Ad Hoc Networks 140 (2023) 103040

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

BitCover: Enhanced BitTorrent for interactive VoD streaming over 5G and WiFi-Direct

Vladimir Rocha *, Carlo K. da S. Rodrigues

Center for Mathematics, Computing and Cognition – CMCC Federal University of ABC – UFABC, Avenida dos Estados, 5001, CEP 09210-580, Santo André, SP, Brazil

ARTICLE INFO

MSC:
68M12
68M14

Keywords:
Streaming
5G
BitTorrent
WiFi-direct
Mobility
Interactivity

ABSTRACT

This paper proposes the BitCover algorithm, which is a BitTorrent-based solution for video-on-demand streaming with user interactivity over 5G cellular networks. Its design primarily resides in: (i) forming clusters of mobile nodes to cooperatively exchange video pieces during playback, (ii) leveraging WiFi-Direct communication to offload part of the video-piece delivery from the 5G mobile-backhaul network, and (iii) deploying a novel peer-selection policy which improves communication-channel usage by prioritizing peers that own more downloaded video-pieces. Performance evaluation is carried out through simulations, assessing a set of metrics on a variety of multimedia scenarios. The final results are able to outline the algorithm's effectiveness compared to other recent literature proposals. Within this context, our key contribution is thus to provide valuable insights for multimedia-application projects concerning 5G-cellular networks.

1. Introduction

The today's well-known notable appeal of video-on-demand (VoD) streaming service has at first posed a huge challenge for cellular network operators: the increase in video traffic greatly outpaced the improvement on the cellular network capacity [1]. Nevertheless, as the 5G technology started being deployed, a new communication era has then been settled. This technology is seen as a monumental shift in cellular communication. This is because it is believed to hold tremendous potential for spurring innovations across many vertical industries due to its promised multi-Gbps speed, sub-10 ms latency, and massive connectivity [2–4].

Notwithstanding, high-performance backhauling systems for the radio access network are mandatory for achieving the 5G's entire potential. This is because the high traffic load inside each 5G cell site needs to be eventually forwarded to the core network using the backhaul network. Thereby, the backhaul-network bandwidth capacity may become a serious bottleneck for the 5G-communication's final performance. To help tackling this issue, one alternative is certainly to strive to optimize the bandwidth usage within the cell site by deploying an effective bandwidth-sharing algorithm. This algorithm would therefore help to best leverage the full potential of the 5G technology to provide video-streaming service with adequate system quality of service (QoS), e.g., a system download rate above the video-encoding rate, besides an adequate quality of experience (QoE) on the client side, e.g., a smooth local video playback [5–8].

It is worth mentioning that the Device-to-Device (D2D) communication technology has been envisioned as an orthogonal approach to contribute with the thorough exploitation of the 5G-network capabilities. It has been shown that the D2D communication may improve resource utilization in cellular networks by offloading the traffic from the mobile-backhaul to local direct links between mobile nodes. Under this paradigm, the data traffic through cellular base stations is expected to be more effectively optimized [5–7].

The above scenario is the motivation for this paper, whose research question is summarized as follows: Can we think up an effective solution based on the popular BitTorrent protocol [9,10] combined with the WiFi-Direct technology [11,12] to best leverage the full potential of the 5G technology aiming at adequate interactive video-on-demand (VoD) service? To respond to this, we propose a new BitTorrent-based algorithm called BitCover. Its design concept resides in the combination of several criteria to wisely create clusters (i.e., neighborhoods) of mobile nodes (i.e., network peers) to disseminate video pieces during playback, in addition to leveraging WiFi-Direct technology to alleviate the piece-delivery traffic on the mobile-backhaul, and deploying a novel peer-selection policy which improves communication-channel usage by prioritizing peers that own more downloaded video-pieces.

In BitCover, the decision for using 5G or WiFi-Direct links is function of the physical distance between the two communicating endpoints (i.e., the network peers which are exchanging video pieces). In case

---

* Corresponding author.
E-mail addresses: vladimir.rocha@ufabc.edu.br (V. Rocha), carlo.kleber@ufabc.edu.br (C. K. da S. Rodrigues).

https://doi.org/10.1016/j.adhoc.2022.103040
Received 9 August 2022; Received in revised form 22 October 2022; Accepted 6 November 2022

the distance is beyond the WiFi-Direct radio range, the piece delivery occurs through 5G using the mobile-backhaul network; otherwise, the endpoints deploy a WiFi-Direct link only. This is done in such a manner that the pivotal features of both 5G and WiFi-Direct communication technologies may be jointly leveraged.

Performance evaluation is done by means of simulations on various interactive streaming scenarios, in which different metrics are deployed with the prime focus of assessing the system QoS and the client QoE. Within this context, the key contribution of this article is thus to provide the literature with valuable insights for multimedia-application projects, mainly targeting at interactive video-on-demand streaming over 5G cellular networks.

The remainder of this article is organized as follows. Section 2 reviews the original BitTorrent protocol, in addition to characterizing 5G and WiFi-Direct communication technologies. Section 3 discusses related work. In Section 4, we explain our new proposal. Section 5 is devoted to experiments, results and analyses. At last, general conclusions and future work constitute Section 6.

2. Background

2.1. The BitTorrent protocol

The BitTorrent protocol is worldwide recognized by its distinguished efficiency for file replication. Its operation is described in what follows. In BitTorrent, a set of network peers wanting to receive a file is called *swarm*. Let F be the wanted file. For its replication, it must be first divided into pieces and each piece is then divided into blocks. Although blocks are the data unit in the physical network, the replication analysis considers the transmitted pieces only [9,10].

The neighbors of a *swarm* peer p are those other *swarm* peers to which, and from which, peer p can send and receive the pieces of file F. There are two types of *swarm* peers: leecher and seed. The former is a peer that downloads pieces of file F, but also allows other peers to download pieces from it. The latter is a peer that already has all the pieces of file F, but remains in the system to allow other peers to download pieces from it [9]. The piece-exchange process among peers is governed by the peer-selection and the piece-selection policies, detailed in the following subsections.

2.1.1. The peer-selection policy

The peer-selection policy, also called the *choke* algorithm, allows each *swarm* peer to select which other peers from its neighbor set may request the pieces it owns. For that, the peer transfer capacity is divided into four logical data slots, which are filled by two processes: the regular and the optimistic unchokings. In the first process, the three neighbors that upload pieces to p, in case p is a leecher (or download pieces from p, in case p is a seed), at the highest average rates are selected and put in the *unchoked* state, whereas the other neighbors are put in the *choked* state. This selection is typically repeated at every 10 seconds [9,10]. In the second process, peer p randomly selects another peer from its neighbor set and puts it in the *unchoked* state. This selection is typically repeated at every 30 s [9].

2.1.2. The piece-selection policy

The piece-selection policy allows each *swarm* peer to decide which pieces to request when it is in the *unchoked* state. The decision follows the rarest-piece policy, i.e., to find the least replicated piece being shared by the neighbors. For that, peer p keeps a list containing the number of copies of each piece in its neighbor set. This information is used to define the set of the rarest pieces, i.e., the set of the least replicated pieces. Let m be the number of copies of the rarest piece. The index of each piece with m copies in the neighbor set is added to the set of rarest pieces. Peer p then requests the next piece taking into account the set of rarest pieces as well as the available pieces in the neighbor that has just put it in the *unchoked* state. Finally, after receiving the piece, peer p tells all its neighbors about the received piece [9,10].

2.2. The 5G technology

The 5G technology is the fifth generation standard for broadband cellular networks, which cellular phone companies began deploying worldwide in 2019, and is the planned successor to the 4G networks. 5G networks are predicted to have more than 1.7 billion subscribers worldwide by 2025, according to the GSM Association. Like its predecessors, 5G networks are cellular networks, in which the service area is divided into small geographical areas called cells [2,3,13].

All 5G wireless devices in a cell are connected to the Internet and telephone network by radio waves through a local antenna in the cell. The main advantage of the new networks is that they will provide amazing download speeds, eventually up to 10 Gbps. Under the hood, this is achieved by a series of innovations including massive MIMO (multiple-input multiple-output), advanced channel coding, and scalable modulation. Due to the notable capacity, it is expected that these new networks will increasingly be used by Internet Service Providers (ISPs) to offer connectivity for laptops and desktop computers, competing with existing infrastructure such as cable Internet, and also will make possible new applications in Internet of Things (IoT) and machine-to-machine (M2M) areas [3,13].

2.3. The WiFi-Direct technology

WiFi-Direct is a Wi-Fi standard for peer-to-peer wireless connections that allows two electronic devices to establish a direct Wi-Fi connection without an intermediary wireless access point, router, or Internet connection. WiFi Direct is single-hop communication, rather than multihop communication like wireless ad hoc networks [11,12].

This technology is useful for everything, from Internet browsing to file transfer. Additionally, it has the ability to connect devices even if they are from different manufacturers. More precisely, only one of the Wi-Fi devices needs to be compliant with WiFi-Direct to establish a peer-to-peer connection that transfers data directly between them, given a simple device configuration [11,12].

3. Related work

By definition, the interactive VoD streaming serves the content from any position (i.e., video pieces) in the video file, regardless of the request time. For adequate QoS and QoE on the side of the client, we have that time delay between the request and the reception of video by the end user is critical. Hence, and effective algorithmic solution for VoD streaming needs not only to ensure a smooth playback, but also minimizes the piece-delivery delay when users move from one position to another in the video file [14–16].

Considering the above and mostly focusing on algorithms for effective interactive VoD streaming over cellular networks, we next discuss some of the most important and recent related work. The pivotal goal is to especially provide the reader with an essential state-of-the-art overview targeted at our research problem. Nevertheless, we highlight that, for practical deployment, algorithmic solutions, infrastructure-based solutions (e.g., MIMO antennas, content delivery network (CDN), and software defined network (SDN)) are orthogonal to each other, so being able to be used in a combined smart manner (e.g., [6,17–19]).

In [20], mainly inspired by BitTorrent, the authors propose a dynamic device-to-device (D2D) system for video sharing which uses cellular interface and WiFi-Direct technology. They consider node clusters (by physical proximity) watching the same video, besides cellular-traffic offloaded onto WiFi-Direct. Benefiting from video sharing and properties of BitTorrent, their system not only brings improvements on downloading rate plus reduction on traffic consumption, but also supports dynamic joining and leaving. However, despite the experiments demonstrate several advantages of the proposed system (e.g., speed, traffic, and dynamic adaption), the examined scenarios are indeed limited to few devices only, what ends up precluding blunter conclusions.

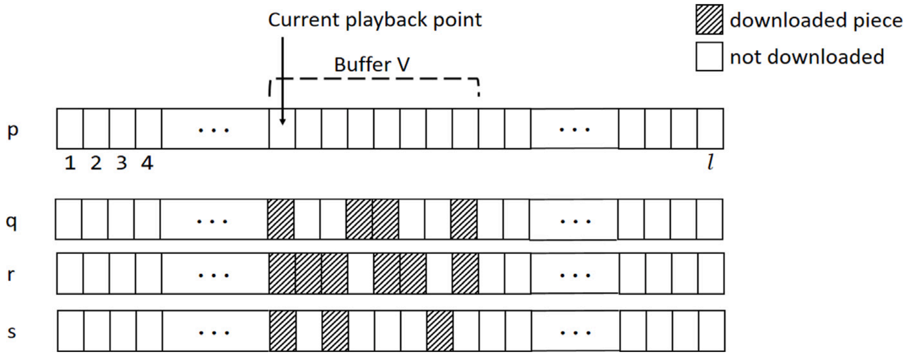


Fig. 1. Piece-coverage criterion.

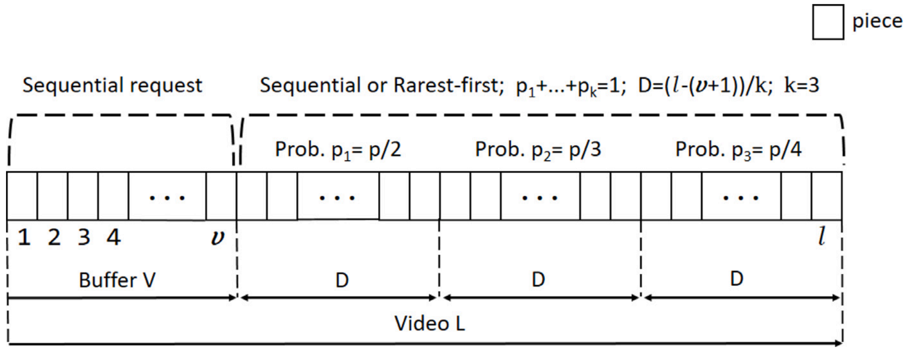


Fig. 2. *Sliding window* W and buffer V.

It is worth saying that, even though the aforementioned work does not refer specifically to 5G technology, but rather to 4G instead, the algorithm therein presented is valuable because it successfully clarifies the overall streaming operation over cellular networks. Likewise, there are other works devoted to 4G (e.g., [6,7,21]) which may also be seen as baseline and solid guidance for the development of new algorithms target at 5G cellular networks, since their respective solutions and results are certain to be inspiration for specific 5G-streaming solutions, which are still relatively scarce as well as are often at their first steps of evolution. Next, we comment on four recent works devoted specifically to 5G networks.

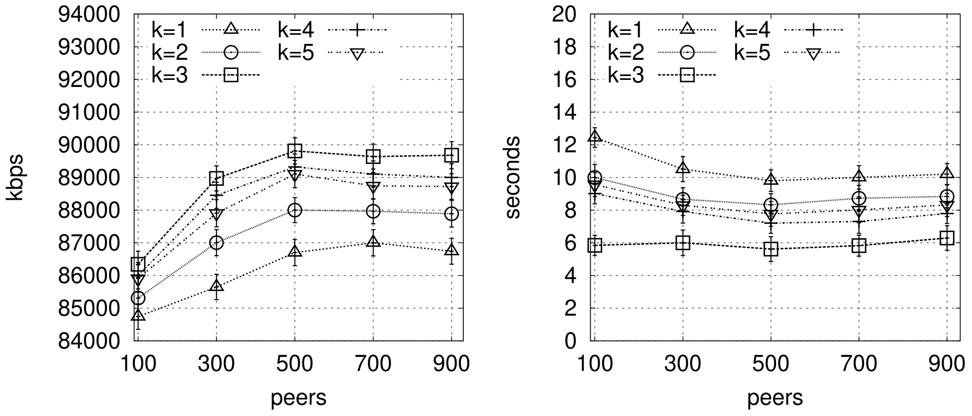
In [5], the authors' proposal is based on the idea of dividing the user local buffer into playing buffer, urgent buffer, and prefetching buffer. The playing buffer contains the file pieces already received by the user; the urgent buffer contains the video pieces for continuous playback; and the prefetching buffer contains the future video pieces in order to support the jump from one position to another. To retrieve pieces for the urgent buffer, a sequential-request policy is used, whereas a rarest-first-request policy is deployed for the prefetching buffer. Despite the limited experiments (i.e., a single scenario and users are static), the simulation results show that the average number and duration of stalling events are adequately reduced comparing to several past literature proposals.

In [22], the authors aim to improve QoE for video consumers by adaptive delivery of multimedia content. At first, they summarize ways of detecting salient regions in observed scenes that attract human gaze besides the current state of dynamic adaptive streaming over HTTP (DASH). In the sequence, they then propose to extend the DASH video delivery system, based on saliency information gathered using eye tracking. The proposed extension enables to enhance the quality in certain regions in video and is particularly suitable for adaptive multimedia content delivery in 5G networks. Despite the solid theoretical contribution, no results of performance evaluation are though presented or discussed unfortunately.

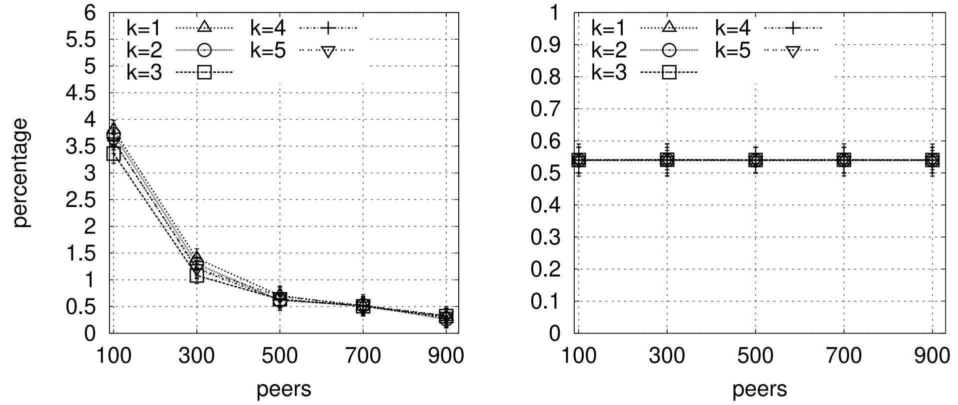
Table 1
Synthesis of related work.

References	Type	Scope	Specificity
[14–16]	Survey	Theory, concepts, definitions, models, and challenges.	Mostly focused on theory, with no new experimental results and findings, but serving as relevant baseline studies for future research.
[6,17–19]	Research	Video-streaming applications for 5G communication networks	Mostly focused on below-transport-layer protocol solutions, e.g., high-speed train and vehicular applications.
[6,7,20,21]	Research	Video-streaming applications for ordinary 4G communication networks.	Chiefly serve as solid baseline research for novel solutions targeted at 5G communication networks.
[5,22–24]	Research	Video-streaming applications for 5G communication networks	Mostly focused on above-transport-layer protocol solutions, e.g., SVC-based and DASH-based schemes.

In [23], the authors propose a multipath-based transmission scheme considering DASH in a 5G-deployed communication scenario. More punctually, they propose a rate adaptation scheme to improve bandwidth utilization and QoE, a segment scheduler to solve the out-of-order problem in multipath-based transmission, and an offloading control scheme based on a partial-segment-based request policy to



(a) Download Rate. (b) Discontinuity Time.



(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 3. Performance of BitCover by varying k in Scenario 1.

prevent buffer underflow due to sudden bandwidth reduction in the non-LOS 5G environment. Quantitative results from simulations suggest that their scheme solves the shortcomings of the so current existing solutions, improving both bandwidth utilization and QoE on the client side.

Finally, in [24], the authors propose to use a combination of the IP Multimedia Subsystem (IMS), the BitTorrent protocol, and an adaptive video streaming scheme to improve the video transfer in 5G networks. The IMS is used to authenticate the peers and to notify its neighbors when its IPs change, using the Service Initiation Protocol (SIP). The BitTorrent protocol allows authenticated peers to join and share a video, which is coded with different SVC (Scalable Video Coding) levels. When a peer wants to download a video piece, it must analyze what SVC level is needed in order to accelerate the transfer. To obtain the best SVC level, they define a process that considers several aspects, such as the space (e.g., device resolution), time (e.g., frame rate), quality (e.g., device capacity), and the peers that could provide the video piece. The simulation results show that the proposal decreases the download total time, even when the traffic congestion and loss rate increase.

In view of the aforementioned works, which are summarized in Table 1, the pivotal differential of our research is that we herein propose a new application-layer algorithm which captures the most important findings of the state-of-the-art literature works on interactive VoD streaming over cellular networks, since it innovatively strives to combine the promising D2D-communication technology, the efficient BitTorrent' replication mechanism, and the 5G technology's capabilities as well.

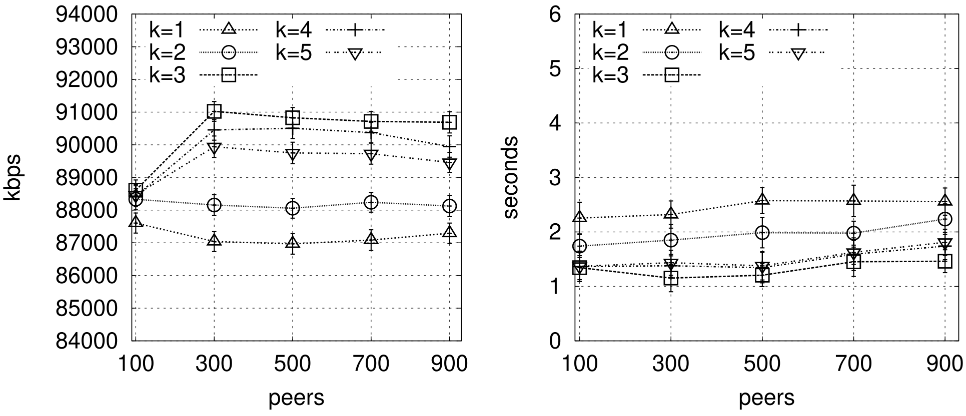
4. The BitCover algorithm

The explanation of BitCover is herein done through the presentation of its two core policies: peer selection and piece selection. To ease understanding, each of them is explained in different subsections in the sequence. Moreover, we also consider the following notation and assumptions. Let g be a peer which is neighbor of another peer p. Peer g is said to be interested in peer p when peer p has pieces that peer g does not have. Conversely, peer g is not interested in peer p when peer p only has a subset of the pieces of peer g.

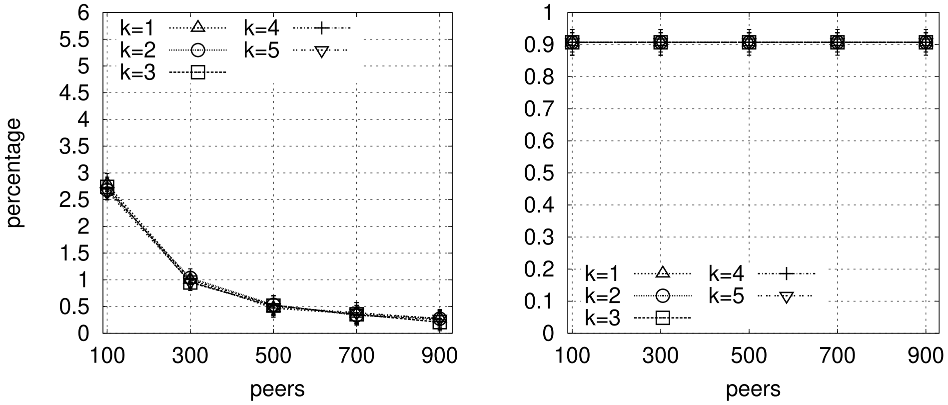
4.1. Peer-selection policy

As in the original BitTorrent algorithm, BitCover's peer selection also consists of two processes: regular unchoking and optimistic unchoking, as explained in what follows. The peer's upload capacity is divided into y data slots. At every δt seconds, these two steps take place:

1. Neighbors interested in peer p are ordered by descending average *upload rate*. That is, the neighbors interested in peer p are ordered according to how fast they can send pieces to it on average.
2. The f first fastest neighbors are then ordered by descending piece-coverage, and one data slot is allocated to each of them up to reaching $y - 1$ data slots. Piece-coverage counts the number of already-downloaded pieces a peer g (among the f first fastest peers) has within its local buffer, V (detailed in the next section). The rationale for the piece-coverage criterion is



(a) Download Rate. (b) Discontinuity Time.



(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 4. Performance of BitCover by varying k in Scenario 2.

that the selected peer will have more pieces to share, so optimizing network communication-channel usage, besides avoiding time waste for managing a greater number of communication connections between peers.

An example of the piece-coverage criterion is given in Fig. 1. Let peers q, r, and s be interested in peer p. It happens that peer p prioritizes to unchoke the peer which has more downloaded pieces in its local buffer V. In this case, peer p will choose peer r, since it is the one with more downloaded pieces.

The above two steps refer to the regular unchoking. In the optimistic unchoking, peer p randomly selects r peers from its neighbor set, and assigns r data slots to them (one for each). This is repeated at every δt seconds too.

4.2. Piece-selection policy

The piece-selection policy is based on a criterion that combines a *sliding window*, W, and an interior buffer, V, as explained in what follows.

Assume the video file, L, is divided into l pieces: $1, 2, \ldots, l$. Let d be the piece that corresponds to the current playback point. Still, let w be the *window* size in number of pieces. Hence, W comprises the following pieces: $[d; d + w]$. Furthermore, W is dynamically updated as the pieces are played and interactivity actions are performed. For example, let Δ_{play} be the number of pieces just played. The first piece of W is then updated to $(d + \Delta_{play})$, while the last piece is updated to $(d + w + \Delta_{play})$.

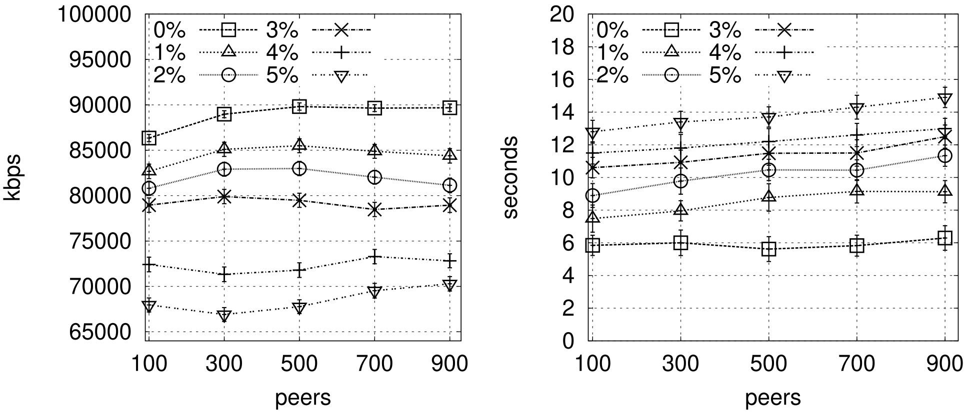
Now, let v be the buffer size in number of pieces. The first piece of V is always coincident with the first piece of W. Before a peer g requests the next piece from a peer p, it checks if all pieces of buffer V have already been retrieved. If do not, peer g requests the next missing in-order piece within V.

If the buffer is full, assume that the first piece of the window is x, consequently the size corresponding to the rest of the video (i.e., $l - (v + x)$) is divided into k identical parts of size $D = (l - (v + x))/k$, in number of pieces. The next piece chosen is probabilistically requested from one of the k defined parts, as explained below.

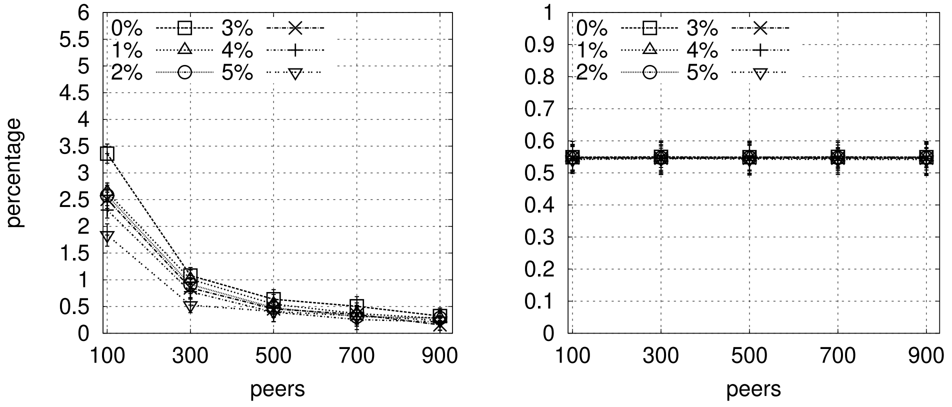
Each ith part, $1 \leq i \leq k$, is chosen with probability $p_i = p/(i + 1)$, with p obtained from $\sum_{i=1}^{k} p_i = 1$. The piece requested in each of the k parts can be done sequentially or according to some other policy, such as the rarest-first (i.e., least replicated). In this work, we use the former. Fig. 2 illustrates this understanding for k=3 and x=1 and the relationship among the parameters mentioned above (i.e., W, V, D, and l). Finally, w and v can be calculated using the formula proposed in [25], in which $w \approx 0,37 \times l$ and $v \approx 0,4 \times w$.

5. Performance evaluation

By means of simulations, this section aims at carrying out the performance evaluation of the BitCover algorithm under different 5G streaming scenarios. For the sake of objectivity and ease of understanding, this section is divided into five different subsections as follows.



(a) Download Rate. (b) Discontinuity Time.



(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 5. Performance of BitCover by varying the packet-loss percentage in Scenario 1.

5.1. Network modeling

The experiments consider a network formed by n peers wanting to watch a same video file. Each peer plays the role of a mobile client that performs interactivity actions during video playback. This network encompasses a single 5G cell site covering a square area with a side length of 500 m. Peers' handoffs (i.e., ongoing-connection transfers between different cell sites) are not considered, and we have a steady-state analysis within the cell site, in which the value of n never changes: when a peer leaves, another one is added to the network. It is worth mentioning that, under a steady-state analysis, the state variables which define the system behavior are unchanging in time.

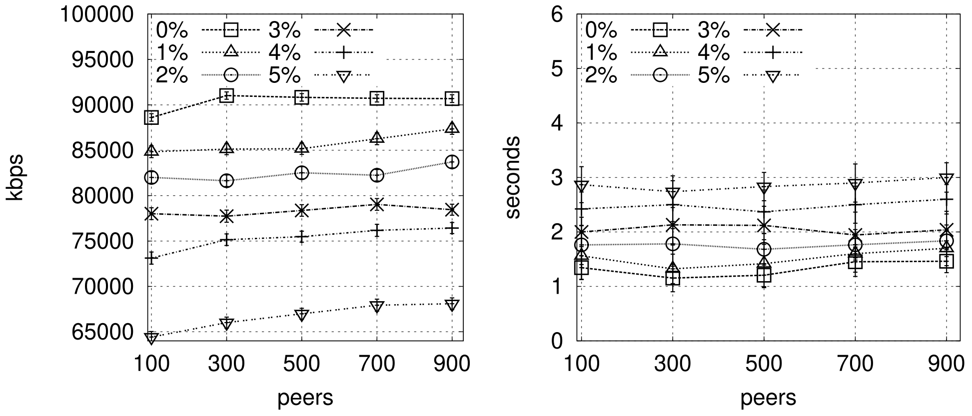
Among the n peers, there are always a tracker and a seed which operate uninterruptedly. The tracker may be reachable via 5G, and it is in charge of coordinating the communication between all peers within the cell site. This coordination especially includes creating a list of the peers that are currently watching the video. This list is periodically disseminated via broadcast to all peers within the cell site. By its turn, the seed may be reachable via 5G or WiFi-Direct, depending on the separation physical-distance as detailed next.

The data transmission, i.e., piece delivery, between two any communicating peers within the cell site can occur by means of 5G or WiFi-Direct. A peer's total communication-channel capacity is 100 Mbps when using the 5G technology, and 250 Mbps when using the WiFi-Direct technology, based on the 802.11 standard. Under a logical view, each peer has a communication channel capacity divided into five concurrent TCP data-stream slots: three upload slots for regular *unchoking*; one upload slot for *optimistic unchoking*; and one download slot for piece retrieval. Each slot's effective data rate might though not be the same, since it depends on the connection to the remote peer. Besides, video-piece packets can suffer from delays and be lost as well.

Table 2
WiFi-Direct configuration delays.

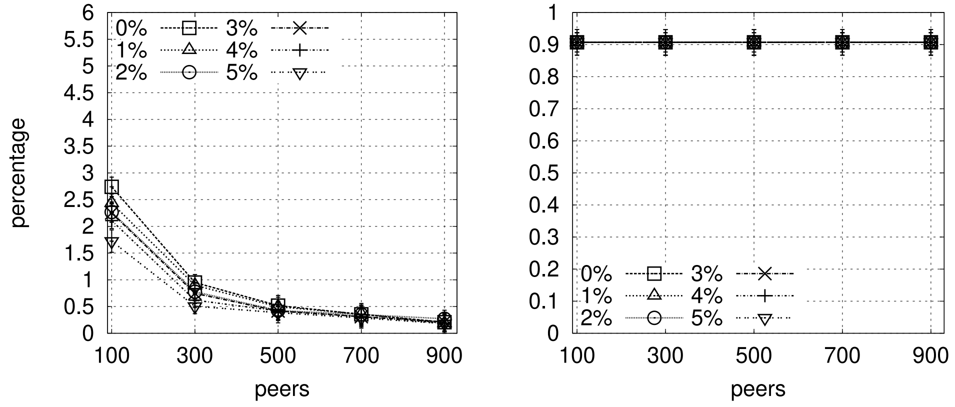
Parameter	Time (ms)	Definition
channelDelay	400	Time required to send a package at MAC layer.
authenticationDelay	100	Time needed to accept an invitation.
powerDelay	100	Time needed when the process goes to sleep.
switchingDelay	500	Time needed to find a channel at discovery phase.

For the WiFi-Direct connection, we use the WiFi-Direct implementation simulator [26], which includes both delays and losses. For the delays, the simulator specifies several different parameters [27] such as those informed in Table 2. All of them are assigned constant values, except the *channelDelay* parameter. It lies in the range of 0–400 ms according to the physical distance as obtained in [28,29]. For the losses, the simulator focuses on the physical distance, deploying the Friis's path loss formula to compute the signal strength between two peers. Thereby, delays and losses are primarily derived from physical distance, packet size, and signal strength [26]. For the 5G connection, we use delays randomly chosen within the interval of 5 to 20 ms besides an average packet-loss percentage of 4% [2,3]. These values are assumed to encompass all delays/losses occurring in transport and below layers as well [2,3].



(a) Download Rate.

(b) Discontinuity Time.



(c) Initial Seed Overloading.

(d) Piece Rate.

Fig. 6. Performance of BitCover by varying the packet-loss percentage in Scenario 2.

5.2. Mobility and interactivity of peers

Peers move inside an obstacle-free cell site according to the SMOOTH mobility model [30], whose traces are generated by Bonnmotion [31]. This model matches statistical features of real human movements, such as the distance covered by a peer when visiting a popular or the closest location, which follows an inverse power-law distribution with parameter α; the amount of time a peer waits in a location, which follows a truncated power-law distribution with parameter β; and the so-called communities formation, which refer to clusters of peers to represent the social human behavior. The cluster sizes vary in size in function of the location popularity.

The SMOOTH model fits adequately to our experiments, since it mimics real GPS traces collected from different outdoor scenarios (e.g., city, campus, among others). The main configuration values are shown in Table 3, which are those deployed in [32] for the State Fair scenario. This configuration is chosen due to its similarities with both a real-world 5G-cell square area and online-learning class scenarios used in our experiments later, in which people are usually together sharing a same physical environment.

Regarding interactivity, we adopt a behavior model inspired by the works of [25,33]. The interactivity actions may be of the following types: *Play*, *Pause*, *Jump Forwards* (*JF*), and *Jump Backwards* (*JB*). The action *Play* means that the peer is watching the video. The action *Pause* indicates that the playback is paused. The actions *JF* and *JB* stand for a jump to a playing point after and before the current one, respectively.

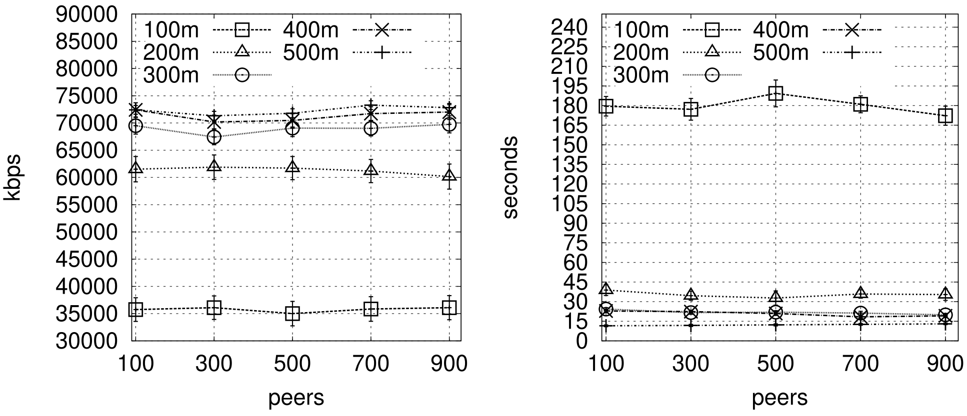
Table 3
SMOOTH configuration parameters.

Parameter	Value	Definition
n	100 to 900	Number of peers to be simulated.
$range$	100	Transmission range of a peer (in meters).
$clusters$	4	Number of clusters within the area.
α	3	In one extreme (i.e., $\alpha \rightarrow 0$), the next location to visit is selected randomly. In the other extreme (i.e., $\alpha \rightarrow \infty$), the next location to visit is the closest one.
β	1	In one extreme (i.e., $\beta \rightarrow 0$), mobile peers pause for a long time at few locations. In the other extreme (i.e., $\beta \rightarrow \infty$), mobile peers pause for a short time at few locations.

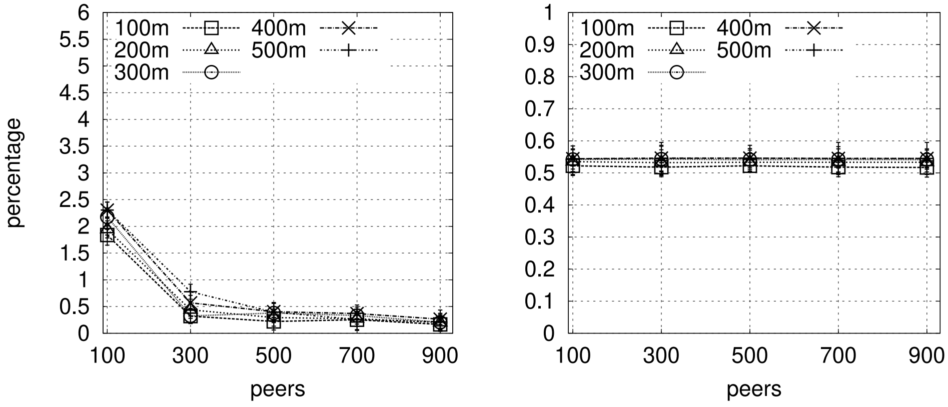
Table 4
Interactivity profiles.

Parameter	Low (LI)	High (HI)
λ	0.005/s	0.025/s
S	14.5% of f_{size}	1.5% of f_{size}
p_{play}; p_{pause}	0.89; 0.01	0.55; 0.15
p_{jf}; p_{jb}	0.05; 0.05	0.15; 0.15
f_{size}	10193 MB	5370 MB

Furthermore, *Play*, *Pause*, *JF*, and *JB* are triggered according to a Poisson distribution with rate λ, and probabilities p_{play}, p_{pause}, p_{jf}, and p_{jb}, respectively. *Play*, *Pause*, *JF*, and *JB* have each the same length



(a) Download Rate. (b) Discontinuity Time.



(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 7. Performance of BitCover by varying the area in Scenario 1.

Table 5
Scenario configuration.

Parameter	Scenario 1	Scenario 2
Video encoding	20 Mbps	20 Mbps
Video length	2199 s	4175 s
Video size	5370 MB	10193 MB
Piece size	4 MB	4 MB
Interactivity profile	High	Low

S, which is measured as a percentage of the video-file size, denoted by f_{size}. The interactivity profiles considered in the experiments are *Low Interactivity* (LI) and *High Interactivity* (HI). Table 4 lists the configuration values of these two profiles, which are the same as used in [25].

5.3. Scenario setup

The video file is encoded at 20 Mbps, which is the minimum bitrate for transmitting 4K-resolution at 60 frames per second, as suggested by YouTube streaming platform [34]. When the simulation begins, n peers join the network at random locations, defined by SMOOTH. Each peer p then performs interactivity actions until it visualizes the end of the file (i.e., the last piece) or downloads all pieces. When that occurs, peer p then leaves the network, and a new peer joins the network in a random location. The simulation ends when n peers leaves the network. It is important to note that the n peers that end the simulation did not necessarily initiate it.

We organize the experiments in two scenarios belonging to the online-learning domain. This domain is chosen due to its current relevance in modern society. The configuration for each scenario is based on the consolidated information given by [35,36], as detailed next. In the first scenario (Scenario 1), a high-interactivity user watches a short video lecture. The video lasts for $\approx$ 37 min (2199 s) for a total size of 5370 MB. In the second scenario (Scenario 2), a low-interactivity user watches a long video lecture of $\approx$ 70 min (4175 s) for a total size of 10193 MB. Moreover, we assume a video-piece size of 4 MB, as suggested in the works of [37,38]. For ease of reference and unless otherwise stated, Table 5 recaps all configuration values of the two streaming scenarios used in the experiments to follow.

5.4. Simulation and metrics

The experiments are implemented in the PeerSim [39] simulation environment. The hardware platform is an Intel Core i7 (2.6 GHz), 24 GB of RAM, running a GNU/Linux operating system. Table 6 has the four performance metrics assessed in the experiments. They may together yield evidences of system QoS and client QoE. The computed results have 95% confidence intervals that are within 5% of the reported values, for a total of 30 simulation runs.

For quality judgment, we assume D_R must be at least the same rate as the video encoding for a satisfactory QoS [40]. For D_T, we assume an upper limit of 5 ms (latency) for each visualized piece for a satisfactory QoE [41,42], implying a total interruption time of nearly 10 s [43,44]. For SO, we assume values below 3.4% for a suitable performance in P2P VoD server-assisted systems [45,46]. As for P_R, to the best of our

Table 6
Performance metrics used for evaluation.

Metric	Notation	Definition
Download rate	D_R	It estimates the peer's average rate (in kB/s) to receive video pieces that may be visualized by it.
Discontinuity time	D_T	It estimates the peer's total average interruption time (in seconds) during the video playback.
Initial seed overloading	S_O	It estimates the percentage of piece requests (over the total piece requests of all peers) that is handled by the initial seed.
Piece rate	P_R	It lies within the interval of 0.0 to 1.0. It is the ratio of V_P to T_P, where V_P estimates the peer's total average number of pieces received and visualized by it, whereas T_P estimates the peer's total average number of received pieces which may be visualized by it. The closer P_R is to 0.0 (1.0), the less (more) efficient the piece-selection policy becomes.

Table 7
Metric limits for the experiments.

Metric	Limit
Download rate	above 20 Mbps
Discontinuity time	below 10 s
Initial seed overloading	below 3.4%

Table 8
Configuration parameters of BitCover's peer-selection policy.

Parameter	Value	Rationale
y	4 slots	The same quantity of slots used by original BitTorrent regular unchoking.
f	30 peers	As suggested in [47] for speed performance.
r	1 slot	The same quantity of slots used by original BitTorrent optimistic unchoking.
δt	10 s	The same time used by original BitTorrent protocol.
WiFi-Direct coverage	100 m	We recall that the 802.11n standard has radio coverage of 70 m (indoor) and 250 m (outdoor). We therefore herein assume 100 m as a suitable WiFi-Direct radio coverage.

Table 9
Optimizations achieved by BitCover.

–	Scenario 1		Scenario 2	
Optimization	Min	Max	Min	Max
Gain at D_R	2.2%	4.2%	1.1%	4.8%
Reduction at D_T	44.9%	52.6%	40.3%	53.1%
Reduction at S_O	0.8%	16.1%	1.7%	26.1%
Gain at P_R	0.2%	0.4%	0.0%	0.1%

Table 10
Packets via WiFi-Direct.

Side length (R)	Percentage
100 m	80%
200 m	40%
300 m	20%
400 m	10%
500 m	5%

knowledge, there are not any literature references with numeric values adherent to our investigated scenarios. For the scenario configuration used in the experiments, Table 7 recaps the limits mentioned above.

5.5. Results and analysis

For ease of understanding, this subsection is divided into four other subsections as detailed next. Section 5.5.1 empirically delves into the BitCover's peer-selection policy. Sections 5.5.2 and 5.5.3 evaluate how BitCover behaves under noisy 5G and WiFi-Direct channels, respectively. At last, Section 5.5.4 carries out a competitive analysis between the BitCover, MTV [20], and IB-A [25] proposals.

5.5.1. Analysis of the peer-selection policy

This analysis is done in two parts. We first determine the near-ideal value of parameter k. By near-ideal value, we mean a value that is most likely to result in a suitable final performance of BitCover. Thereafter, we assess the optimization yielded by the deployment of the BitCover's piece-coverage criterion. Otherwise stated, Table 8 presents all configuration values of the peer-selection policy's parameters of BitCover.

(a) Near-ideal value of k

Fig. 3 refers to the analysis of Scenario 1, whereas Fig. 4 refers to Scenario 2. They both show the computed results for the previously defined performance metrics in function of the number of peers, n, within the simulated covered area.

See that the near ideal value of k may be inferred to be three, since it clearly best optimizes all performance metrics. Using smaller values of k tend to lead the BitCover's piece-selection policy to operate as an ordinary sequential-request policy (i.e., pieces are strictly requested in order), what is not adequate due to the user's interactivity profile. Besides, greater values of k eventually causes the BitCover's piece-selection policy to operate as a rarest-first policy (i.e., least-replicated pieces are requested first), what in not adequate either, since it compromises the continuity time and, hence, precludes adequate streaming services.

(b) Optimization due to the piece-coverage criterion

We herein mount Table 9. It is derived from the results shown in Figs. 3 and 4, and outlines the maximum and minimum optimizations achieved by BitCover with $k = 3$ compared to BitCover with $k = 1$. The former is the best configuration, since it deploys the near-ideal value of k. The latter relates to neglecting the piece-coverage criterion, since it mostly operates under a strict sequential piece-selection policy. We clarify that the maximum (minimum) optimization is computed by observing the maximum (minimum) difference between resulting values obtained by BitCover with $k = 3$ and with $k = 1$, respectively, considering each of the performance metrics. We may note that using the piece-coverage criterion indeed improves the system performance as a whole, and most notoriously the discontinuity-time metric (D_R), for which we achieve an impressive reduction of up to 53.1%.

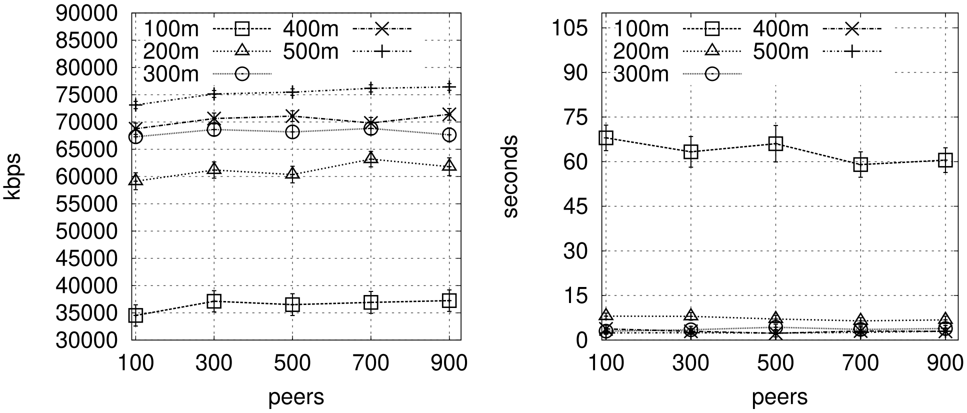
5.5.2. Degradation under noisy 5G channel

The packet losses considered in the experiments of previous subsection occurs exclusively in function of the Wi-Fi Direct loss. Nevertheless, noise interference may corrupt the packets being delivered through 5G channels too, thereby contributing to the total packet-loss in the network.

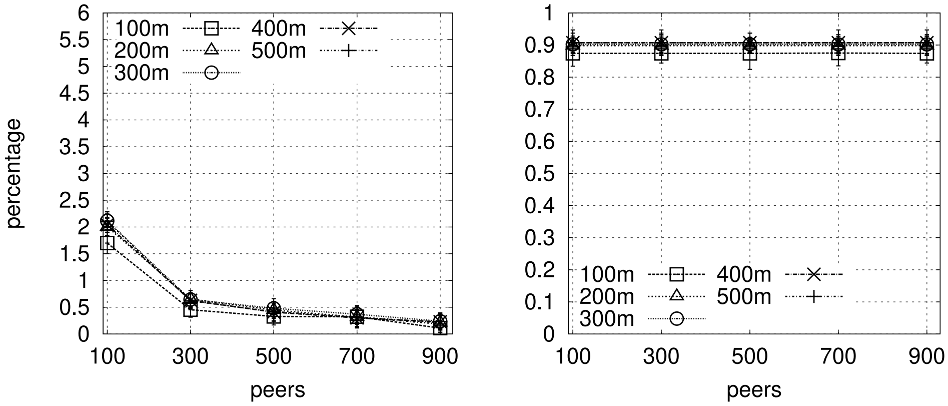
We herein therefore aim to assess the performance degradation of BitCover due to a noisy 5G channel, what results in a more realistic channel modeling. For that, we additionally consider a packet-loss

Table 11
Brief description of competing algorithms.

Reference	Algorithm	Description
[25]	IB-A	•It is focused on MANETs. It deploys δt =100 s to guarantee long-life endpoints' connections in highly dynamic networks. •Its peer-selection policy uses *upload rate* and *indirect reciprocity* as criteria. Peers that share more pieces on average are prioritized. •Its piece-selection policy uses a sliding window W and an interior buffer V (as in BitCover), but the video file is not partitioned. •Intermediary peers (between endpoints) store pieces routed through them.
[20]	MTV	•It uses a video server to send pieces (via 5G links) to peers which cannot get them from their neighbors via WiFi-Direct links. •Its peer-selection policy prioritizes those peers who have more downloaded-pieces. •Its piece-selection policy partitions the video into predefined windows, and pieces are then randomly selected from each of them.



(a) Download Rate. (b) Discontinuity Time.



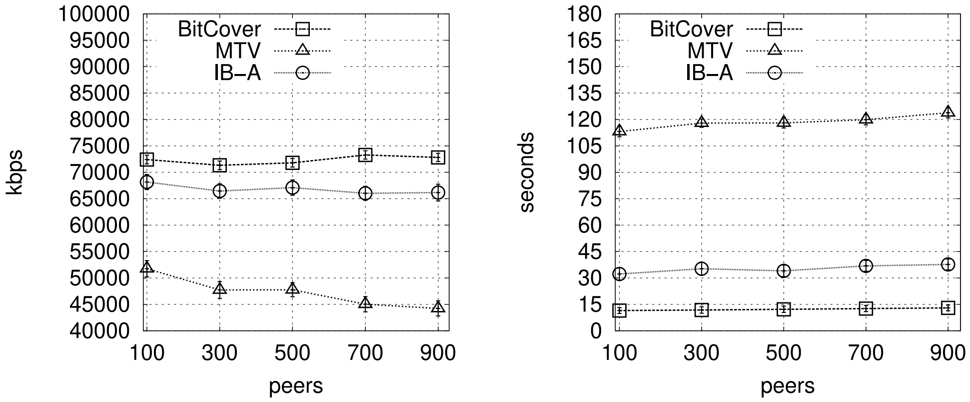
(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 8. Performance of BitCover by varying the area in scenario 2.

percentage varying from 0% to 5% [2,3]. The obtained results are discussed separately according to the scenario.

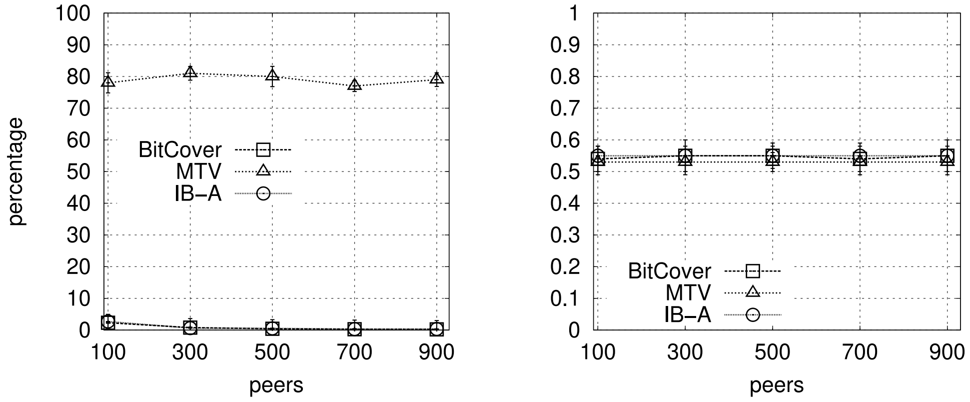
(a) Analysis of Scenario 1

Fig. 5 shows the computed results for the performance metrics in function of n. We have that D_R has adequate values, i.e., above the video-encoding rate (i.e., 20 Mbps), regardless of the packet-loss percentage. D_T presents adequate values too, i.e., below the maximum limit of 10 s, but only when the packet-loss percentage is limited up to 1%. At last, both P_R and S_O have adequate values, since they are within the defined limits for a satisfactory system performance: $P_R \approx$ 55%, and $S_O \leq$ 3,4%. In fact, these last two metrics are not affected by packet losses, since they do not take into account duplicate requests due to retries. The overall conclusion is that BitCover thus has an adequate performance.



(a) Download Rate.

(b) Discontinuity Time.



(c) Initial Seed Overloading.

(d) Piece Rate.

Fig. 9. Comparative analysis for scenario 1.

(b) Analysis of Scenario 2

Fig. 6 depicts the computed results for the performance metrics in function of *n*. The observations that follow are similar to those already discussed in the case of Scenario 1. Regardless of the packet-loss percentage, the values of D_R are adequate, i.e., above the video-encoding rate, to provide a satisfactory service. The values of D_T are below the limit of 10 s, even though they now increase according with the packet-loss percentage. At last, the values of both S_O and P_R are within the corresponding limits (3.4% for S_O) for a satisfactory service. The overall conclusion is again that BitCover therefore has an adequate performance.

5.5.3. Degradation due to traffic increase on WiFi-Direct channel

We herein aim to assess the performance degradation of BitCover when the traffic is increased on the WiFi-Direct channel. To this end, we decrease the area over which the peers move. Recall that BitCover peer opts to deploy WiFi-Direct based on the radio coverage distance, considering 100 m as the threshold. We analyze square areas with a side length, *R*, ranging from 100 m to 500 m. Moreover, we assume $k = 3$ and a 4%-packet-loss percentage within Scenarios 1 and 2, respectively. Table 10 presents the percentage of packets delivered via WiFi-Direct. These experimental results are discussed in the following.

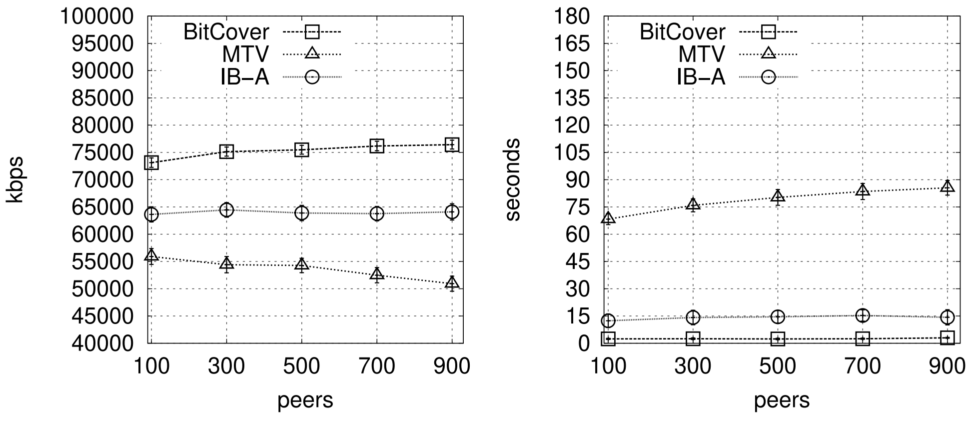
Fig. 7 shows the computed values under Scenario 1. See that D_R presents values above the minimum required limit to provide a satisfactory service, regardless of *R*. For D_T, only at $R = 500$ m, we may achieve a satisfactory service, which clearly deteriorates as more delivers occur via WiFi-Direct. This is because the WiFi channel is unreliable and error prone, yielding more duplicate packets and, thereby, consuming more resources. The other two metrics maintain similar behavior as in the previous experiments since they are not affected by the time spent delivering duplicate packets.

The results concerning Scenario 2 are depicted in Fig. 8. The general observation is quite similar to that of the previous scenario, with a better performance for the D_T and P_R metrics. Particularly, all values of D_T provide a satisfactory service, except at R =100 m. The overall conclusion is that BitCover has an adequate performance for both scenarios.

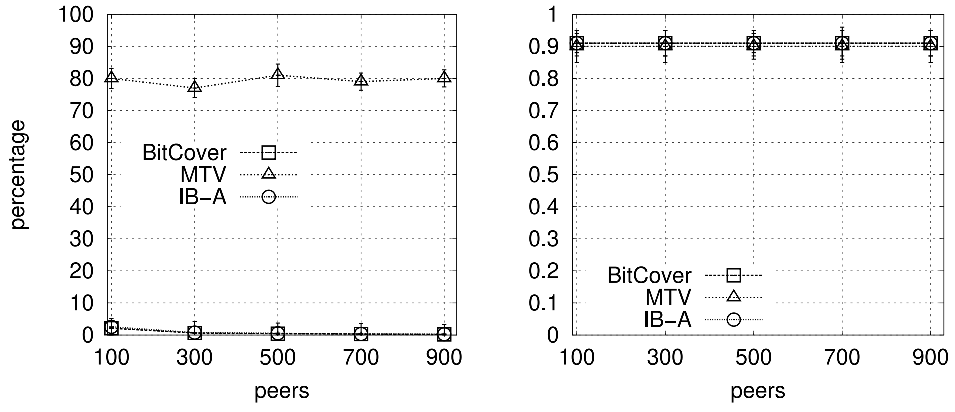
5.5.4. Competitive analysis

The experiments to follow competitively compare BitCover against IB-A [25] and MTV [20] proposals, which are succinctly described in Table 11. We choose IB-A because it is the most recent and efficient BitTorrent's adaptation targeted at interactive streaming with mobile nodes. By its turn, MTV is chosen because of being a BitTorrent-based proposal focused on interactive-streaming over 5G that also deploys WiFi-Direct, as already discussed in Section 3.

The computed results for all metrics are presented in Fig. 9 and 10, where each of them refers to one of the scenarios. Notwithstanding, for the sake of clarity, we mention that we divide the discussion of the numeric results in accordance with two different views. The first delves into an algorithm-design analysis, in which we compare the proposals by outlining their main design differences to possibly explain why and how their corresponding performances differ. The second carries out a more quantitative evaluation, in which the main goal is to outline the numerical optimization achieved by BitCover with respect to the two other proposals.



(a) Download Rate. (b) Discontinuity Time.



(c) Initial Seed Overloading. (d) Piece Rate.

Fig. 10. Comparative analysis for Scenario 2.

(a) Algorithm-design analysis

Consider the IB-A proposal and the results depicted in Figs. 9 and 10. Given that all peers have a large and not saturated upload-capacity, it then comes that the upload-rate criterion of IB-A is not as relevant as the piece-coverage criterion of BitCover.

Additionally, the *indirect reciprocity* criterion deployed by IB-A does not produce any performance optimization. This is a consequence of the δt parameter's configuration, which is not adherent to cellular networks. Regarding the network size, both BitCover and IB-A own a limited neighborhood of up to 80 peers, in accordance to the original BitTorrent protocol. This implies that both BitCover and IB-A are similarly affected with respect to their performance metrics for $n \geq 80$. Finally, as for the client's interactivity, these two proposals differ in performance due to their respective criteria used for piece selection.

Now, consider the MTV proposal and the same aforementioned figures. The main issue is the overloaded streaming server of MTV. This server cannot meet the high demand of requests, since almost 80% of them may be sent directly to it via 5G (please see Table 10). This is different in BitCover, once the requests are distributed among the peers.

Furthermore, the larger the network size is, the better BitCover's performance becomes. This happens because it is easier to find a peer that shares video pieces. We though highlight this improvement is not proportional to n, since it is limited to the size of the neighborhood, as already mentioned. In turn, MTV performance slightly decreases. This happens because the more peers are in the network, the more requests must be processed by the server. At last, under the low-interactivity profile, we have better results than under the high-interactivity one in both proposals.

Table 12
Optimizations achieved by BitCover to MTV.

Scenario	1		2	
Metric	Min	Max	Min	Max
D_R	28.5 ± 1.4%	33.1 ± 2.4%	23.5 ± 1.2%	33.4 ± 2.3%
D_T	88.1 ± 0.7%	90.0 ± 1.1%	96.4 ± 0.3%	97.2 ± 0.8%
S_O	97.1 ± 0.3%	99.7 ± 0.2%	97.6 ± 0.4%	99.8 ± 0.1%
P_R	2.2 ± 0.2%	2.5 ± 0.3%	1.0 ± 0.1%	1.1 ± 0.2%

Table 13
Optimizations achieved by BitCover to IB-A.

Scenario	1		2	
Metric	Min	Max	Min	Max
D_R	5.9 ± 0.9%	9.8 ± 0.6%	12.9 ± 1.2%	16.3 ± 1.7%
D_T	52.6 ± 4.1%	66.5 ± 2.4%	79.0 ± 1.4%	83.7 ± 1.3%
S_O	−0.4 ± 0.2%	1.8 ± 0.5%	0.1 ± 0.1%	2.2 ± 0.6%
P_R	0.2 ± 0.1%	0.5 ± 0.2%	0.0 ± 0.1%	0.0 ± 0.1%

(b) Quantitative evaluation

Using the results depicted in the aforementioned figures, we mount Tables 12 and 13. The goal herein is to highlight the maximum and minimum optimizations (i.e., gains at D_R and P_R, besides reductions at D_T and S_O) achieved by BitCover with respect to MTV and IB-A.

Comparing to MTV, we observe that BitCover provides optimizations for all metrics. Nonetheless, the advantage is not that significant specifically at P_R. The rationale for this last is that MTV's predefined windows for piece-selection policy perform quite similarly to the k

divisions defined in BitCover. Notwithstanding, the sequential-request piece selection used in Bitcover outperforms the random one used in MTV.

Now, comparing to IB-A, we observe that BitCover provides optimizations exclusively for D_R and D_T, presenting quite similar values at the other two metrics. We mention that the negative values refer to when IB-A is better than Bitcover. This happens for small-sized networks, but these values may be judged as insignificant, i.e., less than 1%.

It is important to mention that BitCover uses a layered architecture, in which the peer-selection and piece-selection policies are strictly confined to the application layer. This layered architecture may be easily implemented and executed independently of network and physical topologies, which are in the lower layers. Simplicity is the main advantage.

6. Conclusions and future work

This paper proposed the BitCover algorithm, which stands for a BitTorrent enhancement for VoD streaming over 5G cellular networks. This proposal is based on exchanging video pieces using 5G and WiFi-Direct channels, besides deploying a novel peer-selection policy which improves communication-channel usage by prioritizing to select peers that own more downloaded video pieces in their local buffers.

The experiments were based on simulations under *streaming* scenarios featured by different network sizes, user interactivity-profiles, and video sizes. We competitively compared BitCover with two other recent proposals, namely MTV and IB-A. The overall attractive effectiveness of our proposal could be verified by means of different performance metrics, whose computed experimental results are below the imposed limits in the literature concerning suitable VoD streaming services. For instance, a file encoding of 20 Mbps was guaranteed, the total accumulated discontinuity time for video delivery was less than 10 s, and the percentage of piece requests (over the total piece requests from all peers) that is handled by the initial server was limited up to 3.4%, which may be judged as adequate performance results [40,43–46].

Based on the above, we concluded that BitCover may play a decisive role to optimize the bandwidth usage within each 5G cell site and, hence, certainly help to leverage the entire potential of the 5G technology to provide video-streaming service with adequate system QoS and QoE on the client side.

Finally, as an extension of this work and being aware of our modeling limitations, we think of the following future directions. First, we want to compare BitCover against other algorithmic solutions via measurements in real-world scenarios, including devices' energy consumption and efficiency of several different setups. Second, we consider to further exploit the WiFi-Direct technology besides 5G to propose other P2P-based algorithms than BitCover for multimedia streaming service on cellular networks. Finally, due to the expected dense deployment of base stations and the complicated associations between base stations and end-to-end network slices in 5G heterogeneous networks (HetNets) [48,49], we also plan to look into the peers' handoff process by including it in our modeling for new evaluation experiments with BitCover.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All data was generated by external open simulators and their parameters are specified in the text.

References

[1] A. Al-Habashna, G. Wainer, Improving video transmission in cellular networks with cached and segmented video download algorithms, Mob. Netw. Appl. 23 (3) (2018) 543–559, http://dx.doi.org/10.1007/s11036-017-0906-x.

[2] D. Xu, et al., 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, New York, NY, USA, 2020, pp. 479–494, http://dx.doi.org/10.1145/3387514.3405882.

[3] A. Narayanan, et al., A first look at commercial 5G performance on smartphones, in: Proceedings of the Web Conference 2020, New York, NY, USA, 2020, pp. 894–905, http://dx.doi.org/10.1145/3366423.3380169.

[4] L. Shrama, A. Javali, S.K. Routray, An overview of high speed streaming in 5G, in: 2020 International Conference on Inventive Computation Technologies, ICICT, 2020, pp. 557–562, http://dx.doi.org/10.1109/ICICT48043.2020.9112489.

[5] U. Abbasi, H. Elbiaze, Multimedia streaming using D2D in 5G ultra dense networks, in: 15th IEEE Annual Consumer Communications Networking Conference, CCNC, 2018, pp. 1–6, http://dx.doi.org/10.1109/CCNC.2018.8319230.

[6] A. Al-Habashna, G. Wainer, Improving video transmission in cellular networks with cached and segmented video download algorithms, Mob. Netw. Appl. 23 (3) (2018) 543–559, http://dx.doi.org/10.1007/s11036-017-0906-x.

[7] A. Al-Habashna, G. Wainer, Dash-based device-to-device video streaming for cellular networks with high user density, in: 2018 International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 2018, pp. 1–12, http://dx.doi.org/10.1109/SPECTS.2018.8574197.

[8] Y.-J. Yu, A.-C. Pang, M.-Y. Yeh, Video encoding adaptation for QoE maximization over 5G cellular networks, J. Netw. Comput. Appl. 114 (2018) 98–107, http://dx.doi.org/10.1016/j.jnca.2018.04.008.

[9] B. Cohen, Incentives build robustness in BitTorrent, in: First Workshop on Economics of Peer-To-Peer System, Berkeley, USA, 2003, URL https://www.bittorrent.org/bittorrentecon.pdf.

[10] A. Legout, G. Urvoy-Keller, P. Michiardi, Rarest first and choke algorithms are enough, in: IMC 2006, ACM SIGCOMM/USENIX Conference, Rio de Janeiro, RJ, Brazil, 2006, URL http://conferences.sigcomm.org/imc/2006/papers/p20-legout.pdf.

[11] Wikipedia, The WiFi-Direct technology, 2021, URL https://en.wikipedia.org/wiki/Wi-Fi_Direct.

[12] N. Adam, C. Tapparello, W. Heinzelman, Performance evaluation of WiFi direct multi hop ad-hoc networks, in: 2020 International Conference on Computing, Networking and Communications, ICNC, 2020, pp. 661–666, http://dx.doi.org/10.1109/ICNC47757.2020.9049743.

[13] Wikipedia, The 5G technology, 2021, URL https://en.wikipedia.org/wiki/5G.

[14] B. Jedari, et al., Video caching, analytics, and delivery at the wireless edge: A survey and future directions, IEEE Commun. Surv. Tutor. 23 (1) (2021) 431–471, http://dx.doi.org/10.1109/COMST.2020.3035427.

[15] F. Chiariotti, A survey on 360-degree video: Coding, quality of experience and streaming, Comput. Commun. 177 (2021) 133–155, http://dx.doi.org/10.1016/j.comcom.2021.06.029.

[16] M. Wang, C. Xu, S. Jia, G.-M. Muntean, Video streaming distribution over mobile Internet: A survey, Front. Comput. Sci. 12 (2018) 1039–1059, http://dx.doi.org/10.1007/s11704-018-7153-6.

[17] J. Park, J. Lee, J. Lee, M.-S. Lee, Enabling 5G techniques to support HD video streaming to high speed train users, in: 2020 International Conference on Information and Communication Technology Convergence, ICTC, 2020, pp. 875–878, http://dx.doi.org/10.1109/ICTC49870.2020.9289626.

[18] J. Qiao, Y. He, X.S. Shen, Improving video streaming quality in 5G enabled vehicular networks, IEEE Wirel. Commun. 25 (2) (2018) 133–139, http://dx.doi.org/10.1109/MWC.2018.1700173.

[19] L.M. Castaneda Herrera, A. Duque Torres, W.Y. Campo Munoz, An approach based on knowledge-defined networking for identifying video streaming flows in 5G networks, IEEE Lat. Am. Trans. 19 (10) (2021) 1737–1744, http://dx.doi.org/10.1109/TLA.2021.9477274.

[20] Y. Yang, X. Zhang, L. Zhang, MTV: Mobile BitTorrent video sharing using harmonized LTE and WiFi coexistence, in: 2017 IEEE 85th Vehicular Technology Conference, VTC Spring, 2017, pp. 1–5, http://dx.doi.org/10.1109/VTCSpring.2017.8108300.

[21] A. Al-Habashna, S. Fernandes, G. Wainer, Analyzing the effect of LTE-A transmission parameters on video streaming quality of experience, in: Proceedings of the Communications and Networking Symposium, 2018, URL https://dl.acm.org/doi/10.5555/3213200.3213203.

[22] A. Polakovič, R. Vargic, G. Rozinaj, Adaptive multimedia content delivery in 5G networks using DASH and saliency information, in: 2018 25th International Conference on Systems, Signals and Image Processing, IWSSIP, 2018, pp. 1–5, http://dx.doi.org/10.1109/IWSSIP.2018.8439215.

[23] H. Kim, K. Chung, Multipath-based HTTP adaptive streaming scheme for the 5G network, IEEE Access 8 (2020) http://dx.doi.org/10.1109/ACCESS.2020.3038854.

[24] A. Ghani, et al., An adaptive video streaming framework for peer-to-peer 5G networks: Paving the road to 5G-IMS, in: M. Ben Ahmed, S. Mellouli, L. Braganca, B. Anouar Abdelhakim, K.A. Bernadetta (Eds.), Emerging Trends in ICT for Sustainable Development, Springer International Publishing, Cham, 2021, pp. 209–218, URL https://link.springer.com/chapter/10.1007/978-3-030-53440-0_23.

[25] C.K. da S. Rodrigues, V. Rocha, Enhancing BitTorrent for efficient interactive video-on-demand streaming over MANETs, J. Netw. Comput. Appl. 174 (2021) http://dx.doi.org/10.1016/j.jnca.2020.102906.

[26] L. Baresi, N. Derakhshan, S. Guinea, WiDiSi: A Wi-Fi direct simulator, in: IEEE Wireless Communications and Networking Conference, Doha, Qatar, 2016, http://dx.doi.org/10.1109/WCNC.2016.7565169.

[27] Naser Derakhshan, A WiFi direct simulator, 2016, URL https://github.com/nasser1941/WiDiSi.

[28] D. Schoonwinkel, Practical Measurements of Wi-Fi Direct in Content Sharing, Social Gaming Android Applications (Ph.D. thesis), Stellenbosch: Stellenbosch University, 2016.

[29] R.M. Mbala, J.M. Nlong, J.-R.K. Kamdjoug, A framework for multi-hop ad-hoc networking over Wi-Fi direct with android smart devices, Commun. Netw. 13 (4) (2021) 143–158.

[30] A. Munjal, et al., SMOOTH: A simple way to model human mobility, 2011, [Online]. Available at: https://www.eurecom.fr/spyropou/papers/Smooth-Infocom2011.pdf.

[31] N. Aschenbruck, et al., BonnMotion: A mobility scenario generation and analysis tool, in: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, Brussels, Belgium, 2010, pp. 51:1–51:10, http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8684.

[32] A. Munjal, T. Camp, W.C. Navidi, 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, ACM, New York, NY, USA, 2011, pp. 351–360, http://dx.doi.org/10.1145/2068897.2068957.

[33] E.L. Abram-Profeta, K.G. Shin, Providing unrestricted VCR functions in multicast video-on-demand servers, in: Proceedings. IEEE International Conference on Multimedia Computing and Systems, 1998, pp. 66–75, http://dx.doi.org/10.1109/MMCS.1998.693626.

[34] YouTube Help, Choose live encoder settings, bitrates, and resolutions, 2021, URL https://support.google.com/youtube/answer/2853702.

[35] C.K.S. Rodrigues, M.V.M. Rocha, A dispersão de dados como critério para a política de seleção de peers em uma rede BitTorret para streaming sob demanda interativo, Revista De Sistemas E Computação 7 (1) (2017) 36–43, URL https://revistas.unifacs.br/index.php/rsc/article/viewFile/4702/3187.

[36] M. Rocha, M. Maia, Í. Cunha, J. Almeida, S. Campos, Scalable media streaming to interactive users, in: 13th Annual ACM International Conference on Multimedia, Hilton, Singapore, 2005, URL https://dl.acm.org/doi/10.1145/1101149.1101351.

[37] Vuze, Torrent piece size, 2021, URL https://wiki.vuze.com/w/Torrent_Piece_Size.

[38] B. Jones, How to make the best torrents, 2021, URL https://torrentfreak.com/how-to-make-the-best-torrents-081121/.

[39] A. Montresor, M. Jelasity, PeerSim: A scalable P2P simulator, in: IEEE Ninth International Conference on Peer-To-Peer Computing, Washington, USA, 2009, URL https://ieeexplore.ieee.org/document/5284506.

[40] B. Wang, J. Kurose, P. Shenoy, D. Towsley, Multimedia streaming via TCP: An analytic performance study, ACM Trans. Multimedia Comput. Commun. Appl. 4 (2) (2008) http://dx.doi.org/10.1145/1352012.1352020.

[41] D. Soldani, A. Manzalini, Horizon 2020 and beyond: On the 5G operating system for a true digital society, IEEE Veh. Technol. Mag. 10 (1) (2015) 32–42, http://dx.doi.org/10.1109/MVT.2014.2380581.

[42] V.D. Bhamidipati, S. Kilari, Effect of delay/delay variation on QoE in video streaming, 2010, Blekinge Institute of Technology, URL https://www.essays.se/essay/f2b45b41cb/.

[43] R. Huysegems, J. van der Hooft, T. Bostoen, P. Rondao Alface, S. Petrangeli, T. Wauters, F. De Turck, HTTP/2-Based methods to improve the live experience of adaptive streaming, in: Proceedings of the 23rd ACM International Conference on Multimedia, MM '15, ACM, New York, NY, USA, 2015, pp. 541–550, http://dx.doi.org/10.1145/2733373.2806264.

[44] L. Sun, Y. Mao, T. Zong, Y. Liu, Y. Wang, Flocking-based live streaming of 360-degree video, in: Proceedings of the 11th ACM Multimedia Systems Conference, MMSys '20, ACM, New York, NY, USA, 2020, pp. 26–37, http://dx.doi.org/10.1145/3339825.3391856.

[45] I. Gkortsilas, K. Deltouzos, M. Efthymiopoulou, N. Efthymiopoulos, S. Denazis, Liquidstream — A high performance and stable scheduling architecture for P2P video on demand, in: 2012 International Conference on Telecommunications and Multimedia, TEMU, 2012, pp. 89–94, http://dx.doi.org/10.1109/TEMU.2012.6294739.

[46] P.J. Braun, et al., Mobile peer-to-peer assisted coded streaming, IEEE Access 7 (2019) 159332–159346, http://dx.doi.org/10.1109/ACCESS.2019.2950800.

[47] Wikipedia, Bittorrent Protocol Specification v1.0, 2021, URL https://wiki.theory.org/BitTorrentSpecification#Tracker_Response.

[48] W. Wu, F. Yang, Y. Gao, X. Wang, P. Si, Y. Zhang, F.R. Yu, Distributed handoff problem in heterogeneous networks with end-to-end network slicing: Decentralized Markov decision process-based modeling and solution, IEEE Trans. Wireless Commun. (2022) 1, http://dx.doi.org/10.1109/TWC.2022.3190883.

[49] E. Ramadan, A. Narayanan, U.K. Dayalan, R.A.K. Fezeu, F. Qian, Z.-L. Zhang, Case for 5G-aware video streaming applications, in: Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases, in: 5G-MeMU '21, New York, NY, USA, 2021, pp. 27–34, http://dx.doi.org/10.1145/3472771.3474036.

Vladimir Rocha is a Computer Engineering and holds a Ph.D.'s degree in Computer Engineering and master's degree in Computer Science both at the University of São Paulo. Specialist in Peer-to-Peer and Cloud Computing technologies works as a professor at the Center for Mathematics, Computation and Cognition department of the Federal University of ABC — Brazil. His research interests include scalability and performance in distributed structures and systems, such as BitTorrent, DHT and Blockchain.

Carlo Kleber da Silva Rodrigues received the D.Sc. degree in Systems and Computer Engineering from the Federal University of Rio de Janeiro in 2006, the M.Sc. degree in Systems and Computing from the Military Institute of Engineering in 2000, and the B.Sc. degree in Electrical Engineering from the Federal University of Paraiba in 1993. Currently, he is a Professor in the Center for Mathematics, Computation, and Cognition at the Federal University of ABC, working in the subarea of Computer Networks. His research interests include Internet Multimedia Content Distribution (Streaming), Cryptocurrency Analysis (Bitcoin and IOTA), Distributed Ledger Technologies (DLTs — Blockchain and Tangle), and Mobile Wireless Networks.