downloaders by n-peer respectively. Note that in this case, still, the number of uploads by n-peer will never exceed the maximum allowed, i.e. U_{max}.

Denote by μ_n and d_n the number of data blocks uploaded and downloaded cumulatively by n-peer from the moment it joined the session to the current time slot, $n \in \mathcal{N}$, respectively expressed by Eq. (9) and Eq. (10), i.e.

$$\mu_n = \sum_{t=0}^{t_l} \mu_n(t), \tag{9}$$
$$d_n = \sum_{t=0}^{t_l} d_n(t), \tag{10}$$

where $\mu_n(t)$ and $d_n(t)$ are the number of data blocks uploaded and downloaded by n-peer at time $t > 0$ respectively.

Denote by c_n the cumulative share-ratio of n-peer over the period τ_n of its stay in the session, i.e.

$$c_n = \frac{\mu_n}{d_n}, \tag{11}$$

where μ_n and d_n are obtained from Eq. (9) and Eq. (10).

For a new downloader, the initial values of $\mu_n(t)$ and $d_n(t)$ are given by Eq. (12) and Eq. (13).

$$\mu_n(0) = 1. \tag{12}$$
$$d_n(0) = \begin{cases} \sum_{i=1}^{P} \sum_{j=1}^{B} x_{ij} & \text{if } \mathbf{X}(n) \neq \mathbf{0}, \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

Hence, the n-peer's initial share-ratio is

$$c_n(0) = \frac{\mu_n(0)}{d_n(0)}. \tag{14}$$

In this case, Eq. (13) is to ensure that a new downloader's buffer allocated for the file is empty and to prevent a white-washing attack.

During each time slot t_l, downloaders select *target peers* to request data blocks (a target peer could be a seed or a downloader). To identify a free-rider, a threshold value C is introduced, $0 < C \leq 1$, such that a downloader with an age status $a_n = 1$ (i.e. attained old age) and with a cumulative share-ratio less than the threshold value C (i.e. $c_n < C$) is identified by its target peers as a free-rider.

However, for a peer-set with a high seed-to-downloader ratio, an altruistic downloader may attain old age with $c_n < C$ since the chance of its selection as a target peer by other downloaders in the peer-set is slim, therefore, the use of share-ratio c_n to identify free-riders will be unfair to altruistic downloaders in such scenarios. To make the proposed incentive mechanism gentle to downloaders for a scenario of a peer-set with a high seed-to-downloader ratio (in such scenarios, there is no threat to the downloading performance, especially when the ratio is very high), let s_n be the *share-index* of n-peer expressed as

$$s_n = \gamma c_n + (1-\gamma)\frac{N_2}{N_1}, \tag{15}$$

where $\gamma = 1 - \frac{1}{N_1}$, $N_1 \neq 0$ (for a scenario with $N_1 = 0$, $s_n \to \infty, n \in \mathcal{N}_2$).

From Eq. (15) it can be noted that for a scenario of high seed-to-downloader ratio, the inequality $s_n > c_n$ will hold, however, with a gradual decrease in seed-to-downloader ratio, n-peer's share-index s_n converges to its actual share-ratio c_n, consequently, considering the range of C, the use of share-index s_n will be more suitable than the use of share-ratio c_n to identify free-riders for any combination of seeds and downloaders in a peer-set, hence, a downloader with $a_n = 1$ and $s_n < C$ is identified as a free-rider by a target peer.

During the time slot t_l, each peer in the session is characterized by a pair of parameters defined by Eq. (16).

$$\Omega_n(t_l) = \langle a_n, s_n \rangle, \quad n \in \mathcal{N}. \tag{16}$$

During each time slot, each downloader is allowed to send the maximum requests of ρ_{max} for data blocks (in this case, one request per neighboring peer). Request for a data block sent by n-peer to m-peer (as its target peer), $n \neq m$, $n \in \mathcal{N}_1$, $m \in \mathcal{N}$ during time slot t_l (Fig. 5) is expressed by Eq. (17),

$$\rho_n = (\Omega_n(t_l - 1), V, m), \tag{17}$$

where $V \leq P + 0.01 \cdot B$ (Eq. (1)).

To encourage cooperation, target peers that received requests will select and upload to the best cooperating downloaders based on their share-indexes.

If during time slot t_l for a downloader, the event FR expressed by Eq. (18) is *true*, then it will be marked by its target peers as a *free-rider* in the current time slot t_l, subsequently, *all* the requests sent by that downloader will be *rejected*.

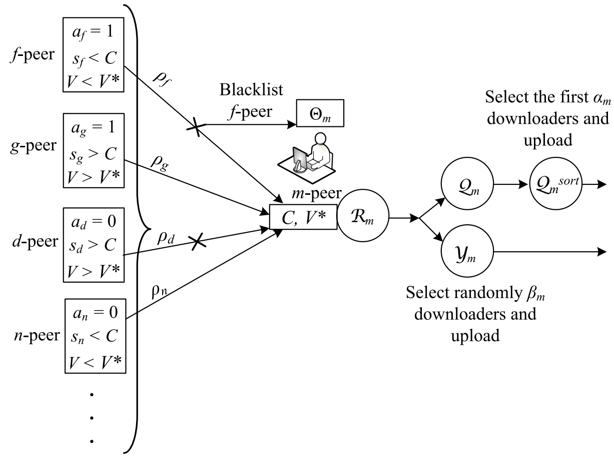

FIGURE 5. Requests sent by downloaders to m-peer as a target peer, $m \in \mathcal{N}$.

$$FR = \{(a_n = 1) \cap (s_n < C)\}, \quad n \in \mathcal{N}_1. \tag{18}$$

From Eq. (4) and Eq. (18), it can be seen that only old downloaders are subjected to free-riding screening during each time slot. The incubation period σ_n was introduced to serve as a *grace period* for newcomers. In Fig. 5, f-peer's request was rejected for failing the free-riding screening as provided in Eq. (18).

To cut further social relationships with free-riders, peers maintain a *Blacklist* where they place any downloader they marked as a free-rider. Denote by Θ_n the set of blacklisted downloaders by n-peer, $n \in \mathcal{N}$. To avoid wasted requests (i.e. requests that will not be served), downloaders do not send download requests to those they have blacklisted, as illustrated in Fig. 5 and Fig. 6, m-peer blacklisted f-peer after failing the free-riding screening, hence, m-peer will stop sending download-requests to f-peer, conversely, f-peer (even though blacklisted by m-peer) can still send download-request to m-peer as well as to other peers in the system. However, it can be noted that the more f-peer keeps sending download-requests, the more it loses social friends, consequently, the more it loses the chance of upgrading its share-ratio since those downloaders that blacklisted it, will no longer send it download-requests.

The system's Tracker also keeps its Blacklist; whenever it is re-contacted by a free-rider for more social friends, the free-rider will be blacklisted by the Tracker and the request will not be entertained (the Tracker also identifies free-riders by using the free-riding condition given by Eq. (18)), in this way, if a new downloader arrived and contacted the Tracker, in response, the Tracker will send a peer-set to the new downloader without those in its Blacklist, subsequently, all the new arriving downloaders will not know about the existence of f-peer.

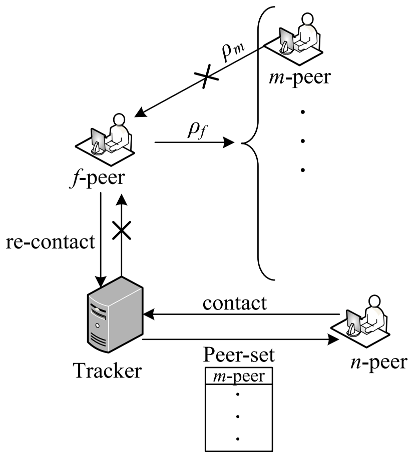

FIGURE 6. Isolation of a free-rider.

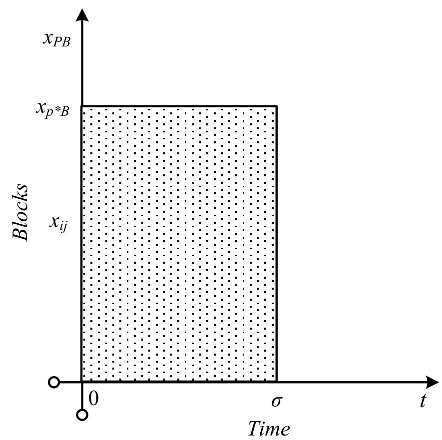

FIGURE 7. Data blocks allowed to be requested by young downloaders.

However, f-peer can decide to increase its upload capacity and still upgrade its share-ratio, provided that it has not been blacklisted by all downloaders in its peer-set. By increasing its upload capacity, f-peer can exchange data blocks reciprocally with other downloaders that have not so far marked it as a free-rider (i.e. those downloaders that have not received download-request from f-peer when it was in a free-riding mode). If after some time, m-peer or the Tracker received a request from f-peer and discovered that it is out of the free-riding mode, then m-peer or the Tracker will un-blacklist f-peer, in this way, f-peer can be able to recover. On the contrary, if f-peer decided to free-ride continuously until it is blacklisted by all the downloaders in its peer-set, then f-peer will be *isolated* (Fig. 6), consequently, its download time will be indefinite and will have to voluntarily exit the session since all the newcomers will not know about its existence due to the Tracker's action.

Free-riders could avoid a penalty by leaving the network before they attain old age and rejoin with different identities (whitewashing attack) to continue misbehaving, as such, in the proposed share-ratio-based incentive mechanism, young downloaders are only allowed to request and download a subset of the file's pieces, to this end, a *demarcating piece* with ID p^*, $p^* < P$, is set as

$$p^* = \lfloor \epsilon \cdot P \rfloor, \tag{19}$$

where $0 < \epsilon < 1$.

In this case, the request of a downloader with age status $a_n = 0$ (i.e. young downloader) is considered only if the value V of the block requested is less than or equals to $V^* = p^* + 0.01 \cdot B$, i.e. $V \leq V^*$ (Fig. 5 and Fig. 7). Therefore, if during time slot t_l for a downloader, the event DF expressed by Eq. (20) is *true*, then the downloader's request will be *rejected*.

$$DF = \{(a_n = 0) \cap (V > V^*)\}, \quad n \in \mathcal{N}_1. \tag{20}$$

Note that ϵ is a protocol's configuration parameter, while p^* and V^* are generated immediately after the file's fragmentation, and are distributed to all peers via the downloaded .torrent file. In Fig. 5, d-peer's request was rejected (even though not in a free-riding mode) for violating the defaulting condition given by Eq. (20).

With download restriction imposed on young downloaders, their uploads are also restricted since they can only upload what they have already downloaded (e.g. they cannot request and download the last piece P, as such, they cannot upload it to others), therefore, the parameter C used in the free-riding condition (Eq. (18)) can be set as $C \leq \epsilon$.

Since the file is useless to a downloader, if at least a single piece of the file is missing in its buffer, therefore, to be gentler to newcomers and to increase the circulation and replication of file's pieces, the demarcating piece p^* can be set very close to P. It can be noted that in this case, even if a misbehaving downloader decided to whitewash several times until it obtained a large portion of the file's pieces, the module responsible for the computation of share-ratio defined by Eq. (9) to Eq. (14) will ensure proper compensation if such downloader decided to behave accordingly in order to obtain the remaining file's pieces.

Generally, in the proposed share-ratio-based incentive mechanism, downloader's request ρ_n (Eq. (17)) will be rejected by a target peer if at least either of Eq. (18) or Eq. (20) is *true*, as expressed in Eq. (21).

$$\rho_n \rightarrow \begin{cases} \textit{reject} & \text{if } FR \cup DF = \textit{True}, \\ \textit{consider} & \text{otherwise.} \end{cases} \tag{21}$$

Note that the proposed share-ratio-based incentive mechanism is not susceptible to Sybil attack since any node created with a fake identity will be treated as a single entity, and provided that the node will request data, it must be screened based on conditions given by Eq. (18) and Eq. (20). Additionally, the proposed incentive mechanism is not a reputation-based incentive, therefore, a collusion attack is not a threat.

Figure 8 provides a simple illustration of client's interface which shows:

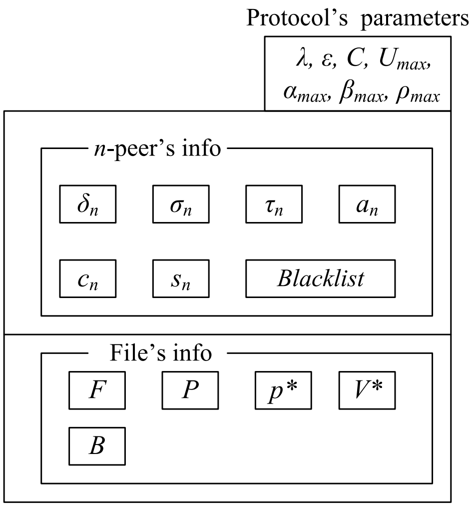

FIGURE 8. n-pee's client interface.

TABLE 1. *n*-peer's key parameters.

Parameter	Description
τ_n	Age
δ_n	Download bandwidth
σ_n	Incubation period
a_n	Age status
c_n	Share-ratio
s_n	Share-index
Θ_n	Set of blacklisted downloaders

- Peer's information, such as a_n, s_n, c_n, etc. (Table 1).
- File's information obtained from the file's metadata (provided in the downloaded .torrent file), such as F, V^*, etc. (Table 2).
- Protocol's configuration parameters, such as λ, ϵ, C, etc. (Table 3).

Note that Fig. 8 captures only the parameters needed by the proposed incentive mechanism.

It can be seen that it is easy to identify a free-rider and a defaulter based on Eq. (18) and Eq. (20). Download-requests sent by downloaders contain their age status indicator a_n, share-index s_n, as well as the value of the requested block V (as expressed in Eq. (17)), as such, whenever a peer received a download-request, it can be able to identify whether the requester is a free-rider, defaulter or cooperator based on the distributed information (C and V^*), (Fig. 5 and Fig. 8).

The actions performed by peers in the system during time slot t_l are described in the following steps (Fig. 5 and Fig. 9):

Step. i At the beginning of time slot t_l, all downloaders will send their requests $\rho_n = (\Omega_n(t_l - 1), V, m)$ to their target peers, $n \in \mathcal{N}_1$, $m \in \mathcal{N} \setminus \Theta_n$, $m \notin \Theta_n$;

TABLE 2. File's parameters.

Parameter	Description
F	File's size
P	Number of pieces which constituted the file
p^*	Demarcating piece
B	Number of blocks contained in each piece
V^*	The maximum value allowed in the young downloaders' requests (i.e. a value which corresponds to the last block of the demarcating piece)

TABLE 3. Protocol's configuration parameters.

Parameter	Description
C	Threshold value for free-riding screening
λ	Value for setting peer's incubation period
ϵ	Value for setting the file's demarcating piece
α_{max}	Maximum number of allowed uploads to old downloaders
β_{max}	Maximum number of allowed uploads to young downloaders
U_{max}	Maximum number of allowed uploads per peer during each time slot ($U_{max} = \alpha_{max} + \beta_{max}$)
ρ_{max}	Maximum number of allowed requests to be dispatched by downloaders during each time slot

Step. ii Each n-peer in the network, $n \in \mathcal{N}$ will:

- **Receive** download-requests from $r_n = |\mathcal{R}_n|$ downloaders;
- If for $k \in \mathcal{R}_n$, $FR = True$ (Eq. (18)), then **reject** the request ρ_k and **blacklist** k-peer;
- If for $k \in \mathcal{R}_n$, $DF = True$ (Eq. (20)), then **reject** the request ρ_k;
- **Separate** from $\mathcal{R}_n$ young downloaders from old downloaders thereby forming $\mathcal{Q}_n$ and $\mathcal{Y}_n$, $|\mathcal{Q}_n| = q_n$, $|\mathcal{Y}_n| = y_n$, as defined in Eq. (5) and Eq. (6) respectively;
- If $q_n < \alpha_{max}$, then $\beta_n \leftarrow (\alpha_{max} - q_n) + \beta_{max}$; else $\beta_n \leftarrow \beta_{max}$;
- If $y_n < \beta_{max}$, then $\alpha_n \leftarrow (\beta_{max} - y_n) + \alpha_{max}$; else $\alpha_n \leftarrow \alpha_{max}$;
  - ◇ If $\alpha_n + \beta_n < q_n + y_n$, then **sort** downloaders in $\mathcal{Q}_n$ in descending order of their share-indexes, such that $\mathcal{Q}_n^{sort} = \{\ldots, g, w, \ldots\}$, where $s_g \geq s_w$:
    - ⊕ **Select** the first α_n downloaders from $\mathcal{Q}_n^{sort}$ and upload their requested data blocks;

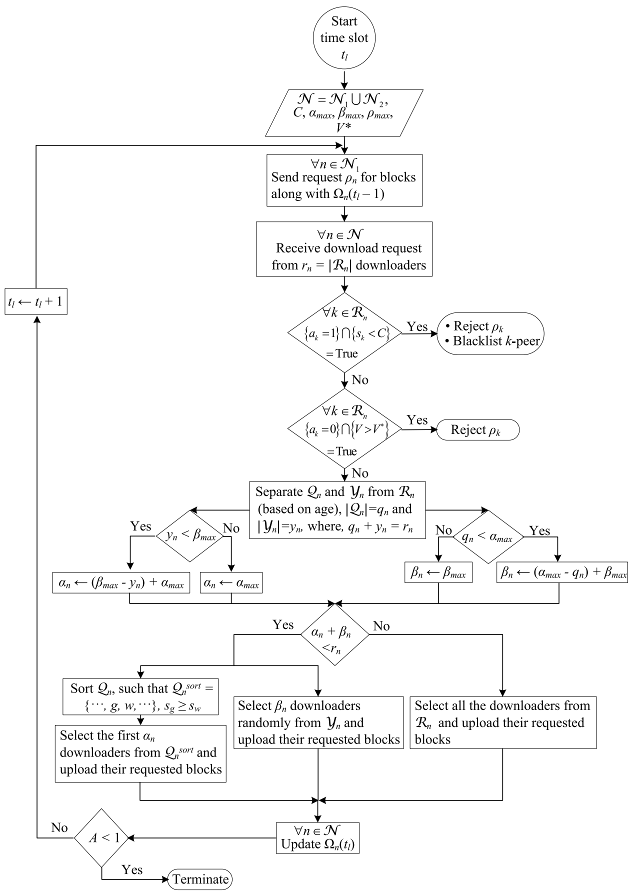

FIGURE 9. Flow chart of actions performed by peers during time slot t_l.

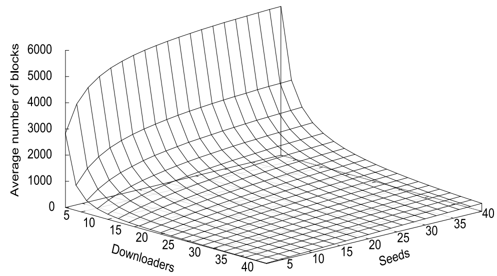

FIGURE 10. Average number of downloaded blocks after 400 slots vs. number of seeds and downloaders.

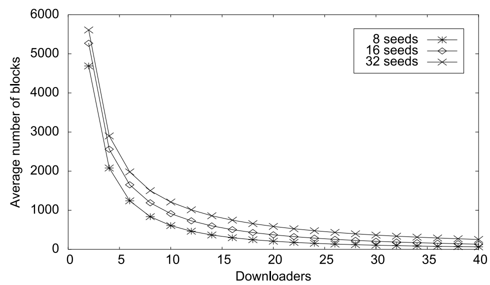

FIGURE 11. Average number of downloaded blocks after 400 slots vs. number of downloaders.

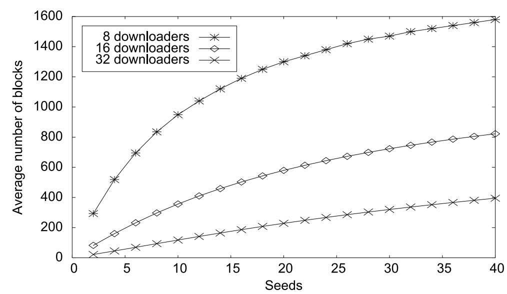

FIGURE 12. Average number of downloaded blocks after 400 slots vs. number of seeds.

⊕ **Select** randomly β_n young downloaders from $\mathcal{Y}_n$, and upload their requested data blocks;

◇ Else if $\alpha_n + \beta_n \geq q_n + y_n$, then **select** all downloaders in $\mathcal{R}_n$ and upload their requested data blocks;

Step. iii At the end of time slot t_l, for all peers, the information contained in $\Omega_n(t_l)$, $n \in \mathcal{N}$ (Eq. (16)) will be updated and used by dowloaders during the next time slot $t_l + 1$ to request data blocks;

Step. iv If $A \geq 1$, then for next time slot $t_l + 1$, repeat all the steps i - iii.

V. SIMULATION EXPERIMENTS

For the experiments, the proposed share-ratio-based incentive mechanism was implemented on a Java-based simulator, i.e. *PeerSim*. A file of 100 MB was introduced and split into pieces of 256 KB, each piece is further split into blocks of 16 KB. During the experiments, time is divided into time slots, and the exchange of data blocks between peers occurs during time slots. All peers have the download capacity of 5 blocks/slot, i.e $\delta_n = 5$, $\forall_n \in \mathcal{N}$, and the maximum number of allowed upload connections per peer during each time slot is 5, i.e. $U_{max} = 5$. The maximum number of requests allowed to be sent by downloaders during each time slot is 5, i.e. $\rho_{max} = 5$.

Firstly, the file downloading process will be analyzed by varying the number of downloaders and seeds in the system. In this case, all downloaders were introduced as full cooperators. All cooperators have enough upload capacity to satisfy the maximum number of allowed uploads (i.e. U_{max}) during each time slot. Each simulation with a given set of peers (downloaders and seeds) lasted for 400 slots. Results presented in Fig. 10 and Fig. 11 have shown that for a fixed number of seeds, as the number of downloaders in the system increases, the average number of downloaded blocks decreases, subsequently, the average file-download time increases as well.

From Fig. 10 and Fig. 12 it can be noted that for a fixed number of downloaders, as the number of seeds in the system increases, the average number of downloaded blocks increases, as such, the average file-download time decreases.

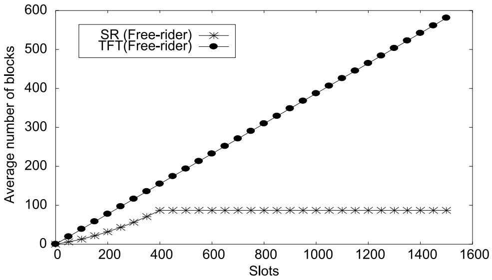

FIGURE 13. Action of SR-based and TFT mechanisms on free-riders for the first scenario.

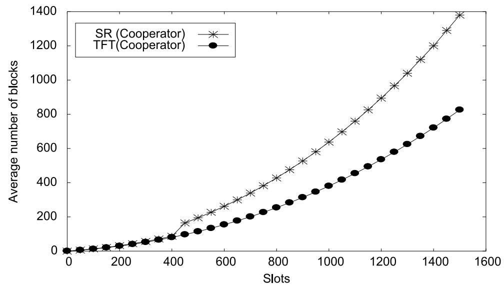

FIGURE 14. Average number of downloaded blocks for the first scenario.

Further, analysis of the effectiveness of the proposed incentive mechanism in preventing free-riding act and the impact of doing so on the system's performance will be conducted, to this end, a system with $N = 80$ peers was considered, where $N_2 = 20$ (are seeds) and $N_1 = 60$ (are downloaders). To define the incubation period σ_n for all peers, $\lambda = 0.32$ was used, since $\delta_n = 5, \forall_n \in \mathcal{N}$, from Eq. (3), $\sigma_n = 400, \forall_n \in \mathcal{N}$, therefore, the screening of downloaders for free-riding and defaulting started after the elapsed of the grace period, i.e. after the 400^{th} time slot. For free-riding condition (Eq. (18)), $C = 0.6$ was used as a threshold value. For defaulting condition (Eq. (20)), to set the demarcating piece p^*, $\epsilon = 0.77$ was used, as such, from Eq. (19), $p^* = 300$. Therefore, all downloaders can only request and download data blocks of pieces whose ID is not greater than 300 before the 400^{th} time slot, in this case, before this period, the value V contained in the requests of all downloaders should be $V \leq V^*$, where $V^* = 300.16$ (from Eq. (1)).

Experiments were conducted for two scenarios. In the **first scenario**, 75% of the downloaders were introduced as cooperators and the rest as free-riders. In the **second scenario**, 75% of the downloaders were introduced as free-riders and the rest as cooperators. In these experiments, free-riders were introduced with zero upload capacities.

Results of the analysis of the *first scenario* are presented in Fig. 13 and Fig. 14. From the graph in Fig. 13, it can be observed that with the proposed share-ratio (SR)-based incentive mechanism, a free-rider steadily kept obtaining data blocks and after the elapsed of the grace period, its download process was frozen, unlike with the BitTorrent's tit-for-tat (TFT) mechanism, a free-rider kept obtaining data blocks due optimistic unchoking. Also from the graph in Fig. 14, it can be noted that the average number of blocks obtained by a cooperator with both SR-based and TFT incentive mechanisms increases with an increase in time slots, however, it can also be noted that after the grace period, a jump has occurred in the curve of SR-based incentive mechanism, this is because, after the grace period, all free-riders were already blocked, and only cooperators could download, and with the blockage of free-riders, more upload resources were made available to cooperators, that is why the average number of downloaded blocks is higher with SR-based incentive mechanism than with TFT incentive mechanism after the grace period.

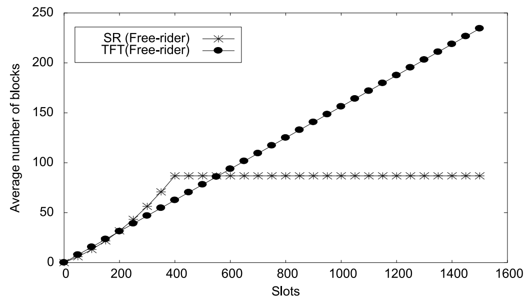

FIGURE 15. Action of SR-based and TFT mechanisms on free-riders for the second scenario.

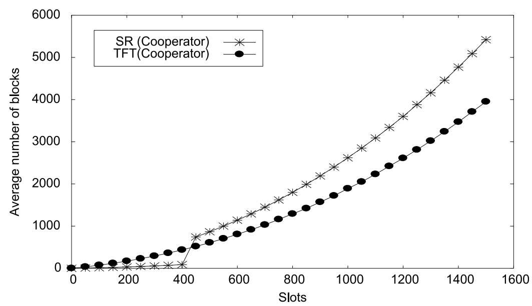

FIGURE 16. Average number of downloaded blocks for the second scenario.

Results of the analysis of the *second scenario* are presented in Fig. 15 and Fig. 16. Like the case of the first scenario, Fig. 15 shows that with SR-based incentive mechanism, free-rider's download process was frozen after the 400^{th} time slot, while with TFT, a free-rider kept obtaining blocks. Fig. 16 shows that before the grace period, the average number of downloaded blocks by cooperator is lower with SR-based incentive mechanism than with TFT, this is because, before the grace period, SR-based incentive does not distinguish free-riders from cooperators due to the imposed grace period, however, after the grace period, all the free-riders were blocked, and hence, a jump has occurred in the curve of SR-based mechanism after the grace period (as the case of the first scenario).

It can be noted that the average file-download time of SR-based mechanism will be lower in the case of the second scenario than with the first scenario (Fig. 17), this is because, with the blockage of free-riders after the elapsed of the grace period in the second scenario, the seeds outnumbered the

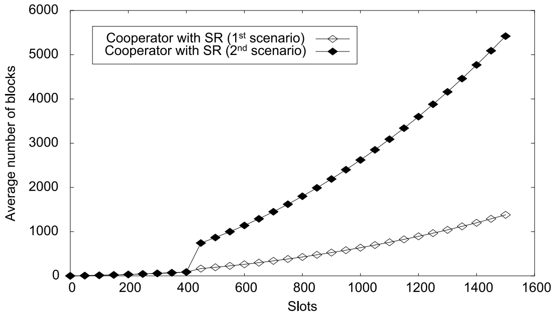

FIGURE 17. **Cooperator's average downloaded blocks with SR-based mechanism for the first and the second scenarios.**

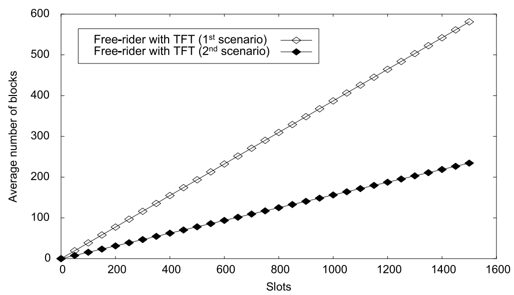

FIGURE 18. **Free-rider's average downloaded blocks with TFT for the first and the second scenarios.**

downloaders, hence, more upload resources are made available to the remaining downloaders.

Several research works conducted such as [19]–[21] revealed that in public P2P communities the second scenario is almost the case, therefore, the application of the proposed incentive mechanism will block the free-riders and allow only a few cooperators to download the file, in this case, the active cooperation and the availability of enough upload resources to cater for the need of downloaders would lead to a low average file-download time, as shown by Fig. 17.

However, the rate at which a free-rider kept obtaining blocks with TFT is higher in the case of the first scenario than in the case of the second scenario (Fig. 18), this is because, in the first scenario, the cooperators outnumbered the free-riders, as such, a free-rider have the chance of obtaining blocks from multiple cooperators due to optimistic unchoking. Since a session may last for a long period of time, a free-rider may end up having all the pieces of the file without any contribution to the system, especially when the file being shared is small in size and a large portion of the downloaders are cooperators (like the first scenario). Additionally, with TFT, a free-rider may decide to behave accordingly after obtaining a significant number of blocks. Unlike TFT, with the SR-based incentive mechanism, regardless of the number of cooperators or free-riders in the system, a free-rider will not obtain any block after its grace period has elapsed (Fig. 19).

In the experiments conducted for both the first and the second scenarios with the share-ratio-based incentive mechanism, the blocks obtained by downloaders were traced

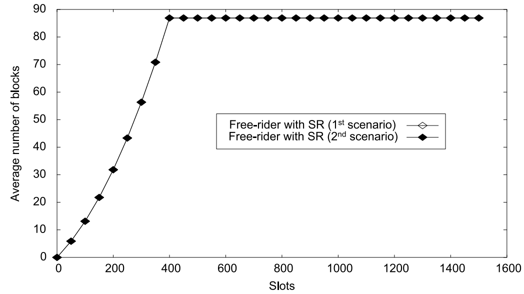

FIGURE 19. **Free-rider's average downloaded blocks with SR-based mechanism for the first and the second scenarios.**

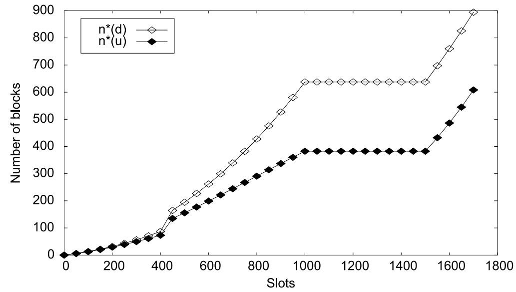

FIGURE 20. **Number of downloaded and uploaded blocks by *n**-downloader in the first scenario.**

before the elapse of the grace period, and none of the downloaders was able to download a block with a value $V > V^*$.

The graph presented in Fig. 20 has shown the number of downloaded and uploaded blocks by n^*-downloader (from the first scenario). After the grace period, n^*-downloader steadily decreases its upload rate, subsequently, its share-ratio decreases, as such, its chance of being selected for upload decreases as well. Finally, after 1000^{th} time slot, n^*-downloader failed the free-riding screening and its download process is frozen. Later, after the 1500^{th} time slot, n^*-downloader started to increase its upload rate, subsequently, raised its share-ratio, and after some time slots its share-ratio exceeded the required threshold value, as such, its number of downloaded blocks started to increase.

VI. CONCLUSION AND FUTURE WORK

An incentive mechanism based on share-ratio was proposed for sharing files using the BitTorrent protocol. With the proposed incentive mechanism, downloaders use their share-ratios to request data, and cooperation is encouraged by giving download priorities to downloaders with the best share-ratios. Additionally, with the proposed scheme, free-riders are isolated to enhance the downloading performance, and newcomers are fairly treated. The proposed incentive mechanism was designed in existing terminologies used in P2P file-sharing networks and does not require any central entity for its operations, therefore, it is easy to implement.

Future works would be focused on the: (1) analysis of the effectiveness of the proposed share-ratio-based incentive mechanism for P2P VoD and live streaming systems; and (2) problem of finding the optimal values of the protocol's configuration parameters for optimal performance.

REFERENCES

[1] *BitTorrent*. Accessed: Mar. 20 2021. [Online]. Available: http://www.bittorrent.com/
[2] *qBittorrent*. Accessed: Mar. 20 2021. [Online]. Available: https://www.qbittorrent.org
[3] *BitComet*. Accessed: Mar. 20 2021. [Online]. Available: https://www.bitcomet.com/en
[4] *BitLord*. Accessed: Mar. 20 2021. [Online]. Available: https://www.bitlord.com
[5] *μTorrent*. Accessed: Mar. 20 2021. [Online]. Available: http://www.utorrent.com/
[6] N. Masinde and K. Graffi, "Peer-to-peer-based social networks: A comprehensive survey," *Social Netw. Comput. Sci.*, vol. 1, no. 5, pp. 2–51, Sep. 2020, doi: 10.1007/s42979-020-00315-8.
[7] T.-H. Hsu and Y.-M. Tung, "A social-aware P2P video transmission strategy for multimedia IoT devices," *IEEE Access*, vol. 8, pp. 95574–95584, 2020.
[8] M. A. Silva, R. Bertone, and J. Schafer, "Topology distribution for videoconferencing applications," in *Proc. 10th Int. Conf. Netw. Future (NoF)*, Oct. 2019, pp. 90–97.
[9] S. Masood, M. A. Shahid, M. M. Sharif, and M. Yasmin, "Comparative analysis of P2P networks," *International J. Adv. Netw. Appl.*, vol. 9, no. 4, pp. 3477–3491, 2018.
[10] M. Amoretti and F. Zanichelli, "P2P-PL: A pattern language to design efficient and robust peer-to-peer systems," *Peer Peer Netw. Appl.*, vol. 11, no. 3, pp. 518–547, May 2018.
[11] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1010–1019, May 2006.
[12] J. Altmann and Z. B. Bedane, "A P2P file sharing network topology formation algorithm based on social network information," in *Proc. IEEE INFOCOM Workshops*, Apr. 2009, pp. 1–6, doi: 10.1109/INFCOMW.2009.5072101.
[13] B. N. Levine, C. Shields, and N. B. Margolin, "A survey of solutions to the sybil attack," Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 224, 2006.
[14] N. K. Saini, V. K. Sihag, and R. C. Yadav, "A reactive approach for detection of collusion attacks in P2P trust and reputation systems," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 312–317, doi: 10.1109/IAdCC.2014.6779340.
[15] C. Zhang, P. Dhungel, D. Wu, Z. Liu, and K. W. Ross, "BitTorrent darknets," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9, doi: 10.1109/INFCOM.2010.5461962.
[16] Z. Liu, P. Dhungel, D. Wu, C. Zhang, and K. W. Ross, "Understanding and improving ratio incentives in private communities," in *Proc. IEEE 30th Int. Conf. Distrib. Comput. Syst.*, 2010, pp. 610–621, doi: 10.1109/ICDCS.2010.90.
[17] X. Chen, X. Chu, and Z. Li, "Improving sustainability of private P2P communities," in *Proc. 20th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2011, pp. 1–6, doi: 10.1109/ICCCN.2011.6005944.
[18] X. Chen, X. Chu, and J. Liu, "Unveiling popularity of BitTorrent darknets," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5, doi: 10.1109/GLOCOM.2010.5683894.
[19] S. B. Handurukande, A.-M. Kermarrec, F. Le Fessant, L. Massoulié, and S. Patarin, "Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems," *ACM SIGOPS Operating Syst. Rev.*, vol. 40, no. 4, pp. 359–371, Oct. 2006.
[20] M. Yang, Z. Zhang, X. Li, and Y. Dai, "An empirical study of free-riding behavior in the Maze P2P file-sharing system," in *Proc. 4th Peer Peer Syst.*, 2005, pp. 182–192.
[21] D. Hughes, G. Coulson, and J. Walkerdine, "Free riding on Gnutella revisited: The bell tolls?" *IEEE Distrib. Syst. Online*, vol. 6, no. 6, p. 1, Jun. 2005.
[22] H. T. Liu, Z. X. Huang, Y. Bai, and Y. H. Qiu, "Auction incentive mechanism in P2P," in *Proc. Int. Conf. Multimedia Ubiquitous Eng. (MUE)*, Apr. 2007, pp. 941–945, doi: 10.1109/MUE.2007.92.
[23] C. Wu, Z. Li, X. Qiu, and F. C. M. Lau, "Auction-based P2P VoD streaming: Incentives and optimal scheduling," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 8, no. 1S, pp. 1–22, Feb. 2012, doi: 10.1145/2089085.2089091.
[24] B. Alotibia, N. Alarifia, M. Abdulghanib, and L. Altoaimy, "Overcoming free-riding behavior in peer-to-peer networks using points system approach," in *Proc. Int. Workshop Peer Peer Archit., Netw., Syst. (P PANS)*, Leuven, Belgium, Apr./May 2019, pp. 1060–1065.
[25] J. Moon and Y. Cho, "A point-based inventive system to prevent free-riding on P2P network environments," in *Computational Science and its Applications—ICCSA* (Lecture Notes in Computer Science), vol. 6785, B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, and B. O. Apduhan, Eds. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-21898-9_39.
[26] F. L. Haddi, M. Rahmani, and M. Benchaba, "A credits-based economy for P2P systems," in *Proc. Int. Conf. Appl. Smart Syst. (ICASS)*, Nov. 2018, pp. 1–6, doi: 10.1109/ICASS.2018.8652019.
[27] C. Wang, H. Wang, Y. Lin, and S. Chen, "A lightweight currency-based P2P VoD incentive mechanism," in *Proc. 3rd Int. Joint Conf. Comput. Sci. Optim.*, 2010, pp. 272–276, doi: 10.1109/CSO.2010.232.
[28] C. Wang, H. Wang, Y. Lin, and S. Chen, "A currency-based P2P incentive mechanism friendly with ISP," in *Proc. Int. Conf. Comput. Design Appl.*, Jun. 2010, pp. 403–407, doi: 10.1109/ICCDA.2010.5540937.
[29] J. Dong, G. Wang, and M. Yang, "A virtual currency based incentive mechanism in P2P network," in *Proc. Hong Kong Conf.*, vol. 30. Singapore: IACSIT Press, 2012, pp. 21–26.
[30] I. Z. Mehr and M. D. T. Fooladi, "Token-based incentive mechanism for peer-to-peer video streaming networks," *J. Supercomput.*, vol. 75, pp. 6612–6631, May 2019, doi: 10.1007/s11227-019-02863-0.
[31] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.
[32] Q. Zhang, Y. Leng, and L. Fan, "Blockchain-based P2P file sharing incentive," in *Proc. IACR Cryptol. ePrint Arch.*, 2018, p. 1152.
[33] T.-M. Wang, W.-T. Lee, T.-Y. Wu, H.-W. Wei, and Y.-S. Lin, "New P2P sharing incentive mechanism based on social network and game theory," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 915–919.
[34] Y. Zhang and Y. Fang, "A fine-grained reputation system for reliable service selection in peer-to-peer networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1134–1145, Aug. 2007.
[35] R. Zhou, K. Hwang, and M. Cai, "GossipTrust for fast reputation aggregation in peer-to-peer networks," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1282–1295, Sep. 2008.
[36] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2004, pp. 825–830.
[37] M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, and H. Song, "Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing," *IEEE Access*, vol. 5, pp. 1382–1397, 2017.
[38] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. Workshop Econ. Peer Peer Syst.*, 2003, pp. 1–10.
[39] P. Rahimzadeh, C. Joe-Wong, K. Shin, Y. Im, J. Lee, and S. Ha, "SVC-TChain: Incentivizing good behavior in layered P2P video streaming," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2017, pp. 1–9.
[40] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free riding in BitTorrent is cheap," in *Proc. 5th Workshop Hot Topics Netw. (HotNet)*, Jan. 2009, pp. 85–90. [Online]. Available: http://dcg.ethz.ch/projects/bitthief/
[41] H. Nishida and T. Nguyen, "A global contribution approach to maintain fairness in P2P networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 6, pp. 812–826, Jun. 2010, doi: 10.1109/TPDS.2009.122.
[42] A. Legout, G. Urvoy-Keller, and P. Michiardi, "Understanding bittorrent: An experimental perspective," Inst. Res. Comput. Sci. Automat., Rocquencourt, France, Tech. Rep. inria-00000156v3, 2005, p. 16. [Online]. Available: https://hal.inria.fr/inria-00000156v3/document

AMINU ADAMU received the B.Sc. degree in computer science and mathematics, the M.Sc. degree in informatics and applied mathematics, and the Ph.D. degree in theoretical computer science from the Department of Informatics and Applied Probability, Peoples' Friendship University of Russia. He is currently an Associate Professor with the Department of Computer Science, Umaru Musa Yar'adua University, Nigeria. He is also working as an Associate Professor (on a technical assistance program) with the School of Information and Communications Technology, University of the Gambia. His research interests include P2P networks, video streaming systems, the IoT systems, fog computing, congestion control, and performance analysis of communication systems.