Journal of Internet Services and Applications, 2024, 15:1, doi: 10.5753/jisa.2024.4010

Fine Tuning of the BitCover Algorithm for Interactive VoD Streaming over 5G Cellular Networks

Carlo Kleber da S. Rodrigues [**Federal University of ABC** | ***carlo.kleber@ufabc.edu.br***]
Vladimir Rocha [**Federal University of ABC** | ***vladimir.rocha@ufabc.edu.br***]

Center for Mathematics, Computing and Cognition, Federal University of ABC, Av. dos Estados, 5001. Bairro Santa Terezinha. Santo André, SP, 09210-580, Brazil.

Received: 5 January 2024 • **Accepted:** 24 April 2024 • **Published:** 28 June 2024

Abstract The BitCover algorithm is an adaptation of the popular BitTorrent algorithm for interactive Video-on-Demand (VoD) streaming over 5G cellular networks. This algorithm has already proven to be an effective solution for granting adequate bandwidth utilization within each cell site, which turns out to be essential for unlocking the whole potential of the 5G technology. Despite its attractive performance, we though wonder if there is still space for optimization since four of its configuration parameters are identical to those of the original BitTorrent, being set with the same numerical values. These parameters are δ_t (unchoking time), N_p (number of neighbors a peer has), y (number of data slots a peer's upload capacity is divided into), and z (number of peers randomly selected in optimistic unchoking). To tackle this issue, we therefore carry out simulation experiments to hopefully determine a more adequate configuration setup for the three first parameters just mentioned, leaving the specific analysis of the last parameter (i.e., z) implicit since it is directly related to the third parameter (i.e., y). Among the major findings, we highlight that BitCover's original performance is enhanced at about 16.7% in terms of download rate, and at 50.1% in terms of discontinuity time. To complement this study, we also present a detailed competitive analysis against two other recent literature proposals, mainly to show the overall effectiveness of the optimized version of the BitCover algorithm. Within this context, our pivotal contribution is to offer helpful insights for designing protocols aimed at 5G cellular networks. Finally, this paper ends with general conclusions and outlines future directions.

Keywords: BitCover, 5G, Streaming, VoD, Interactivity, Mobility

1 Introduction

Introduced globally by major telecommunications companies in 2019, the 5G technology represents the fifth-generation standard for broadband cellular networks. It is intended to succeed the 4G technology and is estimated to have over 1.7 billion subscribers worldwide by 2025, according to the GSM Association [Sandvine, 2023; Pana *et al.*, 2022; Narayanan *et al.*, 2020; Xu *et al.*, 2020].

Similar to its predecessors, the 5G service coverage is organized into discrete geographic regions known as cell sites. Within a cell site, all 5G wireless devices establish connections to the Internet and conventional telephone networks using radio waves transmitted via local cell antennas. The primary benefit of these cutting-edge networks lies in their capability to offer astonishing download speeds, with the potential to reach up to 10 Gbps in the future. This impressive feat is made possible through a combination of technological advancements, such as extensive employment of massive multiple-input multiple-output (MIMO), sophisticated channel coding, and scalable modulation techniques [Narayanan *et al.*, 2020; Xu *et al.*, 2020].

Given the remarkable capabilities of 5G-technology-based communications, it is anticipated that Internet Service Providers (ISPs) will progressively leverage these emerging networks to provide connectivity for laptops and desktop computers, competing with established infrastructure like cable Internet. Moreover, 5G technology is poised to enable novel applications within the realms of the Internet of Things (IoT) and machine-to-machine (M2M) communication [Narayanan *et al.*, 2020; Xu *et al.*, 2020].

Nevertheless, this amazing technological scenario also yields some preoccupation due to the increasing mobile-data traffic volume that needs to be efficiently transmitted across the entire cellular-network architecture, i.e., from cell sites to the core network via the mobile backhaul network (and the other way around). This issue must be tackled, otherwise, communication links may turn out clogged. [Li *et al.*, 2022; Molner *et al.*, 2019; Li *et al.*, 2021]. In particular, a huge component of traffic undoubtedly comes from the skyrocketing popularity of streaming video-on-demand (VoD) services [Sandvine, 2023; Shrama *et al.*, 2020; Lin *et al.*, 2023; Hossain *et al.*, 2019].

Besides the well-known strategies based on mobile edge computing and caching techniques, another promising alternative to tackle the above issue involves a dedicated effort to enhance bandwidth utilization within the cell site through the implementation of efficient bandwidth-sharing algorithms [Sun *et al.*, 2019; Chen *et al.*, 2022; Zhang and Mao, 2019; Safavat *et al.*, 2020; Lin *et al.*, 2023]. With that in mind, we have then proposed the BitCover algorithm [Rocha and Rodrigues, 2023]. It is a BitTorrent-based algorithm for interactive VoD streaming over 5G cellular networks.

The BitCover algorithm has attracted attention due to its effective performance under a variety of multimedia streaming scenarios. Nonetheless, its parameter set has been partly defined according to the original BitTorrent proposal [Cohen, 2003; Legout *et al.*, 2005, 2006]. More precisely, four parameters of BitCover are the same as those of BitTorrent, being assigned with the same numerical value configuration. These parameters are δ_t (unchoking time), N_p (number of neighbors a peer has), y (number of data slots a peer's upload capacity is divided into), and z (number of peers randomly selected in optimistic unchoking). This fact leads one to wonder if there is still space for any performance optimization.

In this work, we therefore carry out detailed analyses based on simulations seeking to determine the most adequate configuration setup for the three first parameters just mentioned above. We explain that the fourth parameter (i.e., z) is directly related to the third parameter (i.e., y) and hence does not need to be analyzed. To complement this study, which may be seen as an extension of our previous work [Rocha and Rodrigues, 2023], we still present a competitive analysis against two recent literature proposals, mainly to show the effectiveness of the derived optimized version of the BitCover algorithm.

The primary contribution of this work is thus to provide the scientific literature with solid theoretical insights into multimedia projects, with a particular focus on enhancing interactive VoD streaming over 5G networks. The subsequent sections of this work are structured in the following manner. Section 2 briefly reviews the BitCover algorithm as originally proposed in [Rocha and Rodrigues, 2023]. Section 3 discusses some seminal and/or recent works related to our research problem. Section 4 presents the experiments to find out the most proper parameter configuration setup for the BitCover algorithm. In Section 5, we carry out the competitive analysis. Finally, in Section 6 we provide conclusions and future directions.

2 The BitCover algorithm

The core elements of the BitCover algorithm's design are as follows: (i) the creation of mobile node clusters to collectively share video pieces while playing back content, (ii) the utilization of WiFi-Direct technology to reduce the load on the 5G mobile-backhaul network when delivering video pieces, and (iii) the implementation of an innovative peer-selection strategy that enhances channel transmission efficiency by giving preference to peers with larger number of video pieces already downloaded into local buffers.

Within BitCover, the choice for utilizing either 5G or WiFi-Direct connections is determined by the physical separation between two endpoints, which are the peers engaged in sharing the video pieces. When the distance exceeds the radio range of the WiFi-Direct technology, i.e., the $radius$, the piece is routed through 5G channels in the mobile-backhaul network. Conversely, when the endpoints are within the WiFi-Direct radio range, they exclusively use a WiFi-Direct channel. This approach is carefully designed to harness the key capabilities of both WiFi-Direct and 5G communication technologies in a complementary manner.

The operation of BitCover is next explained by means of its *peer-selection* and *piece-selection* policies, respectively. For the sake of clarity, all BitCover's configuration parameters are given in Table 1, whose third column indicates whether or not the parameter is used in the BitTorrent proposal [Cohen, 2003], and the fourth column informs the parameters' default values. The goal and importance of each of these parameters in the algorithm's operation are detailed in Subsections 2.1 and 2.2. Additionally, let l be a system parameter denoting the number of pieces the file is divided into.

2.1 Peer-selection policy

The upload capacity of a peer is partitioned into y data slots, and the total number of neighbors a peer may have is N_p. A peer's neighbors consist of other peers with which it can both send and receive pieces of a given file L to be watched.

Let a peer g be a neighbor of another peer p. Peer g expresses interest in peer p when the latter possesses pieces that the former lacks. On the other hand, peer g has no interest in neighbor p when the latter has only pieces already owned by the former. Similarly to the case of the original BitTorrent protocol, the peer-selection policy of BitCover includes two recurring processes: the regular unchoking and the optimistic unchoking.

The regular unchoking is constituted by the two activities explained in the following, which occur at every δt seconds.

1. Neighbors that have manifested interest in peer p are ranked based on their average upload rate (in descending order). In other words, these peers are arranged according to their ability to consistently transmit pieces to peer p at a higher speed.
2. From the ordered list above, f peers are ranked based on descending piece-coverage criterion (explained below). Each of these peers is assigned one data slot, and the value of f is upper bounded by $y - 1$.

The piece-coverage criterion evaluates how many downloaded pieces a peer g already has within its *interior buffer* (see next subsection). The reason behind employing the piece-coverage criterion is to ensure that the chosen peer has a larger number of pieces available for sharing. In this case, the efficiency of network communication is enhanced, the channel utilization is optimized, and the peer-connection management overhead is reduced.

An illustration for the above understanding is provided in Figure 1. Consider that neighbors q, r, and s all express interest in a peer p. Peer p should give preference to the peer that has a larger number of downloaded pieces in its *interior buffer* V, detailed in the next subsection. In this case, peer p will then select peer r since it owns more downloaded pieces.

Regarding the optimistic-unchoking process, we have that peer p randomly chooses z peers (from its neighbor list) and allocates z data slots, assigning one slot to each chosen peer. This process is repeated every δt seconds as well.

2.2 Piece-selection policy

The piece-selection policy is determined by a strategy that integrates a *sliding window*, S_W, plus an *interior buffer*, V. This strategy is elaborated in what follows.

The video file L is split into l pieces: $1, 2, \ldots, l$. Also, let c_p be the piece mapping to the playback point currently

Table 1. BitCover's main configuration parameters and default values.

Parameter	Description	BitTorrent [Cohen, 2003]	Default values [Rocha and Rodrigues, 2023]
δt	Unchoking time, measured in seconds.	Yes	10
N_p	Number of neighbors a peer p has.	Yes	80
y	Number of data slots a peer's uploaded capacity is divided into.	Yes	4
f	Number of fastest neighbors.	No	30
$radius$	WiFi-Direct coverage, measured in meters.	No	100
k	Number of parts the remainder of the video L is divided into.	No	3
z	Number of peers randomly selected in optimistic unchoking. Each of these peers is assigned a data slot, which implies that z is directly related to y.	Yes	1
S_W	sliding window, measured in number of video pieces.	No	$s_w \approx 0.37 \times l$
V	Interior buffer, measured in number of video pieces.	No	$v \approx 0.4 \times w$

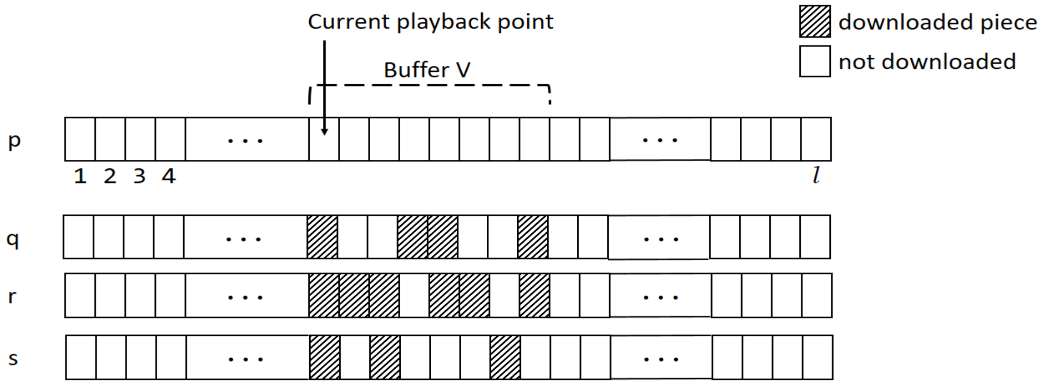


Figure 1. Piece-coverage criterion.

viewed by the user. In addition, consider s_w to be the size of the *window* S_W measured in number of pieces. Thus, the *window* S_W includes the pieces: $[c_p; c_p + s_w]$.

The *window* S_W undergoes dynamic updates while the pieces are viewed and interactive actions are executed. To illustrate this, let Δ_{play} be the number of viewed pieces. The piece in the first position of *window* S_W must be then updated to $(c_p + \Delta_{play})$, and its last position must be updated to $(c_p + s_w + \Delta_{play})$.

Now, consider v as the size (in number of pieces) of *buffer* V. The piece in the first position of V must be coincident with the piece in the first position of W. Prior to requesting the next piece from peer p, a peer g verifies whether all pieces in *buffer* V have already been obtained. If they have not (i.e., *buffer* V is not entirely filled), peer g will request the in-order piece that is missing within *buffer* V.

On the other hand, assume that *buffer* V is entirely filled and that the piece in the first position of W is x. The remaining portion of the video (i.e., $l - (v + x)$) is then subdivided into k equal parts, each one of size $D = (l - (v + x))/k$. In this case, peer g makes a probabilistic request for the next piece from one of the k parts, as explained next.

Each j-th part, $1 \leq j \leq k$, is selected using the probability $p_j = p_r/(j + 1)$, with p_r acquired from $\sum_{j=1}^{k} p_j = 1$. The piece to request within one of the k parts can be obtained in a sequential manner or following an alternative policy, like the rarest-first (i.e., the least replicated piece). In this study, we opt for the former approach since the performance results reported in the literature are satisfactory [Rocha and Rodrigues, 2023]. Figure 2 depicts this explanation in the case $k = 3$ and $x = 1$. Finally, s_w and v are estimated from the formulas derived in [Rodrigues and Rocha, 2021], namely $s_w \sim 0.37 \times l$, and $v \sim 0.4 \times s_w$.

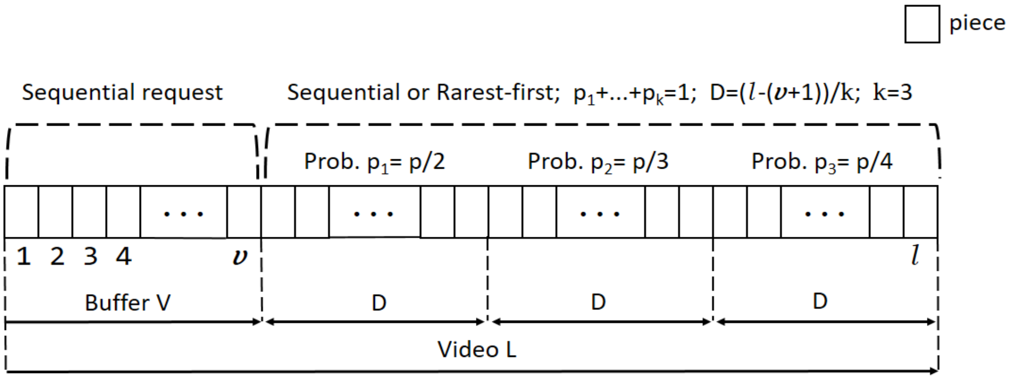


Figure 2. Graphical view of the video division in pieces, for $k = 3$ and $x = 1$.

3 Literature Review

Being aware of the numerous works related to the BitTorrent protocol in the scientific literature, this section mostly aims to discuss a few of the most pivotal proposals that aid in accomplishing the purpose of this work, thereby providing the reader with both a historical and a state-of-the-art overview of this research field. More specifically, some of the most seminal and/or recent BitTorrent adaptations for streaming over wireless networks are discussed herein. To simplify and make it easier to understand, the terms *node* and *peer* will have the same meaning from this point forward.

With a specific focus on Mobile Ad Hoc Networks (MANETs), the authors in [Sbai *et al.*, 2008; Krifa *et al.*, 2009; Sbai and Barakat, 2009; Salhi *et al.*, 2009] introduce the BitHoc protocol and various extensions of it falling within the cross-layer protocol classification. They use a peer-selection policy that seeks to balance between piece-sharing and piece-diversification efforts. Additionally, they propose the utilization of an optimal neighbor scope, which involves setting a hop count limit for routes that connect two peers. Network topology routing information is obtained through a proactive protocol known as Optimized Link State Routing (OLSR). While their approach demonstrates efficiency, validated through simulation experiments, it is worth noting that it does not adequately address client mobility and lacks support for client interactivity.

Within the layered-architecture protocol classification and also focused on MANETs, the authors in [Quental and Gonçalves, 2011] introduce the Mobile-BitTorrent protocol. This proposal considers two distinct categories of peers: *disseminator* and *common*. The *disseminator* periodically sends content messages via broadcast to its immediate 1-hop neighbors, and can respond to piece requests through unicast communication. On the other hand, the *common* is capable of responding to requests from any neighbor via unicast and has the capacity to receive messages via broadcast and unicast. The authors emphasize the importance of prudent message broadcasting to mitigate the risk of packet collisions. We must mention that the researches of [Casini *et al.*, 2016; Jenkac *et al.*, 2006; Leung and Chan, 2007] offer valuable insights into the efficient dissemination of content through broadcast. Despite demonstrating competitive performance in simulations when compared to the BitTorrent protocol, the primary advantage of Mobile-BitTorrent lies in its simplicity. However, it is important to note that the simulations do not account for either client interactivity or mobility.

In [Yang *et al.*, 2017], the authors introduce a system for streaming over cellular networks, referred to as MTV. More precisely, this system is based on the BitTorrent paradigm and combines mobile networks with the WiFi-Direct communication technology. Its *modus operandi* is to form clusters of nodes (physically close to each other) that are concurrently viewing the same video. Additionally, the MTV system not only enhances download speeds and reduces data traffic but also facilitates that nodes may dynamically join and leave the network. Nonetheless, while showing numerous advantages of the proposed system, such as improved speed, reduced traffic, and dynamic adaptability, the experiments are constrained to scenarios with a limited number of nodes. This limitation hinders the ability to draw more comprehensive conclusions.

The authors in [Rethfeldt *et al.*, 2018] introduce MeNTor, a cross-layer protocol specifically designed for Wireless Mesh Networks (WMNs). The system is developed according to three main aspects: (a) modifying BitTorrent's default peer-selection policy by incorporating the information of 802.11s MAC-layer as well as prioritizing the Airtime Link Metric (ALM) as selection criteria; (b) implementing limitations on concurrent uploads per peer while disabling uploads to random optimistic peers to reduce eventual wireless contention; and (c) deploying a so-called *generous* mode, i.e., altering the peer-selection policy to additionally allow the selection of neighbors with lower average upload rates. Their system demonstrates appealing performance, as verified in real-world scenarios consisting of 25 nodes. However, the experiments do not take into account factors like client mobility or interactivity.

Considering MANETs, Rodrigues [Rodrigues, 2018] creates the Minimum-Distance Indirect-Reciprocity Algorithm (MDIRA), adhering to the layered-architecture protocol classification. The peer selection deploys several criteria, such as geographical distance, upload rate, and indirect reciprocity, while the piece-selection policy mainly employs a *sliding-window* based approach. The experiments demonstrate an interesting performance when compared to an idealized theoretical baseline. However, it is important to note that the scenarios simulated are limited to 25 nodes and a single model of mobility. In the same study, a variant is proposed, which introduces data dispersion as a criterion in the peer-selection policy. However, this addition proves to be ineffective due to the increased computational overload.

In [Rodrigues and Rocha, 2019], we introduce two approaches for MANETs: the Application layer Interactive BitTorrent Algorithm (AIBA) and the Network-layer Interactive BitTorrent Algorithm (NIBA). The former belongs to the layered-architecture protocol classification, whereas the latter lies within the cross-layer protocol classification. More precisely, in AIBA, the peer-selection policy evaluates the upload and download rates as key criteria. By its turn, NIBA employs a peculiar peer-selection policy: the upload and download rates are computed for neighbors given a specified number of hops in distance (or beyond). In both algorithms, the piece-selection policy is founded on the concept of a *sliding window*. The results suggest a slight superiority for NIBA under a limited observation scope once the experiments are confined to a small set of scenarios.

In [Rodrigues and Rocha, 2021], we further propose another application-layer BitTorrent protocol aiming at MANETs, denoted as Interactive BitTorrent - Application (IB-A). In this proposal, we highlight the inclusion of the indirect-reciprocity criterion in the peer-selection policy, and the concept of the *sliding window* in the piece-selection policy. Both aspects of interactivity and mobility are extensively explored in the experiments. This proposal may be seen as an advancement of our two other proposals mentioned above, namely AIBA and NIBA [Rodrigues and Rocha, 2019].

At last, the authors in [Ghani *et al.*, 2021], introduce the Quality Adaptive Model (QAM) framework. This framework is a fusion of the BitTorrent protocol with the IP Multimedia Subsystem (IMS) plus a dynamic video streaming strategy to enhance video transmission on 5G networks. The IMS is employed for peer authentication and for notifying other peers when there is a change in the IP address, using for that the Service Initiation Protocol (SIP). The BitTorrent protocol serves to grant authenticated peers to participate in sharing a video, encoded with various Scalable Video Coding (SVC) levels. To download a video piece, the peer analyzes the necessary SVC level to improve the transfer. To determine the optimal SVC level, the peer takes various factors into account, including device specifications (e.g., resolution), timing considerations (e.g., frame rate), quality-related aspects (e.g., device capacity), and the peers' capability of providing the pieces. Experiment outcomes indicate that this approach reduces the total download time, even under scenarios with increased loss rates and high traffic congestion.

Despite both QAM and BitCover deploying the 5G technology and the BitTorrent protocol, there are advantages of QAM when compared to BitCover and vice versa. Among the advantages of QAM, we mention that the use of SVC can reduce video discontinuities due to eventual bandwidth constraints. Besides, the use of IMS can reduce latency when downloading pieces from closer neighbors. On the other hand, among the advantages of BitCover, we can mention the use of the concepts of *sliding window* and *interior buffer*, which allow downloading pieces based on their ascending order and availability, hence contributing to a smooth video playout. Furthermore, it is worth noting that the deployment of the piece-coverage criterion may increase the download rate because of likely reusing ongoing connections with neighbors. Finally, BitCover's experiments have considered large files and high upload rates [Rocha and Rodrigues, 2023], leveraging the 5G full capacity, whereas those with QAM have been related to small files and low upload rates, i.e., 10 MB and 10 Mbps, respectively [Ghani *et al.*, 2021].

Considering the studies above, which are compiled in Table 2, besides what was explained in Section 2, we may see that BitCover is a rather sophisticated algorithm that incorporates the key insights from the latest research in interactive VoD streaming over 5G networks. It stands out for its innovative combination of D2D communication, the effective replication strategy of BitTorrent, and the capabilities offered by the 5G technology. From this, the key peculiarity of this present work is that we now especially strive to enhance BitCover's performance by probing into new value setups for its configuration parameters. Apart from this, we notably bear in mind that there are also several works dedicated to streaming over wireless mobile networks that do not rely specifically on BitTorrent. So, for readers interested in these kinds of proposals, we would recommend the works of [Ben Rhaiem *et al.*, 2016; Kavitha and Latha, 2016; Shah *et al.*, 2017; Abbasi and Elbiaze, 2018; Polakovič *et al.*, 2018; Fleury *et al.*, 2019; Kim and Chung, 2020; Shrama *et al.*, 2020; Kumar *et al.*, 2022; Palit *et al.*, 2023; Aparicio *et al.*, 2023] and references therein.

4 Tuning experiments for parameter configurations

This section presents the simulation experiments to determine near-ideal values for the following BitCover's parameters: δ_t, N_p, and y. Recall that these parameters are defined in Table 1. Furthermore, the goal and importance of each of these parameters in the algorithm's operation are explained in Subsections 2.1 and 2.2. The experiments to follow encompass the logical modeling of a VoD streaming system serving n peers that want to watch the same video file, where each peer stands for a system user.

For ease of understanding, this section is divided as follows. In Subsection 4.1, we explain the methodology we used to obtain the near-ideal values, besides mentioning the application domain dealt with in the experiments. Subsection 4.2 discusses peer's mobility and interactivity profiles, respectively. In Subsection 4.3, we present the experiment setup. Finally, simulation results and corresponding analyses constitute Subsection 4.4.

4.1 Methodology and application domain

The experiments are divided into three sets. Each set is devoted to one of the BitCover's parameters for which we want to determine a near-ideal value. To obtain a parameter's near-ideal value, we vary it within a certain range and assess the performance metrics R_D, T_D, O_S, and O_P, which are defined in Table 3. The value that best optimizes most of the performance metrics is then chosen as the near-ideal one.

Table 2. Synthesis of literature works.

References	Brief description
[Sbai *et al.*, 2008; Krifa *et al.*, 2009; Sbai and Barakat, 2009; Salhi *et al.*, 2009]	The authors introduce the BitHoc protocol and several extensions specifically tailored to MANETs. The works do take mobility into account to some extent but do not offer support for interactivity.
[Quental and Gonçalves, 2011]	The authors propose a Mobile-BitTorrent protocol for MANETs. Implementation simplicity is the pivotal advantage. However, the experiments do not take into account either interactivity or mobility.
[Casini *et al.*, 2016; Jenkac *et al.*, 2006; Leung and Chan, 2007]	The authors present investigations regarding broadcast content dissemination. These studies constitute solid baselines for the development of streaming protocols.
[Yang *et al.*, 2017]	Based on BitTorrent, the authors introduce a streaming system named MTV, designed for video sharing on cellular networks. The system leverages both mobile networks and WiFi-Direct communication technology.
[Rethfeldt *et al.*, 2018]	The authors present MeNTor, a cross-layer protocol devoted to WMNs. Neither mobility nor interactivity is though considered in the experiments.
[Rodrigues, 2018]	The authors propose the Minimum-Distance Indirect-Reciprocity Algorithm (MDIRA) which fits into the layered-architecture protocol classification. The experiments indicate an attractive performance when contrasted with a theoretical baseline scheme.
[Rodrigues and Rocha, 2019]	Focusing on MANETS, we propose the Application-layer Interactive BitTorrent Algorithm (AIBA) and the Network-Layer Interactive BitTorrent Algorithm (NIBA). The experiments include both client mobility and interactivity.
[Rodrigues and Rocha, 2021]	We propose another application-layer BitTorrent protocol aiming at MANETs, denoted as Interactive BitTorrent - Application (IB-A). This proposal may be seen as an advancement of our two aforementioned proposals, namely AIBA and NIBA [Rodrigues and Rocha, 2019].
[Ghani *et al.*, 2021]	Focusing on streaming over wireless mobile networks, this work proposes the Quality Adaptive Model (QAM) framework, which employs a fusion of the BitTorrent algorithm, the IP Multimedia Subsystem, and a dynamic video streaming strategy.
[Ben Rhaiem *et al.*, 2016; Kavitha and Latha, 2016; Shah *et al.*, 2017; Abbasi and Elbiaze, 2018; Polakovič *et al.*, 2018; Fleury *et al.*, 2019; Kim and Chung, 2020; Shrama *et al.*, 2020; Kumar *et al.*, 2022; Palit *et al.*, 2023; Aparicio *et al.*, 2023]	These are works that do not use the BitTorrent paradigm for streaming videos over wireless mobile networks. These works are mainly cited herein as references for readers interested in other than BitTorrent-based solutions for streaming services.

Moreover, we add that the experiments refer to streaming scenarios within the online-learning domain, which is one of the most relevant and popular application domains (e.g., [Tseng *et al.*, 2023; Ng *et al.*, 2023; Comsa *et al.*, 2023]).

4.2 Mobility and interactivity

Steady-state analysis is conducted within a square cell site spanning 400 meters on each side. In this cell site, n peers move freely without encountering obstacles. These peers follow a SMOOTH mobility model [Munjal *et al.*, 2011], whose peers' initial positions and subsequent movement traces are generated by the Bonnmotion tool [Aschenbruck *et al.*, 2010]. Handoffs, representing connection transfers between different mobile cells, are not taken into account in this analysis. It is assumed that the value of n remains constant: new peers replace departing ones, thereby ensuring a consistent network size within the cell site.

The SMOOTH model replicates statistical characteristics observed in real human mobility patterns, including: the distance a peer travels to visit a popular or the nearest venue, which conforms to an inverse power-law distribution characterized by the parameter α; the duration a peer spends waiting in a venue, which adheres to a truncated power-law distribution with the parameter β; and the phenomenon of community formation, where groups of peers form clusters to emulate social interactions and human behavior.

Table 4 presents the main setup values for the SMOOTH model. These values are used for the State Fair scenario (defined in [Munjal *et al.*, 2011]) because of their resemblance to both a real-world square area of a 5G cell site and the distance-learning class simulated in this work. In these settings, individuals typically gather round, occupying the same physical space.

Regarding the interactivity profile of a peer, we utilize a behavior model influenced by the research of [Abram-Profeta and Shin, 1998; Rodrigues and Rocha, 2021]. The interactive actions include *Play*, *Pause*, *Jump Backwards* (J_B), and *Jump Forwards* (J_F). The *Play* action indicates that the user is viewing the video. The *Pause* action means that the playback has been temporarily halted. The J_B and J_F ac-

Table 3. Performance metrics used for evaluation

Metric	Notation	Description
Download Rate	R_D	It estimates the peer's average rate (in kB/s) to receive video pieces that may be visualized by it. The greater this value is, the better the corresponding proposal is likely to be. This metric is an adequate estimate to evaluate system QoS.
Discontinuity Time	T_D	It estimates the peer's total average interruption time (in seconds) during the video playback. The smaller this value is, the better the corresponding proposal is likely to be. This metric is an adequate estimate to evaluate QoE.
Initial Seed Overloading	O_S	It estimates the percentage of piece requests (over the total piece requests of all peers) that are handled by the initial seed. The smaller this value is, the better the corresponding proposal is likely to be, since data traffic tends to be more uniformly distributed among mobile nodes.
Protocol Overhead	O_P	It lies within the interval of 0.0 to 1.0. It is the ratio of the video size in kilobytes to the total kilobytes transferred by the peer (considering all protocol messages). The closer O_P is to 0.0 (1.0), the more (less) overhead is generated by the protocol. In other words, the closer O_P is to 1.0, the better the corresponding proposal is likely to be, since less overhead exists in the system.

Table 4. SMOOTH setup values.

Parameter	Value	Description
$clusters$	4	The quantity of clusters in the region
α	3	When $\alpha \to 0$, the destination to explore is chosen at random. When $\alpha \to \infty$, the destination to explore is the nearest.
β	1	When $\beta \to 0$, peers linger for extended periods at a small number of locations. When $\beta \to \infty$, peers linger for a short period at a small number of locations.

tions represent skipping to a point in the video that is ahead of or behind the current playback point, respectively.

Additionally, *Play*, *Pause*, J_B, and J_F are activated in accordance with a Poisson distribution of rate λ, and with probabilities p_{pl}, p_{ps}, p_{bw}, and p_{fw}, respectively. *Play*, *Pause*, J_B, and J_F own a common length S, computed as a percentage of the video file size, f_{size}. Table 5 lists the parameters and values of the three interactivity profiles utilized in the simulations, which are the same as used in [Rodrigues and Rocha, 2021].

4.3 Experiment setup

The experiments are conducted within the PeerSim [Montresor and Jelasity, 2009] simulation environment, using a hardware platform consisting of an Intel Core i7 (2.6 GHz) processor, 24 GB of RAM, and executing on a Linux operating system.

The four performance metrics evaluated in the experiments (namely R_D, T_D, O_S, and O_P) may together yield evidence of system QoS and client QoE. The outcomes exhibit 95% confidence intervals that fall within 5% of the reported average values, based on a total of 30 simulation executions.

To assess quality, we stipulate that R_D should match or exceed the video encoding rate (i.e., it is assumed to be 20 Mbps as detailed further) to grant a satisfactory QoS [Wang *et al.*, 2008]. Regarding T_D, we set an upper limit of 5 ms of latency for each viewed piece to ensure a satisfactory QoE [Soldani and Manzalini, 2015; Bhamidipati and Kilari, 2010], resulting in a total interruption time of nearly 10 seconds. As for O_S, we consider values under 3.4% as indicative of satisfactory performance in P2P VoD server-assisted scenarios [Gkortsilas *et al.*, 2012; Braun *et al.*, 2019]. For O_P, we assume values above 0.39 [Legout *et al.*, 2005]. For ease of reference, Table 6 recaps all these numerical values.

When establishing a WiFi-Direct connection, we utilize the simulator defined in [Baresi *et al.*, 2016]. The WiFi-Direct simulator was integrated into PeerSim, serving as a tool to retrieve the WiFi connection parameters and values mentioned below. This simulator defines various parameters for delays, as detailed in Table 7 [Baresi *et al.*, 2016]. All of these parameters are assigned constant values, with the exception of the *channelDelay*, which varies within the range of 0-400 ms based on the physical distance [Schoonwinkel, 2016; Mbala *et al.*, 2021]. As for losses, the simulator also takes into account the physical distance and employs Friis's path loss formula to calculate the signal strength between two endpoints. Hence, losses and delays mainly stem from fac-

Table 5. Configuration values for interactivity profiles.

Parameter	Low	Medium	High
λ	0.005/s	0.014/s	0.025/s
S	14.5% of f_{size}	3.5% of f_{size}	1.5% of f_{size}
p_{pl}; p_{ps}	0.89; 0.01	0.71; 0.05	0.55; 0.15
p_{fw}; p_{bw}	0.05; 0.05	0.12; 0.12	0.15; 0.15
f_{size}	10193 MB	10193 MB	10193 MB

tors such as physical distance, signal strength, and packet size [Baresi *et al*., 2016]. In the case of the 5G connection, we only simulate delay and losses. For the delays, we randomly select them from a range of 5 to 20 ms, along with an average packet-loss rate of 4% [Xu *et al*., 2020; Narayanan *et al*., 2020]. These values, defined at the application layer, are assumed to encompass all the losses and delays that occur in the transport and lower layers [Xu *et al*., 2020; Narayanan *et al*., 2020].

At last, the video is encoded at a rate of 20 Mbps, which aligns with the minimum bitrate required for transmitting 4k-resolution content at 60 fps (frames per second), in accordance with the recommendations from the YouTube system [YouTube Help, 2021]. This video corresponds to a long lecture spanning approximately 70 minutes (4175 seconds), resulting in a total file size of 10193 MB. We adopt a video piece size of 4 MB, as proposed by the works of [**?**Jones, Ben, 2021]. Otherwise stated, the configuration setup for BitCover's parameters in the experiments to follow may be seen in Table 1.

4.4 Simulation results and analyses

The figures in this subsection depict the outcomes of the simulation experiments. The Y-axis corresponds to the metric being analyzed. The X-axis has 15 bars put into three groups. The first group relates to the low-interactivity profile, the second group to the medium-interactivity profile, and the last group to the high-interactivity profile. Furthermore, each bar within each group stands for the network size (i.e., the number of peers the network has). More precisely, from left to right, we have 100, 200, 300, 400, and 500 peers, respectively.

We recall that the overall methodology to obtain the near-ideal values for each parameter is to vary it within a certain reasonable range to find out the one that best optimizes the majority of the performance metrics. Besides, to help with our intricate analysis which spans correlated variables, we look into two separate views of potential impact on the determination of each parameter's near-ideal value, namely network size and interactivity profile.

Table 6. Baseline values.

Metric	Baseline
R_D	above 20 Mbps
T_D	below 10 s
O_S	below 3.4%
O_P	above 0.39

4.4.1 Parameter δ_t

For δ_t varying from 8 to 160 seconds, Figures 3, 4, 5, and 6 depict the results computed for R_D, T_D, O_S, and O_P, respectively. For instance, we may easily note in Figure 3 that: (i) the larger the network size is, the larger the value of R_D tends to be; (ii) the interactivity profile does not impact on R_D; and (iii) the value $\delta_t = 10$ s optimizes the metric R_D, independently of both network size and interactivity profile.

To summarize, Table 8 brings the main observations achieved in the experiments for δ_t. From this, the general conclusion is that δt should be set to 10 seconds once this value may be used to optimize three performance metrics. The only exception resides in O_P because fewer messages are sent as δ_t increases. Moreover, notice that $\delta t = 10$ s is also the default value used for BitCover (original), as indicated in Table 1.

4.4.2 Parameter N_p

For N_p varying from 60 to 160 peers, Figures 7, 8, 9, and 10 then depict the numerical results computed for R_D, T_D, O_S, and O_P, respectively. For instance, we may note in Figure 7 that: (i) the network size generally does not impact on R_D; (ii) the more interactivity, the larger the value of R_D; and (iii) the value $N_p = 80$ peers maximizes the metric R_D, independently of both network size and interactivity profile.

Concisely, Table 9 shows the main observations achieved in the experiments for N_p. From this, the general conclusion is that N_p should be set to 80 peers since this is the value that best optimizes most of the performance metrics. As seen in the metric δ_t, the only exception resides in O_P. The rationale lies in the fact that more messages are sent as N_p increases. At last, $N_p = 80$ peers (neighbors) is also the default value used for BitCover (original), as it may be seen in Table 1.

4.4.3 Parameter y

For y varying from 3 to 15 slots, Figures 11, 12, 13, and 14 then depict the results computed for R_D, T_D, O_S, and O_P, respectively. For instance, we may note in Figure 11 that: (i) the larger the network size is, the larger the value of R_D tends to be; (ii) the more interactivity, the larger the value of R_D; and (iii) the value $y = 9$ slots maximizes the metric R_D, independently of both network size and interactivity profile.

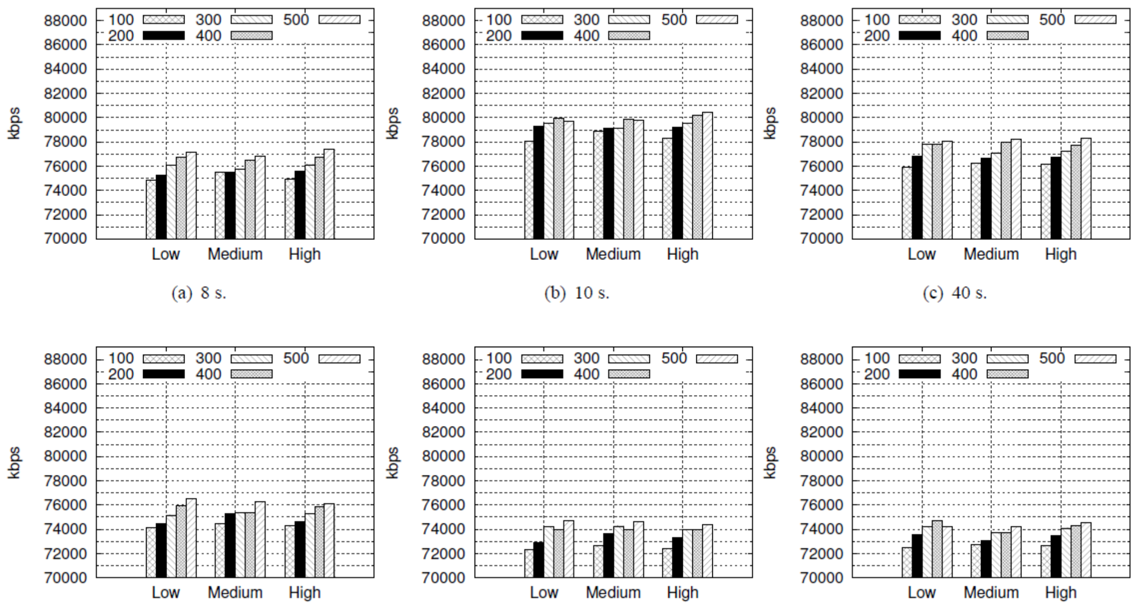
In a nutshell, Table 10 presents the main observations achieved in the experiments for y. From this, the general conclusion is that y should be set to nine slots. This value is beyond twice the default value used for BitCover (original). Even though there is a tie, the metrics R_D and T_D may be fairly judged as more important than O_S and O_P. The

Table 7. WiFi-Direct setup delays.

Parameter	Duration (ms)	Description
$channelDelay$	400	Duration needed for sending a segment at MAC layer.
$authenticationDelay$	100	Duration required to authenticate an invitation.
$powerDelay$	100	Duration when the peer goes into a sleep mode.
$switchingDelay$	500	Duration required to discover a channel.

Table 8. Analysis of δ_t - varying from 8 to 160 seconds.

Metric	Reference figures	Network size	Interactivity	Near-ideal δ_t
R_D	Figure 3	The larger the better	Same	**10 s**
T_D	Figure 4	The larger the better	Same	**10 s**
O_S	Figure 5	The larger the better	Same	Any
O_P	Figure 6	The larger the worse	The more the worse	160 s



(a) 8 s. (b) 10 s. (c) 40 s.

(d) 80 s. (e) 120 s. (f) 160 s.

Figure 3. R_D for the analysis of δ_t.

Table 9. Analysis of N_p - varying from 60 peers to 160 peers.

Metric	Reference figures	Network size	Interactivity	Near-ideal N_p
R_D	Figure 7	Same	The more the better	**80**
T_D	Figure 8	Same	The more the better	**80**
O_S	Figure 9	The larger the better	Same	60 or **80**
O_P	Figure 10	The larger the worse	The more the worse	60

supportive reason is that, by definition, the two first metrics notoriously play much more decisive roles in final system performance due to being directly related to QoS and QoE, respectively.

5 Competitive analysis

Let BitCover-OPT denote the original BitCover proposal configured with the near-ideal values just determined in the previous section, namely $\delta_t = 10$ s, N_p= 80 peers (neighbors), and $y = 9$ slots. The goal of this section is to competitively compare BitCover-OPT against the original BitCover, IB-A, and MTV proposals through simulation experiments. For the sake of proper organization, we opt to divide this section into two subsections as follows. Subsection 5.1 briefly reviews the crucial operational features and design of both IB-A and MTV proposals, and Subsection 5.2 presents and analyzes the results of the simulation experiments.

5.1 Literature proposals: IB-A and MTV

IB-A and MTV are BitTorrent-based proposals for interactive streaming over wireless mobile networks. The former

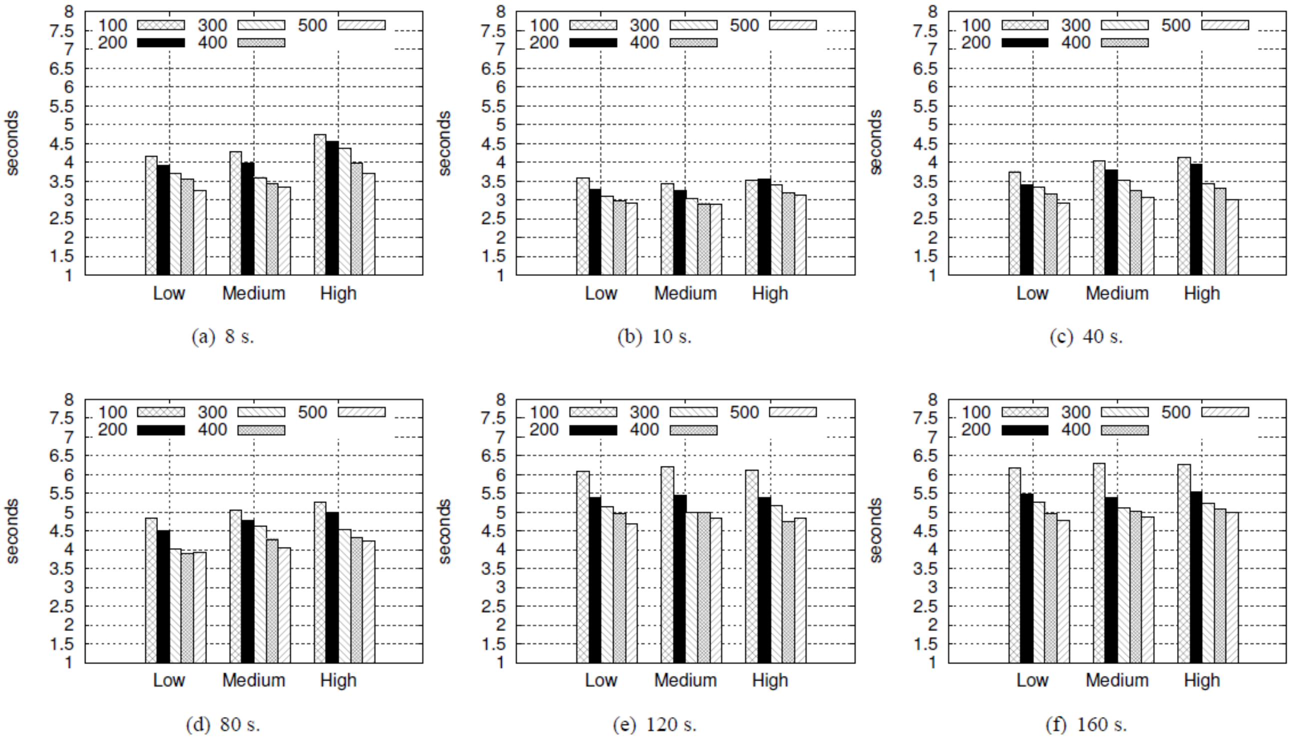


Figure 4. T_D for the analysis of δ_t.

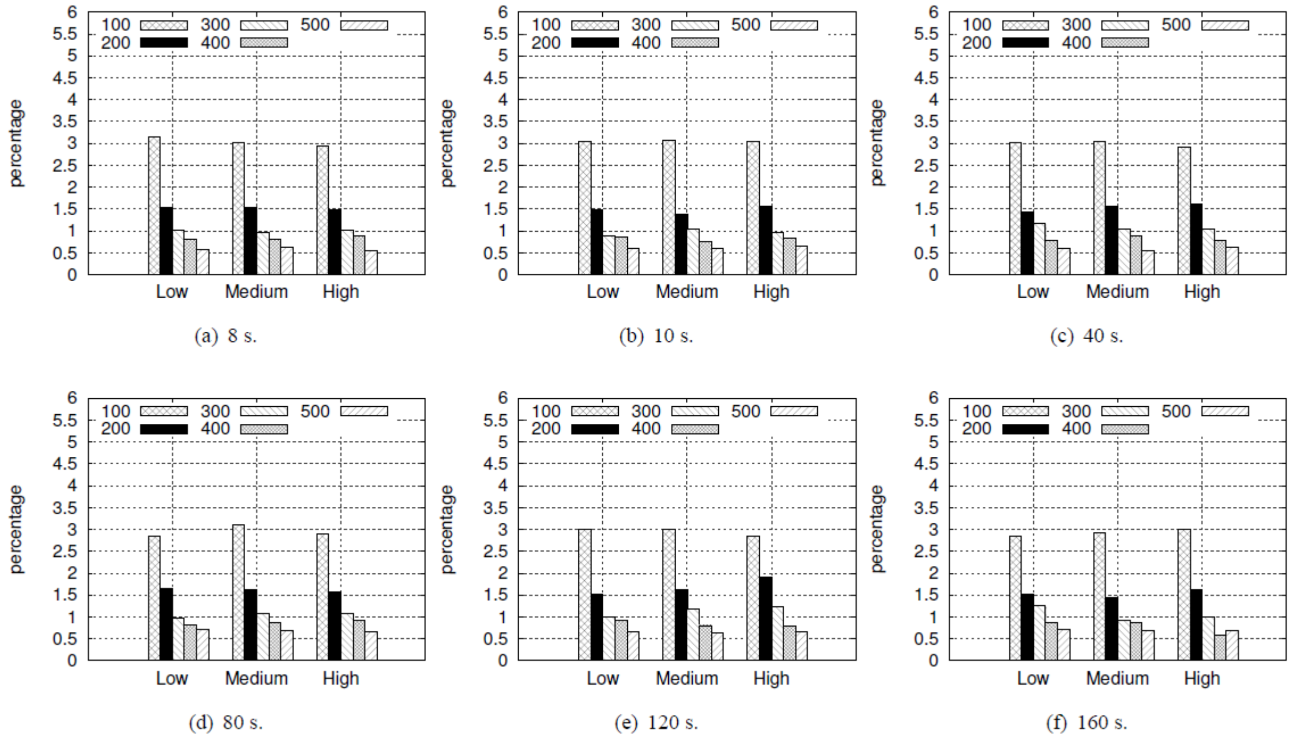
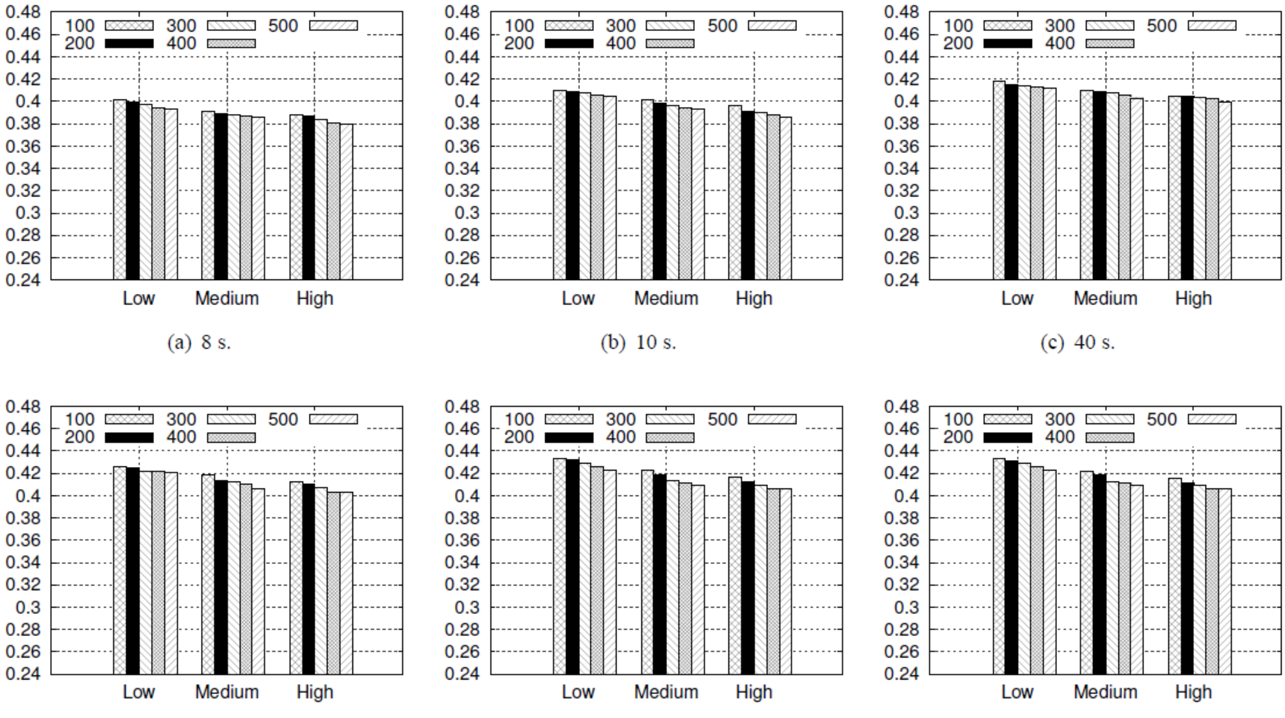


Figure 5. O_S for the analysis of δ_t.

Table 10. Analysis of y - varying from 3 slots to 15 slots.

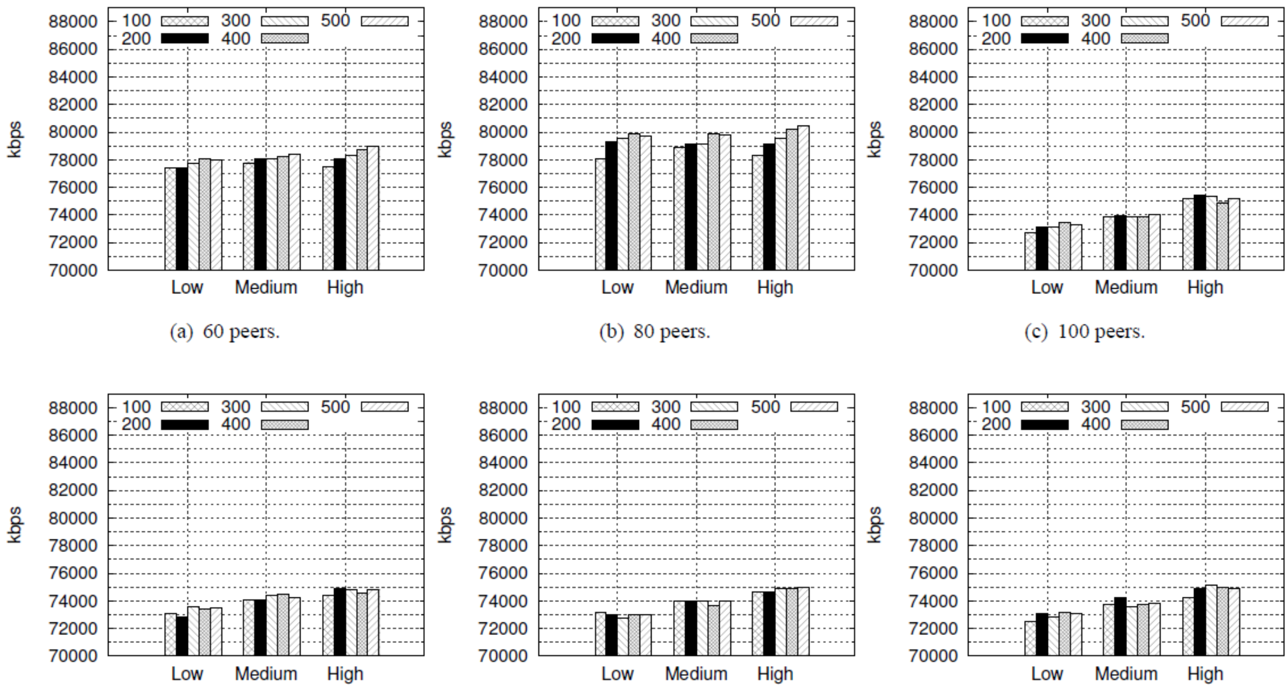
Metric	Reference figures	Network size	Interactivity	Near-ideal y
R_D	Figure 11	The larger the better	The more the better	**9**
T_D	Figure 12	The larger the better	The more the better	**9**
O_S	Figure 13	The larger the better	Same	3
O_P	Figure 14	The larger the worse	The more the worse	3



(a) 8 s. (b) 10 s. (c) 40 s.

(d) 80 s. (e) 120 s. (f) 160 s.

Figure 6. O_P for the analysis of δ_t.



(a) 60 peers. (b) 80 peers. (c) 100 peers.

(d) 120 peers. (e) 140 peers. (f) 160 peers.

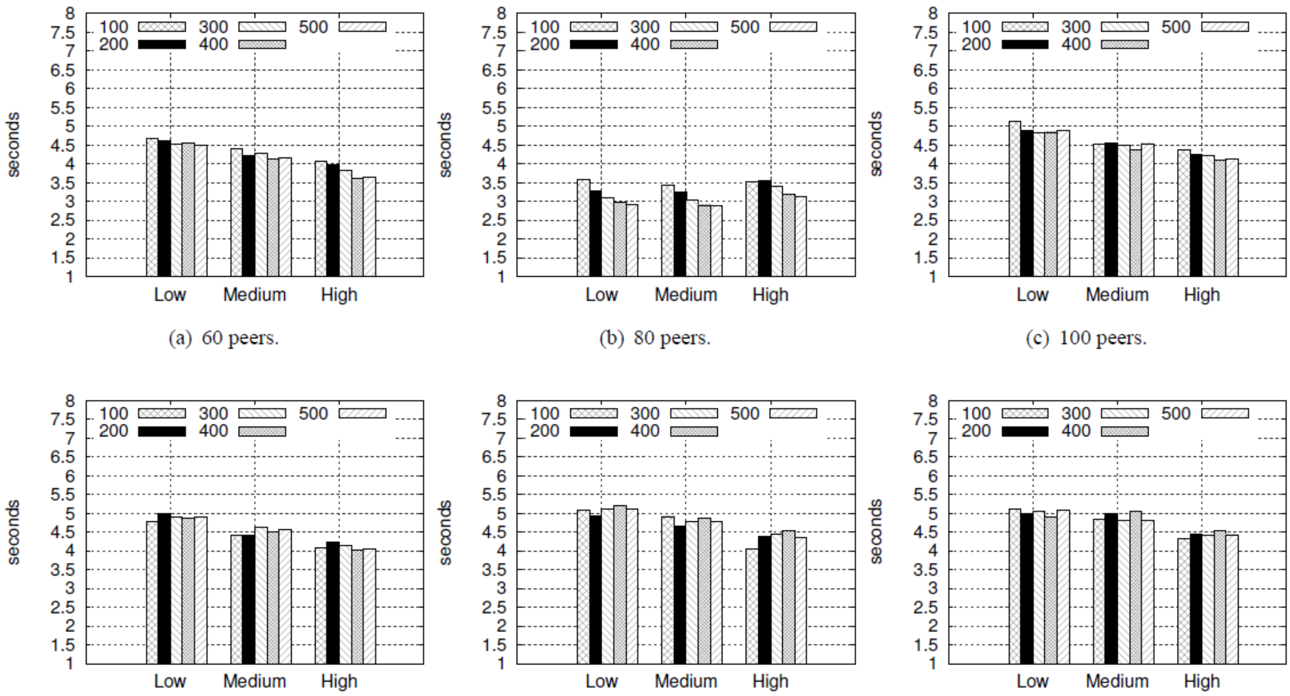
Figure 7. R_D for the analysis of N_P.

is one of the most recent BitTorrent adaptation that targets MANETs, whereas the latter deploys WiFi-Direct technology and focuses on cellular networks.

Even though IB-A and MTV are not designed with an original focus on 5G-technology architectures, we conjecture they serve our goal of competitive comparison with BitCover since they are recent efficient BitTorrent-based proposals deployed on wireless mobile communications. See that IB-A and MTV are discussed in Section 3 and, for ease of reference, they are now summarized in Table 11, mainly to outline their corresponding operational features and design in the experiments to follow.

5.2 Simulation experiments

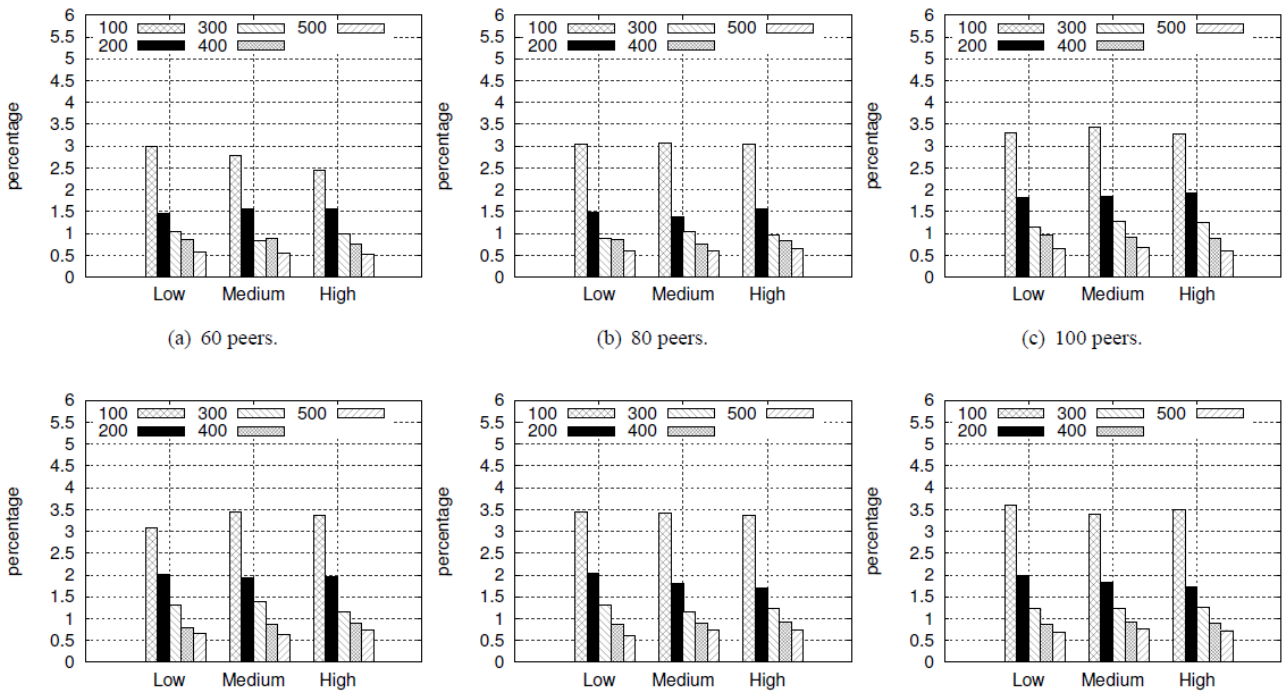
We explain that, based on the figures to appear in the following, we first carry out a qualitative competitive analysis in which the main goal is to visually settle a qualitative perfor-



(a) 60 peers. (b) 80 peers. (c) 100 peers.

(d) 120 peers. (e) 140 peers. (f) 160 peers.

Figure 8. T_D for the analysis of N_P.



(a) 60 peers. (b) 80 peers. (c) 100 peers.

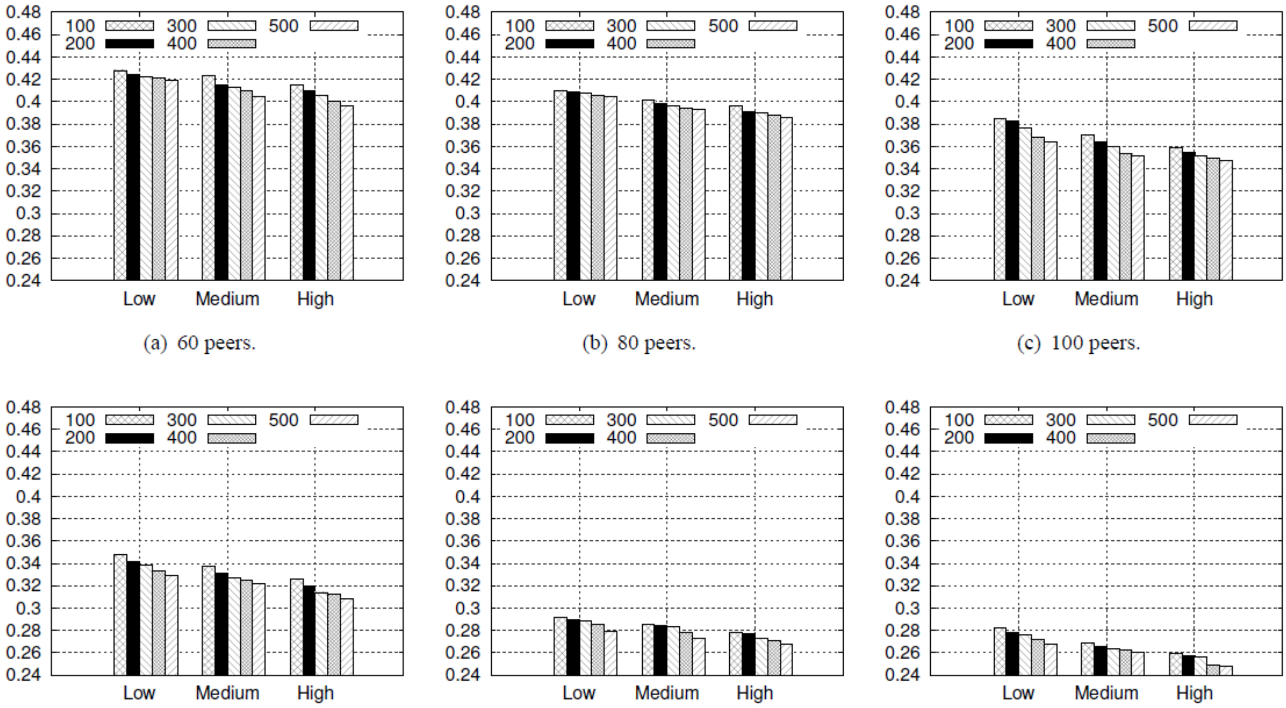
(d) 120 peers. (e) 140 peers. (f) 160 peers.

Figure 9. O_S for the analysis of N_P.

mance rank among the four proposals (i.e., BitCover original, BitCover-OPT, IB-A, and MTV), spanning the three interactivity profiles for each of the four previously defined metrics. Subsequently, we then proceed to a quantitative competitive analysis in which the main goal is to numerically compute the optimization achieved by BitCover-OPT (compared to the other proposals) as well elucidate the corresponding rationale.

So, consider initially Figure 15. It brings the outcomes for the R_D metric. From this figure, we have the two following overall observations. First, all proposals achieve R_D above the 20 Mbps video-encoding rate baseline. Second, BitCover-OPT performs the best among all four proposals, independently of network size and interactivity profile, followed by BitCover (original), IB-A, and MTV, respectively.

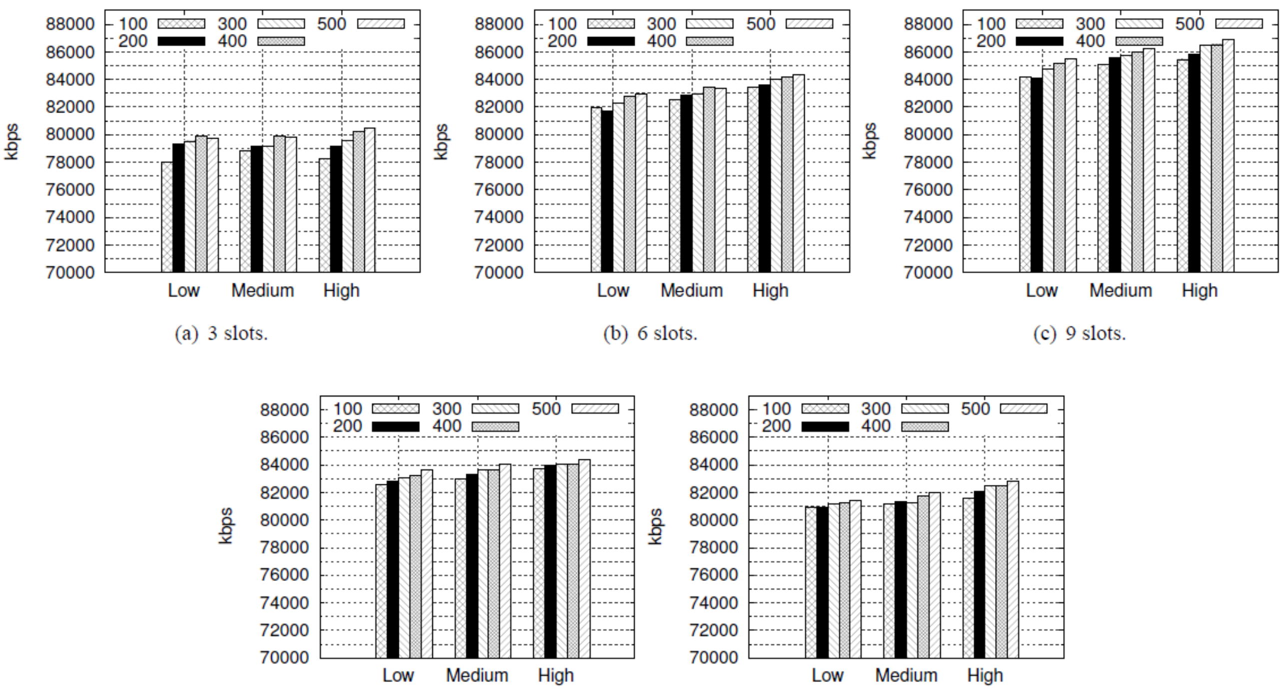
Now see Figure 16. It owns the outcomes for the T_D metric. From this figure, we highlight the two following main observations. First, BitCover (original and optimized ver-



(a) 60 peers. (b) 80 peers. (c) 100 peers.

(d) 120 peers. (e) 140 peers. (f) 160 peers.

Figure 10. O_P for the analysis of N_P.



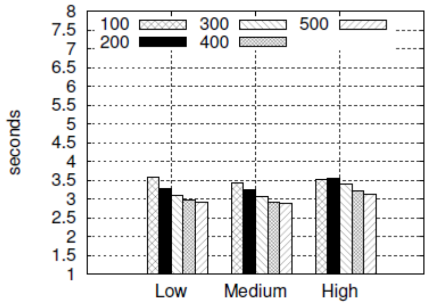
(a) 3 slots. (b) 6 slots. (c) 9 slots.

(d) 12 slots. (e) 15 slots.

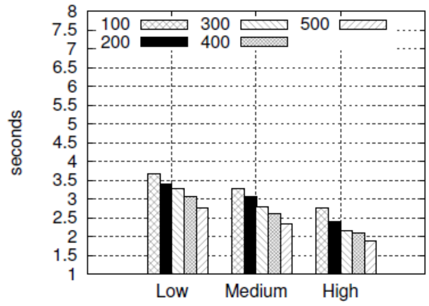
Figure 11. R_D for the analysis of y.

sions) are the only ones to own T_D below the 10-second interruption-time baseline (see Table 6). Second, BitCover-OPT performs the best, no matter the network size and interactivity profile, followed by BitCover (original), IB-A, and MTV, respectively.

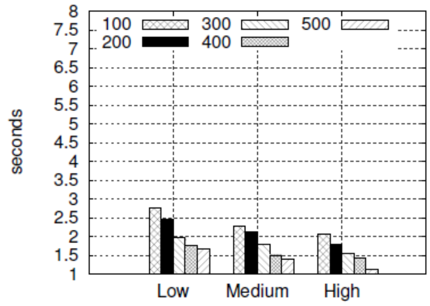
At this time see Figure 17. It shows the outcomes for the O_S metric. To better visualize the differences among the four proposals, we deliberately cut the Y-axis on the 6.0% value once MTV responds to $\approx$ 80% of the total piece requests. From this figure, we may get to the two following general observations. First, BitCover (original) and IB-A are the only ones to achieve O_S below the experiment baseline of 3.4%, i.e., these are the two best proposals in terms of the O_S metric. Second, BitCover (original) performs best among all proposals, followed by IB-A, BitCover-OPT, and MTV, respectively.



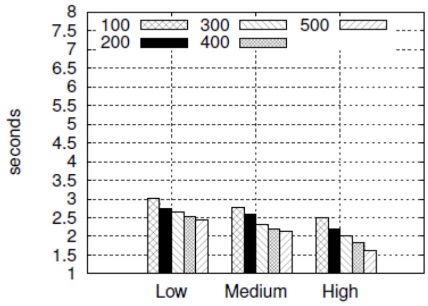
(a) 3 slots.



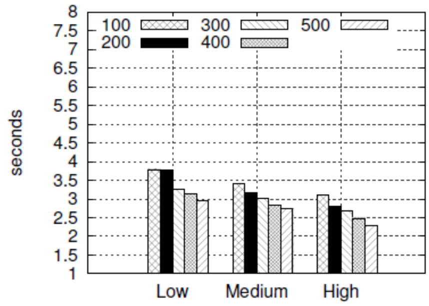
(b) 6 slots.



(c) 9 slots.

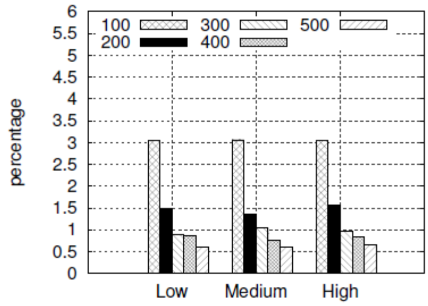


(d) 12 slots.

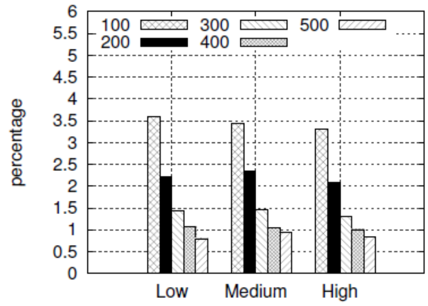


(e) 15 slots.

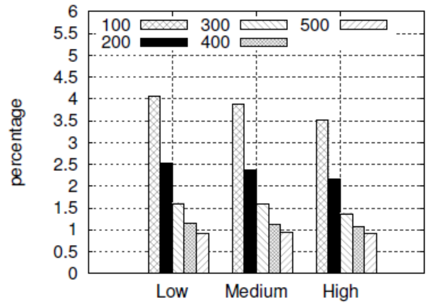
Figure 12. T_D for the analysis of y.



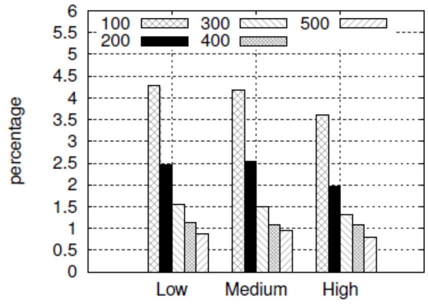
(a) 3 slots.



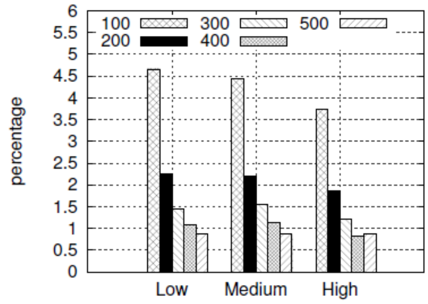
(b) 6 slots.



(c) 9 slots.



(d) 12 slots.



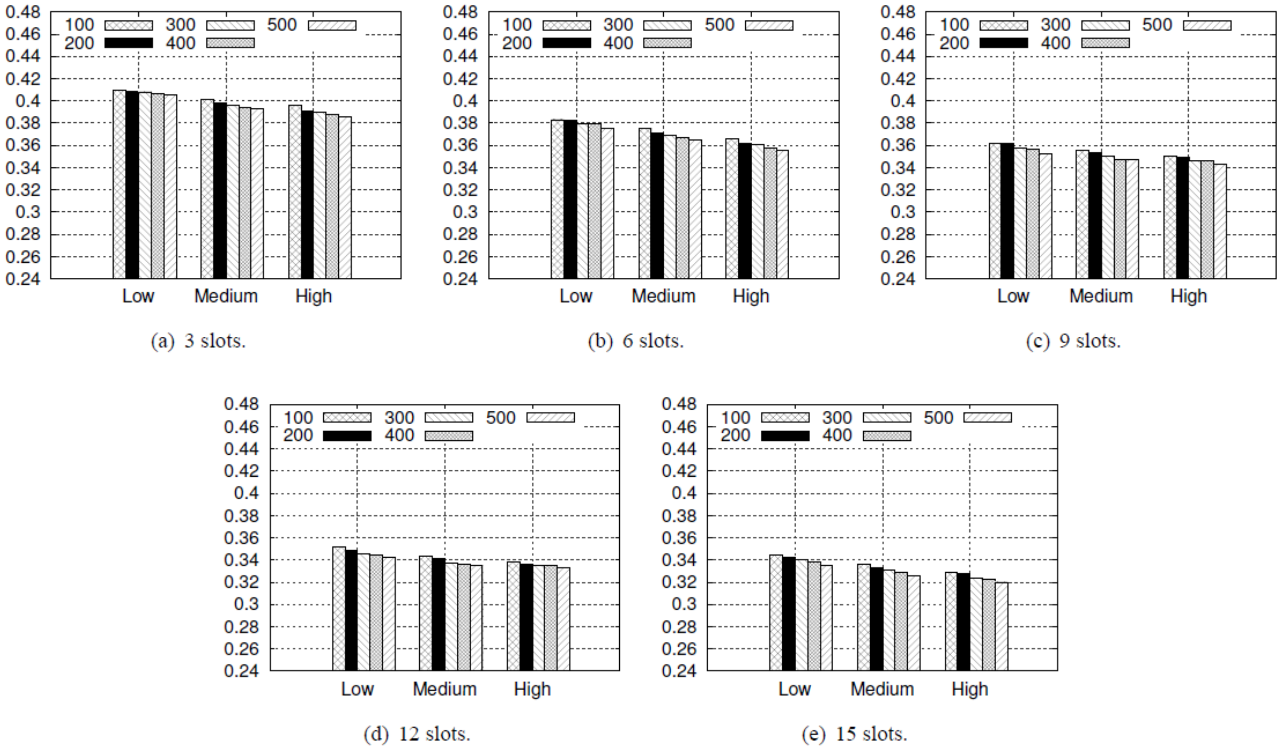
(e) 15 slots.

Figure 13. O_S for the analysis of y.

Now see Figure 18, in which we may note the outcomes for the O_P metric. From this figure, we can outline the following observations. First, the original BitCover and IB-A are the only ones that achieve O_P above the experiment baseline of 0.39, i.e., they are the two best proposals in terms of the O_P metric, differing very little from each other. Second, BitCover-OPT appears in third place, and MTV comes in the last place.

Finally, for the quantitative competitive evaluation, we mount Table 12. The values therein computed stand for average optimizations achieved by the BitCover-OPT with respect to BitCover (original), IB-A, and MTV, taking over the three interactivity profiles. We clarify that positive numbers refer to increases (gains), whereas negative numbers refer to decreases (reductions). From this table along with the observations obtained from the figures just discussed above, we may get to the following comparative concluding remarks.

i) Compared to BitCover (original)



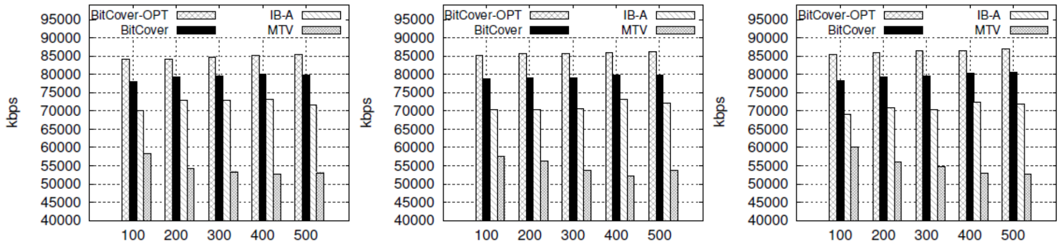
(a) 3 slots. (b) 6 slots. (c) 9 slots. (d) 12 slots. (e) 15 slots.

Figure 14. O_P for the analysis of y.

Table 11. Main operational features and design.

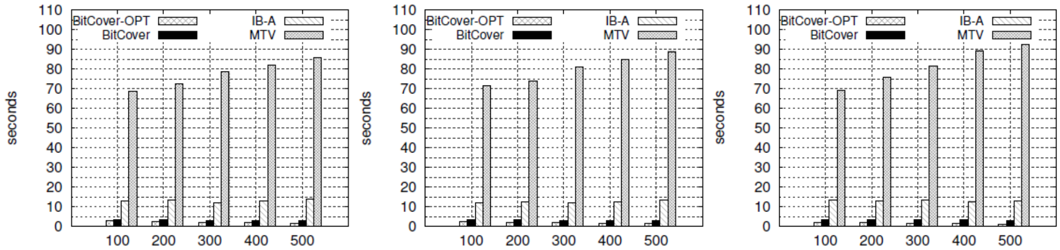
Proposals	Description
IB-A [Rodrigues and Rocha, 2021]	• The focus is on MANETs, and it utilizes δt =100 s to ensure the same peers' connections in an extremely dynamic environment. • The peer-selection approach it employs considers *upload rate* and *indirect reciprocity*. Priority is given to peers which, on average, share a greater number of pieces. • The piece-selection approach incorporates a *sliding window* with an *interior buffer*, similar to what is used in BitCover. However, the video file is not divided into partitions. • The peers positioned between route endpoints retain pieces that pass through them. • For an adequate and fair competitive analysis in the experiments, we model the Ad-Hoc routing delay/loss as the BitCover's 5G delay/loss, assuming therein included all other delays and losses of below-application layers (please see Subsection 4.3).
MTV [Yang *et al.*, 2017]	• The system relies on a video server to transmit the video via cellular links to peers that are unable to obtain it from their neighbors through WiFi-Direct connections. • The peer selection gives preference to peers with a higher number of downloaded pieces. • The piece selection involves dividing the video into predefined partitions and, subsequently, choosing any piece from a random partition. • For an adequate and fair competitive analysis in the experiments, we model its cellular delay/loss as the BitCover's 5G delay/loss, assuming therein included all other delays and losses of below-application layers (please see Subsection 4.3).

BitCover-OPT is superior. It increases the value of R_D (up to 16.7%) as well as reduces the value of T_D (up to 50.1%). As a result, we achieve better levels of both system QoS and QoE, respectively.



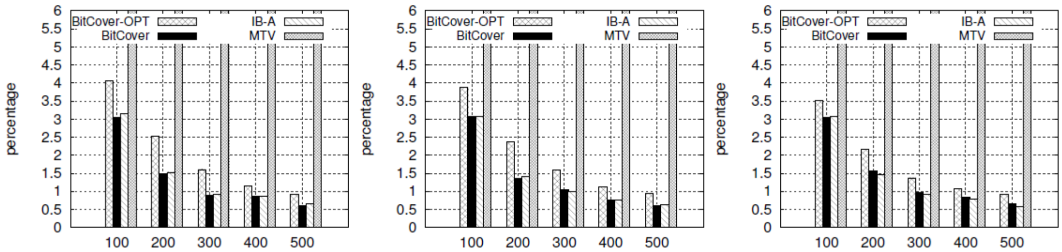
(a) Low. (b) Medium. (c) High.

Figure 15. Performance in terms of R_D.



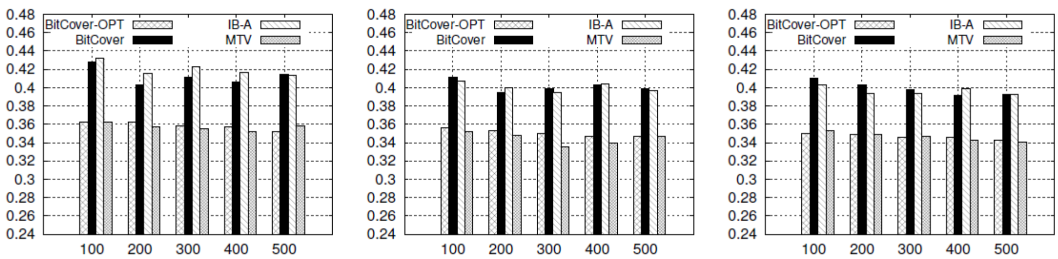
(a) Low. (b) Medium. (c) High.

Figure 16. Performance in terms of T_D.



(a) Low. (b) Medium. (c) High.

Figure 17. Performance in terms of O_S.



(a) Low. (b) Medium. (c) High.

Figure 18. Performance in terms of O_P.

The aforementioned superiority mostly reveals the importance of adequately tuning the configuration parameters, more especially increasing the number of data slots (i.e., setting y to nine slots). We highlight that slot increments allow BitCover-OPT to transfer more pieces, helping it to take advantage of the 5G bandwidth speed, thereby improving its performance. Still, even though the two other remaining metrics (i.e., O_S and O_P) are not improved, they end up being not that decisive to jeopardize performance due to the 5G technological infrastructure's native data transmission and processing capacities.

ii) Compared to IB-A

BitCover-OPT is superior too. It is able to increase the value of R_D (up to 46.9%) as well as reduce the value of T_D (up to 88.2%), resulting in better levels of both system QoS and QoE, respectively.

The above superiority mainly comes as a result of the implementation of the piece-coverage criterion. It optimizes the utilization of the communication channels while also preventing the inefficient use of time in handling numerous communication connections among peers. As also outlined in the comparison with BitCover (original), the other two remaining metrics (i.e., O_S and O_P) are not improved herein either. This is due to the same reasons previously mentioned above.

We shall note that large values of δ_t are important for BitTorrent-like protocols to guarantee long-life endpoints' connections when operating in little dynamic scenarios (like in MANETs), whereas large values of δ_t negatively impact their performance in more dynamic scenarios (like in cellular networks). The rationale mainly lies in the fact that higher-upload-capacity peers are likely to get into the system more frequently. Hence, these protocols had better trigger the so-called unchocking process within narrower time intervals (see Subsection 2.1).

iii) Compared to MTV

BitCover-OPT is also superior. The value of R_D is significantly increased (up to 264.7%), and the value of T_D is decreased (up to 98.5%). This implies that the levels of both system QoS and QoE, respectively, are notably improved.

BitCover's superiority in this case mainly comes as a result of its piece-retrieving policy, which follows a sequential piece-request order. We recall that MTV's piece-retrieving is randomly executed, which may prevent a satisfactory smooth video playout and, hence, yields low levels of QoS and QoE, respectively.

Another supportive reason for BitCover-OPT's superior performance resides in MTV's overloaded streaming server: it is unable to cope with the substantial demand of requests coming directly through the 5G network (nearly 80% of the total requests, as assessed in [Rocha and Rodrigues, 2023]). In BitCover-OPT, these requests are distributed among multiple peers. At last, unlike what was perceived in the comparisons against the two other proposals, BitCover-OPT now beats its competitor (i.e., MTV) considering all metrics, except in the case of the O_P metric whose corresponding outcomes remain almost the same for both algorithms.

6 Conclusions and future works

This paper had the goal of enhancing the BitCover algorithm's performance, which consists of a BitTorrent adaptation for VoD streaming over 5G cellular networks. Overall, BitCover mainly facilitates the exchange of video pieces through both 5G and WiFi-Direct communication channels while introducing a novel peer-selection policy. This policy can enhance channel utilization by giving preference to peers with a larger number of downloaded pieces stored in their local buffers.

To achieve our goal we carried out extensive simulations to determine near-ideal values for the parameter configuration of BitCover. More precisely, we varied its main configuration parameters within a certain reasonable range, while assessing four pivotal performance metrics. The value that best optimized most of the performance metrics was then chosen as the near-ideal one. The experiments were conducted through simulations in various streaming scenarios, encompassing different interactive profiles, network sizes, and video sizes. Additionally, we competitively contrast the optimized version of BitCover with other modern approaches, namely the IB-A and the MTV literature proposals.

Among the major findings, we may highlight that BitCover's original performance is enhanced at about 16.7% in terms of download rate, and at 50.1% in terms of discontinuity time. Additionally, we had optimizations of around 46.9% (download rate) and 88.2% (discontinuity time) compared to IB-A, and 264.7% (download rate) and 98.5% (discontinuity time) compared to MTV. The overall effectiveness of the optimized version of BitCover is thus confirmed by these results, implying that it owns the potential to significantly influence the optimization of bandwidth utilization within a 5G cell site. In other words, BitCover does contribute to unlocking the full capabilities of the 5G technology in delivering video streaming with the desired system QoS and QoE for end-users.

At last, we have in mind the following future directions for this work. First, to propose a generic parameter-configuration model for the BitCover algorithm aiming at its performance enhancement in other domains than online learning, thereby broadening the scope and applicability of our research. Second, to incorporate other competing traffic (e.g., voice and data) in our experiments to analyze the system congestion, as well as to improve the loss-channel model (e.g., using Rayleigh loss). Third, to investigate and include the process of peer handoff into our model, especially due to the dense installation of base stations and the intricate interactions between them and end-to-end network slices in Heterogeneous Networks (HetNets) [Wu *et al.*, 2022; Ramadan *et al.*, 2021]. And fourth, to delve into strategies based on mobile edge computing and caching techniques to combine them with BitCover, including the proposal of parameterized models for multimedia-content placement to alleviate data traffic within the backhaul infrastructure of the radio access network [Chen *et al.*, 2022; Pana *et al.*, 2022; Li *et al.*, 2022; Lin *et al.*, 2023].

Table 12. Optimizations achieved by BitCover-OPT.

BitCover (Original)		IB-A		MTV		
Min	Max	Min	Max	Min	Max	Metric
+11.4%	**+16.7%**	+33.3%	**+46.9%**	+144.1%	**+264.7%**	R_D
–25.1%	**–50.1%**	–78.9%	**–88.2%**	–96.2%	**–98.5%**	T_D
+40.1%	+75.6%	+30.2%	+50.1%	–95.1%	–98.9%	O_S
–9.7%	–16.34%	–13.1%	–17.1%	≈ 0%	≈ 0%	O_P

Declarations

Authors' Contributions

All authors equally contributed to the writing of this article, read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

References

Abbasi, U. and Elbiaze, H. (2018). Multimedia streaming using D2D in 5G ultra dense networks. In *15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–6. DOI: 10.1109/CCNC.2018.8319230.

Abram-Profeta, E. L. and Shin, K. G. (1998). Providing unrestricted VCR functions in multicast video-on-demand servers. In *Proceedings. IEEE International Conference on Multimedia Computing and Systems*, pages 66–75. DOI: 10.1109/MMCS.1998.693626.

Aparicio, J. A. O., Polo, D. C., Carmona-Murillo, J., and Chidean, M. I. (2023). Bitrate analysis in 5g networks for video streaming services using l-moment ratio diagrams. In *IEEE EUROCON 2023 - 20th International Conference on Smart Technologies*, pages 440–444. DOI: 10.1109/EUROCON56442.2023.10198966.

Aschenbruck, N. *et al*. (2010). BonnMotion: A Mobility Scenario Generation and Analysis Tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, pages 51:1–51:10, Brussels, Belgium. DOI: 10.4108/ICST.SIMUTOOLS2010.8684.

Baresi, L., Derakhshan, N., and Guinea, S. (2016). WiDiSi: A Wi-Fi direct simulator. In *IEEE Wireless Communications and Networking Conference*, Doha, Qatar. DOI: 10.1109/WCNC.2016.7565169.

Baresi, L., Derakhshan, N., and Guinea, S. (2016). WiDiSi: A Wi-Fi direct simulator. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–7. DOI: 10.1109/WCNC.2016.7565169.

Ben Rhaiem, O., Chaari Fourati, L., and Ajib, W. (2016). Network coding-based approach for efficient video streaming over MANET. *Computer Networks*, 103:84–100. DOI: 10.1016/j.comnet.2016.04.002.

Bhamidipati, V. D. and Kilari, S. (2010). Effect of delay/delay variation on QoE in video streaming. Master's thesis, Blekinge Institute of Technology. Available at: `https://bth.diva-portal.org/smash/record.jsf?pid=diva2%3A832092&dswid=219`. https://bth.diva-portal.org/smash/record.jsf?pid=diva2

Braun, P. J. *et al*. (2019). Mobile Peer-to-Peer Assisted Coded Streaming. *IEEE Access*, 7:159332–159346. DOI: 10.1109/ACCESS.2019.2950800.

Casini, E., Benincasa, G., Morelli, A., Suri, N., and Breedy, M. (2016). An experimental evaluation of data distribution applications in tactical networks. In *MILCOM Military Communications Conference*, pages 1267–1272. IEEE. DOI: 10.1109/MILCOM.2016.7795505.

Chen, Y., Cai, Y., Zheng, H., Hu, J., and Li, J. (2022). Cooperative caching for scalable video coding using value-decomposed dimensional networks. *China Communications*, 19(9):146–161. DOI: 10.23919/JCC.2022.00.006.

Cohen, B. (2003). Incentives build robustness in BitTorrent. In *First Workshop on Economics of Peer-to-Peer System*, Berkeley, USA. Available at: `https://www.bittorrent.org/bittorrentecon.pdf`. https://www.bittorrent.org/bittorrentecon.pdf Access date: March 3rd, 2024.

Comsa, I.-S., Molnar, A., Tal, I., Imhof, C., Bergamin, P., Muntean, G.-M., Muntean, C. H., and Trestian, R. (2023). Improved Quality of Online Education Using Prioritized Multi-Agent Reinforcement Learning for Video Traffic Scheduling. *IEEE Transactions on Broadcasting*, 69(2):436–454. DOI: 10.1109/TBC.2023.3246815.

Fleury, M., Kanellopoulos, D., and Qadri, N. N. (2019). Video streaming over MANETs: An overview of techniques. *Multimedia Tools Applications*, 6:23749–23782. DOI: 10.1007/s11042-019-7679-0.

Ghani, A. *et al*. (2021). An Adaptive Video Streaming Framework for Peer-To-Peer 5G Networks: Paving the Road to 5G-IMS. In Ben Ahmed, M., Mellouli, S., Braganca, L., Anouar Abdelhakim, B., and Bernadetta, K. A., editors, *Emerging Trends in ICT for Sustainable Development*, pages 209–218, Cham. Springer International Publishing. DOI: 10.1007/978-3-030-53440-$0_2$3.

Gkortsilas, I., Deltouzos, K., Efthymiopoulou, M., Efthymiopoulos, N., and Denazis, S. (2012). Liquidstream — A high performance and stable scheduling architecture for P2P video on demand. In *2012 International Conference on Telecommunications and Multimedia (TEMU)*, pages 89–94. DOI: 10.1109/TEMU.2012.6294739.

Hossain, M. F., Mahin, A. U., Debnath, T., Mosharrof, F. B., and Islam, K. Z. (2019). Recent research in cloud radio access network (C-RAN) for 5G cellular systems - A survey. *Journal of Network and Computer Applications*, 139:31–48. DOI: 10.1016/j.jnca.2019.04.019.

Jenkac, H., Stockhammer, T., and Wen Xu (2006). Asynchronous and reliable on-demand media broadcast. *IEEE Network*, 20(2):14–20. DOI: 10.1109/MNET.2006.1607891.

Jones, Ben (2021). How to Make the Best Torrents. Available at: `https://torrentfreak.com/how-to-make-the-best-torrents-081121/`. https://torrentfreak.com/how-to-make-the-best-torrents-081121/. Access date: March 3rd, 2024.

Kavitha, S. and Latha, T. (2016). Efficient Buffer Management Protocol for Multicast Streaming in MANET. *Procedia Computer Science*, 92:222–232. DOI: 10.1016/j.procs.2016.07.349.

Kim, H. and Chung, K. (2020). Multipath-Based HTTP Adaptive Streaming Scheme for the 5G Network. *IEEE Access*, 8. DOI: 10.1109/ACCESS.2020.3038854.

Krifa, A., Sbai, M. K., Barakat, C., and Turletti, T. (2009). BitHoc: A content sharing application for wireless ad hoc networks. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–3, Galveston, TX, USA. DOI: 10.1109/PERCOM.2009.4912792.

Kumar, S., Wang, N., Rahulan, Y., and Evans, B. (2022). Edge Computing-Based Layered Video Streaming Over Integrated Satellite and Terrestrial 5G Networks. *IEEE Access*, 10:19971–19985. DOI: 10.1109/ACCESS.2022.3151998.

Legout, A., Urvoy-Keller, G., and Michiardi, P. (2005). Understanding BitTorrent: An Experimental Perspective. Technical report. Available at: `https://hal.inria.fr/inria-00000156`. https://hal.inria.fr/inria-00000156 Access date: March 3rd, 2024.

Legout, A., Urvoy-Keller, G., and Michiardi, P. (2006). Rarest first and choke algorithms are enough. In *IMC 2006, ACM SIGCOMM/USENIX conference*, Rio de Janeiro, RJ, Brazil. Available at: `http://conferences.sigcomm.org/imc/2006/papers/p20-legout.pdf`. http://conferences.sigcomm.org/imc/2006/papers/p20-legout.pdf Access date: March 3rd, 2024.

Leung, M. and Chan, S. . G. (2007). Broadcast-Based Peer-to-Peer Collaborative Video Streaming Among Mobiles. *IEEE Transactions on Broadcasting*, 53(1):350–361. DOI: 10.1109/TBC.2006.889093.

Li, K., Ai, X., Fang, J., Zhou, B., Le, L., and Wen, J. (2021). Coordination of macro base stations for 5g network with user clustering. *Sensors*, 21(16). DOI: 10.3390/s21165501.

Li, Y., Ma, H., Wang, L., Mao, S., and Wang, G. (2022). Optimized content caching and user association for edge computing in densely deployed heterogeneous networks. *IEEE Transactions on Mobile Computing*, 21(6):2130–2142. DOI: 10.1109/TMC.2020.3033563.

Lin, P., Ning, Z., Zhang, Z., Liu, Y., Yu, F. R., and Leung, V. C. M. (2023). Joint Optimization of Preference-Aware Caching and Content Migration in Cost-Efficient Mobile Edge Networks. *IEEE Transactions on Wireless Communications*, pages 1–1. DOI: 10.1109/TWC.2023.3323464.

Mbala, R. M., Nlong, J. M., and Kamdjoug, J.-R. K. (2021). A Framework for Multi-Hop Ad-Hoc Networking over Wi-Fi Direct with Android Smart Devices. *Communications and Network*, 13(4):143–158. DOI: 10.1109/COMCAS.2017.8244815.

Molner, N., de la Oliva, A., Stavrakakis, I., and Azcorra, A. (2019). Optimization of an integrated fronthaul/backhaul network under path and delay constraints. *Ad Hoc Networks*, 83:41–54. DOI: 10.1016/j.adhoc.2018.08.025.

Montresor, A. and Jelasity, M. (2009). PeerSim: A scalable P2P simulator. In *IEEE Ninth International Conference on Peer-to-Peer Computing*, Washington, USA. DOI: 10.1109/P2P.2009.5284506.

Munjal, A., Camp, T., and Navidi, W. C. (2011). SMOOTH: A Simple Way to Model Human Mobility. In *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '11, page 351–360, New York, NY, USA. ACM. DOI: 10.1145/2068897.2068957.

Narayanan, A., Ramadan, E., Carpenter, J., Liu, Q., Liu, Y., Qian, F., and Zhang, Z.-L. (2020). A First Look at Commercial 5G Performance on Smartphones. In *Proceedings of The Web Conference 2020*, page 894–905, New York, NY, USA.

Ng, D. T. K., Ching, A. C. H., and Law, S. W. (2023). Online learning in management education amid the pandemic: A bibliometric and content analysis. *The International Journal of Management Education*, 21(2):100796. DOI: 10.1016/j.ijme.2023.100796.

Palit, B., Sen, A., Mondal, A., Zunaid, A., Jayatheerthan, J., and Chakraborty, S. (2023). Improving ue energy efficiency through network-aware video streaming over 5g. *IEEE Transactions on Network and Service Management*, 20(3):3487–3500. DOI: 10.1109/TNSM.2023.3250520.

Pana, V. S., Babalola, O. P., and Balyan, V. (2022). 5G radio access networks: A survey. *Array*, 14:100170. DOI: 10.1016/j.array.2022.100170.

Polakovič, A., Vargic, R., and Rozinaj, G. (2018). Adaptive Multimedia Content Delivery in 5G Networks using DASH and Saliency Information. In *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5. DOI: 10.1109/IWSSIP.2018.8439215.

Quental, N. C. and Gonçalves, P. A. d. S. (2011). Exploiting application-layer strategies for improving BitTorrent performance over MANETs. In *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, pages 691–692. IEEE. DOI: 10.1109/CCNC.2011.5766574.

Ramadan, E., Narayanan, A., Dayalan, U. K., Fezeu, R. A. K., Qian, F., and Zhang, Z.-L. (2021). Case for 5G-Aware Video Streaming Applications. In *Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases*, 5G-MeMU '21, page 27–34, New York, NY, USA. DOI: 10.1145/3472771.3474036.

Rethfeldt, M., Beichler, B., Danielis, P., Uster, F., Haubelt, C., and Timmermann, D. (2018). MeNTor: A wireless-mesh-network-aware data dissemination overlay based on BitTorrent. *Ad Hoc Networks*, 79:146–159. DOI: 10.1016/j.adhoc.2018.06.013.

Rocha, V. and Rodrigues, C. K. S. (2023). BitCover: Enhanced BitTorrent for interactive VoD streaming over 5G and WiFi-Direct. *Ad Hoc Networks*, 140:103040. DOI: 10.1016/j.adhoc.2022.103040.

Rodrigues, C. K. S. (2018). Efficient BitTorrent-Like Algorithms for Interactive On-Demand Multimedia Streaming over MANETs. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, WebMedia '18, page 29–36, New York, USA. ACM. DOI: 10.1145/3243082.3243083.

Rodrigues, C. K. S. and Rocha, V. (2021). Enhancing BitTorrent for efficient interactive video-on-demand streaming over MANETs. *Journal of Network and Computer Applications*, 174. DOI: 10.1016/j.jnca.2020.102906.

Rodrigues, C. K. S. and Rocha, V. E. M. (2019). Towards Adapting BitTorrent for Interactive On-Demand Multimedia Streaming over MANETs. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*, WebMedia '19, page 469–472, New York, USA. ACM. DOI: 10.1145/3323503.3360618.

Safavat, S., Sapavath, N. N., and Rawat, D. B. (2020). Recent advances in mobile edge computing and content caching. *Digital Communications and Networks*, 6(2):189–194. DOI: 10.1016/j.dcan.2019.08.004.

Salhi, E., Sbai, M. K., and Barakat, C. (2009). Neighborhood selection in mobile P2P networks. In Chaintreau, Augustin, Magnien, and Clemence, editors, *Algotel*, Carry-Le-Rouet, France. Available at: `https://hal.inria.fr/inria-00383344/file/algotel.pdf`. https://hal.inria.fr/inria-00383344/file/algotel.pdf.

Sandvine (2023). 2023 Global Internet Phenomena Report. White Paper. Available at: `https://www.sandvine.com/global-internet-phenomena-report-2023`. https://www.sandvine.com/global-internet-phenomena-report-2023.

Sbai, M. K. and Barakat, C. (2009). Revisiting P2P Content Sharing in Wireless Ad Hoc Networks. In Spyropoulos, T. and Hummel, K. A., editors, *Self-Organizing Systems*, pages 13–25, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-10865-5_2.

Sbai, M. K., Barakat, C., Choi, J., Hamra, A. A., and Turletti, T. (2008). Adapting BitTorrent to Wireless Ad Hoc Networks. In Coudert, D., Simplot-Ryl, D., and Stojmenovic, I., editors, *Ad-hoc, Mobile and Wireless Networks*, pages 189–203, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-85209-4.

Schoonwinkel, D. (2016). *Practical Measurements of Wi-Fi Direct in Content Sharing, Social Gaming Android Applications*. PhD thesis, Stellenbosch: Stellenbosch University. Available at: `https://api.semanticscholar.org/CorpusID:63755014`. https://api.semanticscholar.org/CorpusID:63755014.

Shah, N., Abid, S., Qian, D., and Mehmood, W. (2017). A survey of P2P content sharing in MANETs. *Computers and Electrical Engineering*, 57:55–68. DOI: 10.1016/j.compeleceng.2016.12.013.

Shrama, L., Javali, A., and Routray, S. K. (2020). An Overview of High Speed Streaming in 5G. In *2020 International Conference on Inventive Computation Technologies (ICICT)*, pages 557–562. DOI: 10.1109/ICICT48043.2020.9112489.

Soldani, D. and Manzalini, A. (2015). Horizon 2020 and Beyond: On the 5G Operating System for a True Digital Society. *IEEE Vehicular Technology Magazine*, 10(1):32–42. DOI: 10.1109/MVT.2014.2380581.

Sun, Y., Peng, M., and Mao, S. (2019). A Game-Theoretic Approach to Cache and Radio Resource Management in Fog Radio Access Networks. *IEEE Transactions on Vehicular Technology*, 68(10):10145–10159. DOI: 10.1109/TVT.2019.2935098.

Tseng, T. H., Wu, T.-Y., Lian, Y.-H., and Zhuang, B.-K. (2023). Developing a value-based online learning model to predict learner's reactions to internet entrepreneurship education: The moderating role of platform type. *The International Journal of Management Education*, 21(3):100867. DOI: 10.1016/j.ijme.2023.100867.

Wang, B., Kurose, J., Shenoy, P., and Towsley, D. (2008). Multimedia Streaming via TCP: An Analytic Performance Study. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(2). DOI: 10.1145/1352012.1352020.

Wu, W., Yang, F., Gao, Y., Wang, X., Si, P., Zhang, Y., and Yu, F. R. (2022). Distributed Handoff Problem in Heterogeneous Networks with End-to-End Network Slicing: Decentralized Markov Decision Process-based Modeling and Solution. *IEEE Transactions on Wireless Communications*, pages 1–1. DOI: 10.1109/TWC.2022.3190883.

Xu, D., Zhou, A., Zhang, X., Wang, G., Liu, X., An, C., Shi, Y., Liu, L., and Ma, H. (2020). Understanding Operational 5G: A First Measurement Study on Its Coverage, Performance and Energy Consumption. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication*, page 479–494, New York, NY, USA. DOI: 10.1145/3387514.3405882.

Yang, Y., Zhang, X., and Zhang, L. (2017). MTV: Mobile BitTorrent Video Sharing Using Harmonized LTE and WiFi Coexistence. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5. DOI: 10.1109/VTCSpring.2017.8108300.

YouTube Help (2021). Choose live encoder settings, bitrates, and resolutions. Available at: `https://support.google.com/youtube/answer/2853702`. https://support.google.com/youtube/answer/2853702.

Zhang, T. and Mao, S. (2019). Cooperative Caching for Scalable Video Transmissions Over Heterogeneous Networks. *IEEE Networking Letters*, 1(2):63–67. DOI: 10.1109/LNET.2019.2911972.