Avgouleas *et al. J Wireless Com Network* (2021) 2021:69
https://doi.org/10.1186/s13638-021-01935-1

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A wireless caching helper system with heterogeneous traffic and random availability

Ioannis Avgouleas[1*], Nikolaos Pappas[2] and Vangelis Angelakis[2]

*Correspondence: ioannis.avgouleas@liu.se
[1] Department of Computer and Information Science (IDA), Linköping University, 58183 Norrköping, Sweden
Full list of author information is available at the end of the article

Abstract

Multimedia content streaming from Internet-based sources emerges as one of the most demanded services by wireless users. In order to alleviate excessive traffic due to multimedia content transmission, many architectures (e.g., small cells, femtocells, etc.) have been proposed to offload such traffic to the nearest (or strongest) access point also called "helper". However, the deployment of more helpers is not necessarily beneficial due to their potential of increasing interference. In this work, we evaluate a wireless system which can serve both cacheable and non-cacheable traffic. More specifically, we consider a general system in which a wireless user with limited cache storage requests cacheable content from a data center that can be directly accessed through a base station. The user can be assisted by a pair of wireless helpers that exchange non-cacheable content as well. Files not available from the helpers are transmitted by the base station. We analyze the system throughput and the delay experienced by the cached user and show how these performance metrics are affected by the packet arrival rate at the source helper, the availability of caching helpers, the caches' parameters, and the user's request rate by means of numerical results.

Keywords: Wireless communication, Wireless caching, Mutiple caching helpers, Performance evaluation, Queueing analysis, Non-cacheable traffic, Cacheable traffic, Throughput, Delay

1 Introduction

Wireless video has been one of the main generators of wireless data traffic. It is expected to originate 75% of the global mobile traffic by 2020 [1] and inevitably contribute to networks' congestion and delays. One of the most promising technologies to cope with such issues is caching popular files in helper nodes that constitute a wireless distributed caching network that assists base stations by handling requests for popular files [2, 3].

Wireless caching helpers can store a number of popular files and transmit them to the requesting users more efficiently, considering that helpers have been deployed in such a way that the wireless channel between helpers and users is more efficient than the one between users and base stations. Wireless networks with caching capabilities can significantly reduce cellular traffic and delay as well as simultaneously increase throughput [4, 5].

Springer Open

© The Author(s) 2021. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

In this paper, we study a wireless system that serves heterogeneous traffic which we distinguish between two non-overlapping classes: (i) cacheable and (ii) non-cacheable traffic. The former originates from content that is promising to cache because it is frequently requested, e.g., popular movies, trending music tracks, static parts of web pages, etc. On the other hand, non-cacheable traffic consists of content that is unlikely to be frequently requested such as chat messages or dynamic parts of web pages, and thus, it is not sensible to cache. A user with limited cache storage requests cacheable content from a data center using a base station which has direct access to it through a backhaul link. Two wireless nodes within the proximity of the user exchange non-cacheable content and have limited cache storage. Therefore, they can act as caching helpers for the cached user by serving its requests for cacheable content when they do not exchange non-cacheable traffic. Files not available at the helpers can be fetched by data center through the base station. Additionally, the source helper is equipped with a queue whose role is to save excessive packets of non-cacheable traffic with the intention of transmitting them to the destination helper in a subsequent time slot. Concerning caching, we assume the content placement is given and hierarchical.

1.1 Related work

Various content placement strategies have been studied in scientific literature, e.g., caching the most popular content everywhere [2], probabilistic caching [6, 7], cooperative caching [8–12], or caching based on location, e.g., geographical caching [13].

Additionally, several different performance metrics have been considered. In earlier studies of wireless caching, cache hit probability (or ratio) [13], and the density of successful receptions or cache-server requests [6, 13] have been commonly investigated as a means of evaluating the performance of wireless caching systems. Furthermore, there are several studies regarding energy efficiency or consumption of the different caching schemes [13–16] as well as taking into account the traffic load of the wireless links [17, 18]. Methods that reduce traffic load by optimizing the offloading probability or gain can be found in [19–21].

More recently, a considerable amount of research works analyze wireless caching systems by considering throughput [7, 8] and/or delay [22]. Regarding the latter, the majority of research works cope with mitigating the backhaul or transmission delay under the assumption that traffic or requests are saturated. However, there are works that take into account stochastic arrivals of requests at different nodes [23, 24].

Caching has been applied to several different network realizations, e.g., FemtoCaching [3] in which the so-called femto base station (FBS) serve a group of dedicated users with random content requests while simultaneously the non-dedicated users might be served with delay due to cache misses or no FBS availability. The coded/uncoded cached contents are stored in multiple small cells, the so-called femtocells. Given the file requests distribution and the cache size of each femtocell, the content placement is studied such that the downloading time is minimized.

The advent of vehicular networks necessitates the use of caches to reduce the latency of content streaming and increase the offered quality of service (QoS) [25, 26]. Supporting vehicle-to-everything connections urges the exploration of alternative data routing protocols in order to avoid incurring excessive end-to-end delay and backhaul resources

allocation. On the contrary, moving computational and storage resources to the mobile edge computing seems encouraging [27–29]. This can be done, e.g., by employing a new paradigm known as local area data network [30], or other advances in radio access networks (RANs) for Internet of Things (IoT) [31].

Many contemporary works consider to jointly optimize the problems of content caching (or placement), computing, and allocating radio resources. They usually consider and solve separately these important issues by formulating the computation offloading or content caching as convex optimization problems with different metrics, e.g., service latency, network capacity, backhaul rate etc. [32, 33]. Works that simultaneously address the aforementioned problems together and propose a joint optimization solution for the fog-enabled IoT or cloud RANs (C-RANs) can be found in [34, 35], respectively.

For some applications, e.g., broadcast or multicast applications, single transmissions from the base station to more than one user are useful. The authors in [36] propose a content caching and distribution scheme for smart grid enabled heterogeneous networks, in which each popular file is stored in multiple service nodes with energy harvesting capabilities. The optimization of the total on-grid power consumption, the user association scheme, and the radio resource allocation improves the reliability and performance of the wireless access network. The evolution of 5G mobile networks is going to incorporate cloud computing technologies. The authors in [37] propose the concept of "Caching-as-a-Service" (CaaS) based on C-RANS as a means to cache anything, anytime, and anywhere in the cloud-based 5G mobile networks with the intention of satisfying user demands from any service location with high QoS. Furthermore, they discuss the technical details of virtualization, optimization, applications, and services of CaaS in 5G mobile networks.

A key distinction among research papers in wireless caching is the assumption regarding the availability of caching helpers. Many papers consider that caching helpers can serve users requests whenever the requested file is cached while others adopt the assumption that caching helpers might be unable to assist user requests when, for example, serve other users of interest [3, 13]. To the best of our knowledge, the proposed wireless caching model has not been studied in the literature. For instance, [8] does not take into account hierarchical caching even if it serves both two types of traffic with the assistance of one caching helper.

1.2 Contribution

In this paper, we study a wireless system in which we distinguish traffic between cacheable and non-cacheable[1]. When a cached user experiences a local cache miss, it requests cacheable content from a data center connected to a base station through a backhaul link. Two wireless nodes within the user's proximity exchange non-cacheable files and have limited cache storage. Therefore, they can act as caching helpers for the cached user by serving its requests for cacheable content when they do not exchange non-cacheable content for their own purposes. The source helper is equipped with an infinite queue whose role is to save packets of non-cacheable traffic for transmission to the destination

[1] This work was partly presented in [38].

helper in a subsequent time slot. Files not available at the helpers can be transmitted by the base station.

We analyze the system throughput assuming that transmitting nodes have random access to the channel and, hence, the probabilities by which the caching helpers are available can be tuned. By adapting the availability of the caching helpers, we want to guarantee that user D will be served with non-cacheable traffic according to specific requirements, i.e., stability in our case. First, we characterize the system throughput concerning the case in which the queue at the source helper is stable as well as unstable. Moreover, we formulate a mathematical optimization problem to optimize the probabilities by which the helpers are available to assist the cached user to maximize the system throughput. Subsequently, we characterize the average delay experienced by the user from the time of a local cache miss until it receives the requested cached file. Finally, we provide numerical results to show how the packet arrival rate of non-cacheable traffic at the source helper, the availability of caching helpers, random access to the channel, caching parameters, and the user's request rate affect the system throughput and the delay.

1.3 Organization of the paper

In Sect. 2, we present the system model comprising the network, the caching, the transmission, and the physical layer model. Section 3 provides the analytical derivation of throughput for the cases of stable and the unstable queue at the source helper. The average delay performance is given in Sect. 4. In Sect. 5, we numerically evaluate our theoretical analysis of the previous sections and summarize the results. Finally, Sect. 6 concludes our research work.

2 System model

2.1 Network model

We consider a network system with four wireless nodes: a pair of caching helpers S and D, a random user U within the coverage of the helpers and a base station (BS) node connected to a datacenter (DC) through a backhaul link as depicted in Fig. 1. We consider slotted time and that a packet transmission takes one time slot.

Helper S is equipped with an infinite queue Q and the packet arrivals follow a Bernoulli process with average arrival rate λ. It transmits packets to the destination helper D. In each time slot, user U requests for a file in its own cache. In case U's cache miss, which happens with probability q_U, it requests the file from external resources, i.e., the caching helpers or the data center (through the BS). The data center stores the whole library and, hence, every file that U may request.

Requesting a file directly from the BS is not necessarily the best policy since the link connecting BS and U might be problematic. Consequently, limited throughput or increased delay might be experienced instead of fetching the file from one of the caching helpers. Moreover, the BS is not always available to help U; this happens with probability α in each time slot. Therefore, it is preferable to U when it is served by the caching helpers.

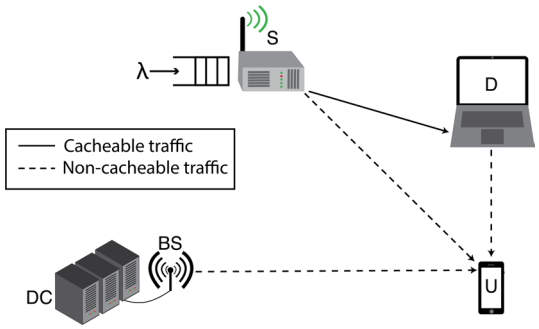

Fig. 1 An example of our system model. Caching helpers *S* and *D* can be access points with storage capabilities. Node *S* has a dedicated connected user *D*, who is randomly generating requests for non-cacheable content. At the same time, there is a mobile device *U* within both helpers' proximity, which requests cached content from external resources with some probability in each time slot. Device *U* also has access to the data center *DC* through the *BS*, but the connection can be problematic, so it is preferred to be served by *D* or *S*, when possible. Please note that interference to the transmitters is not depicted

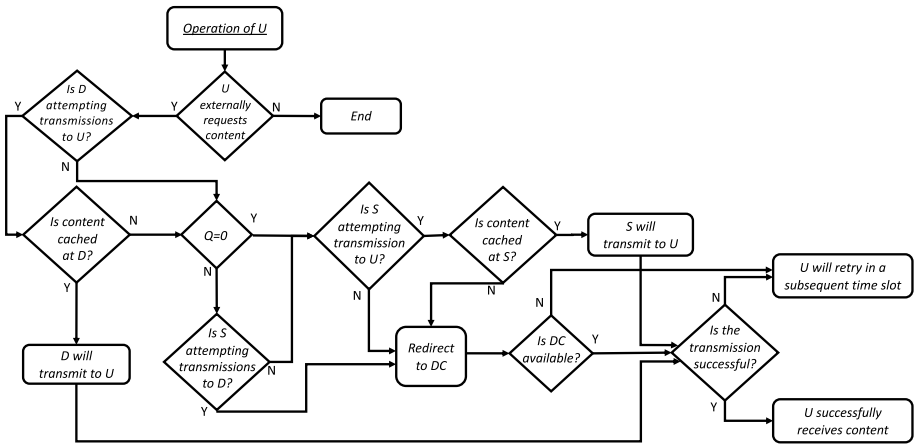

Fig. 2 Operation of *U* in the described protocol

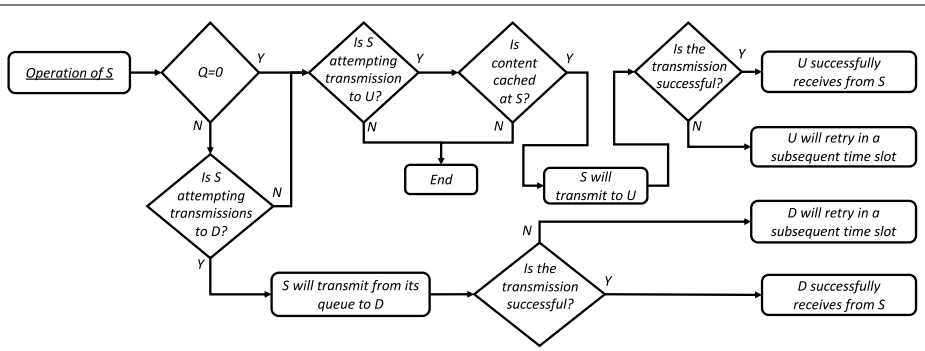

Fig. 3 Operation of *S* when user *U* requests a file *f* from external resources

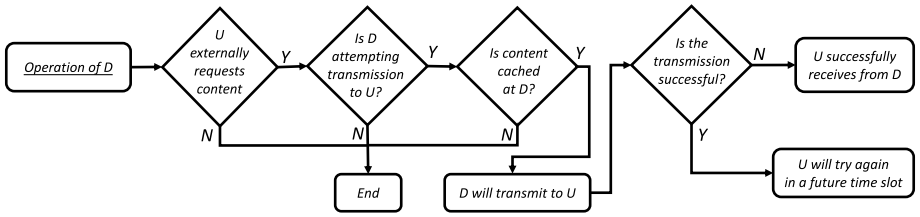

Fig. 4 Operation of *D* when user *U* requests a file *f* from external resources

The flowchart of user U's operation with respect to its request content search is shown in Fig. 2. The operations of caching helpers S and D, represented as flowcharts, are depicted in Figs. 3 and 4, respectively.

2.2 Cache placement and access

We assume the content placement is given and hierarchical, i.e., when the user node requests for a file that is not stored in its most popular files, it first probes the closest caching helper which stores the next most popular files. If this probe fails, then the second caching helper is probed for the requested file. If it also cache misses, then the file can be found in the data center. Additionally, the source helper is equipped with a queue whose role is to save the excessive non-cacheable traffic with the intention of transmitting it to the destination helper in a subsequent time slot.

Furthermore, the user device U and the caching helpers D and S have cache capacity to M_U, M_D, and M_S files, respectively, and $M_U \leq M_D \leq M_S$ holds. We also consider the collaborative most popular content (CMPC) policy. According to CMPC, user U stores the first M_U most popular files in its own cache, helper D stores the next most M_D popular files, and S stores the next most M_S popular files. Following CMPC requires exchange of information among devices, e.g., the cache size of each device and the content placement in each device. We assume that this information exchange is negligible.

2.3 Transmission model

In each time slot, S will attempt transmission of non-cacheable content to D with probability q_S (if its' queue is not empty) and is available for U with probability $1 - q_S$. We assume that the caching helpers assist U only when specific conditions apply: D will attempt transmission to U with probability q_D, and S will help U only when it is not transmitting to D. When the source caching helper S is transmitting to helper D and the user U requests a file from external resources, then U can be served by D or by DC. In that case, there are two parallel transmissions one from S to D and one from D (or DC, respectively) to U. If the caching helper S is available for U, then there are no parallel transmissions since only one of S, D, or DC can help U at the same time slot.

Regarding DC, we model its availability with a probability α to model the fact that it is not always available to serve U due to serving other users or failure. If the DC is always available to U, then $\alpha = 1$. On the other hand, if the DC is not available for U, then $\alpha = 0$

.

We summarize the aforementioned events and notation in Table 1. Additionally, the operation of U, S, and D as flowcharts can be found in Figs. 2, 3, and 4.

Table 1 Notation table

Notation	Description
q_U	Probability of U requesting a file from external resources, i.e., D, S, or the DC
p_{hD}	Probability of cache hit at D
p_{hS}	Probability of cache hit at S
q_S	Probability of S being available for D
q_C	Probability of S being available for U
q_D	Probability of D being available for U
α	Probability of DC being available for U
$P_{i\to j}$	Success probability of link $i \to j$, when i is transmitting
$P_{i\to j/k}$	Success probability of link $i \to j$, when i and k are transmitting
$\mathsf{P}(i \Rightarrow j)$	Probability of queued node i attempting transmissions to j

2.4 Physical layer model

The wireless channel is modeled as Rayleigh flat-fading channel with additive white Gaussian noise. A packet transmitted by i is successfully received by j if and only if the signal-to-interference-plus-noise (SINR) between i and j exceeds a minimal threshold θ. Let $P_{tx}(i)$ be the power measured at 1 m distance from the transmitting node i, and $r(i, j)$ be the distance in m between i and j . Then, the power received by j when i transmits is $P_{rx}(i,j) = A(i,j)h(i,j)$, where $A(i, j)$ is a unit-mean exponentially distributed random variable. The receiver power factor $h(i, j)$ is given by $h(i,j) = P_{tx}(i)(r(i,j))^{-p}$, where $p \in [2, 7]$ is the path loss exponent. Self-interference is modeled using the self-interference coefficient $g \in [0, 1]$. The success probability in link (ij) is given by [39]:

$$P_{i\to j/\mathcal{T}} = \exp\Big(-\frac{\theta \mathrm{n_j}}{\mathrm{h(i,j)}}\Big)(1+\theta \mathrm{r(i,j)^p g})^{-\mathrm{l}} \times \prod_{k\in\mathcal{T}\setminus\{i,j\}} \Big(1+\frac{\theta h(k,j)}{h(i,j)}\Big)^{-1},$$

with T denoting the set of transmitting nodes at the same time, n_j denoting the noise power at j, and $l = 1$ when $j \in T$ and $l = 0$ otherwise.

3 Throughput analysis

In this section, we analyze the throughput of the system depicted in Fig. 1 . We are interested in the weighted sum of the throughput that helper S provides D along with the throughput realized by the cached user U. By denoting the former with T_S and the latter with T_U, the weighted sum throughput T_w is given by:

$$T_w = wT_S + (1-w)T_U, \text{for } w \in [0,1]. \tag{1}$$

The average service rate of caching helper S is:

$$\begin{aligned}\mu = {} & q_S(1-q_U)P_{S\to D} + q_S q_U q_D p_{hD} P_{S\to D/D} \\ & + q_S q_U (1-q_D p_{hD})\alpha P_{S\to D/DC} \\ & + q_S q_U (1-q_D p_{hD})(1-\alpha)P_{S\to D}.\end{aligned} \tag{2}$$

As a corollary of the Loynes theorem [40], we obtain that if the arrival and the service process of a queue are strictly jointly stationary and the queue's average arrival rate is less than the queue's average service rate, then the queue is stable. Thus, in our model, the queue at helper S is stable if and only if $\lambda < \mu$. Finite queueing delay is a ramification of a stable queue, and, hence, by adding the aforementioned constraint we can enforce finite queueing delay on our wireless system. Moreover, the stability at S also implies that packets arriving at the queue will eventually be transmitted [40].

The throughput from S to D, denoted as T_S, depends on the stability of the queue Q at S and is $T_S = \lambda$ if the queue is stable or $T_S = \mu$ otherwise. Thus:

$$T_S = \mathbb{1}(\lambda < \mu)\lambda + \mathbb{1}(\lambda \geq \mu)\mu, \tag{3}$$

with $\mathbb{1}(.)$ denoting the indicator function.

The throughput realized by U, denoted by T_U, depends on whether the queue at S is empty or not. The former happens with probability $\mathsf{P}(Q = 0)$ and the latter with probability $\mathsf{P}(Q \neq 0)$. Therefore:

- The queue at S is empty and U requests a file from external resources. In this case, U will be served: (i) by D with probability q_D, or (ii) by S with probability q_C in case of D's failure, or (iii) by the data center with probability α in case both helpers fail.
- If the queue at S is non-empty and U requests a file from external resources, then there are two cases: either (i) helper S attempts transmission to the destination helper D (which happens with probability q_S) or (ii) helper S is available to serve U. In the first case, U will be served by D with probability q_D or by the data center in case D fails to serve U. In the second case, U will be served by D with probability q_D, or by S with probability q_C in case D fails, or by the data center in case both helpers fail to serve U.

Considering all the details above, the throughput realized by user U is:

$$\begin{aligned}
T_U = {} & P(Q=0)q_U \Big[q_D p_{hD} P_{D\to U} + (1 - q_D p_{hD}) q_C p_{hS} P_{S\to U} \\
& + (1 - q_D p_{hD})(1 - q_C p_{hS})\alpha P_{DC\to U}\Big] \\
& + P(Q \neq 0) q_U q_S \\
& \times \Big[q_D p_{hD} P_{D\to U/S} + (1 - q_D p_{hD})\alpha P_{DC\to U/S}\Big] \\
& + P(Q \neq 0) q_U (1 - q_S) \\
& \times \Big[q_D p_{hD} P_{D\to U} + (1 - q_D p_{hD}) q_C p_{hS} P_{S\to U} \\
& + (1 - q_D p_{hD})(1 - q_C p_{hS})\alpha P_{DC\to U}\Big],
\end{aligned} \tag{4}$$

where we have to differentiate cases of stable/unstable queue due to different $P(Q = 0)$ and $P(Q \neq 0)$ for each case. When the queue at S is stable, the probability that Q is not empty is given by: $P(Q \neq 0) = \lambda/\mu$.

In case the average arrival rate is greater than the average service rate, i.e., $\lambda > \mu$, then the queue at S is unstable and can be considered saturated. Consequently, we can

apply a packet dropping policy to stabilize the system and the results for the stable queue can be still valid.

If the queue at S is unstable, the throughput realized by U is:

$$\begin{aligned} T'_U = & \, q_U q_S q_D p_{hD} P_{D\to U/S} \\ & + q_U q_S (1 - q_D p_{hD}) \alpha P_{DC\to U/S} \\ & + q_U (1 - q_S) q_D p_{hD} P_{D\to U} \\ & + q_U (1 - q_S)(1 - q_D p_{hD}) q_C p_{hS} P_{S\to U} \\ & + q_U (1 - q_S)(1 - q_D p_{hD})(1 - q_C p_{hS}) \alpha P_{DC\to U} \end{aligned} \tag{5}$$

We formulate the following mathematical optimization problem to optimize the probabilities q_S, q_C, and q_D that maximize the weighed sum throughput when the queue at helper S is stable:

$$\max . \, w\lambda + (1 - w) T_U \tag{6a}$$

$$\text{s.t. } 0 \leq \lambda < \mu \tag{6b}$$

$$0 \leq q_S, q_C, q_D \leq 1 \tag{6c}$$

The first constraint ensures the stability of the queue at helper S and the second one defines the domain for the decision variables. To solve the aforementioned problem for the case in which the queue at S is unstable, we have to drop the first constraint and replace the expressions for λ and T_U with the ones for μ and T'_U, respectively. In Sect. 5, we provide results for maximizing the weighted sum throughput for some practical scenarios.

4 Delay analysis

Delay experienced by users is another critical performance metric concerning wireless caching systems. In this section, we study the delay that user U experiences when requesting cacheable content from external sources until that content is received. Let $P(S \Rightarrow D) = q_S P(Q \neq 0)$, $P(S \not\Rightarrow D) = 1 - P(S \Rightarrow D)$ and $\bar{q}_i = 1 - q_i$.

The average delay that user U experiences to receive a file from external resources is:

$$
\begin{aligned}
D_U = p_{hD}\big\{ & P(S \Rightarrow D)[(1-q_D)D_{DC,1,D} \\
& + q_D P_{D\to U/S} + q_D \bar{P}_{D\to U/S}(1+D_D)] \\
& + P(S \not\Rightarrow D)[q_D P_{D\to U} + \bar{q}_D p_{hS} D_{S_2}] \\
& + P(S \not\Rightarrow D)\bar{q}_D \bar{p}_{hS} D_{DC,0,D}\big\} \\
& + \bar{p}_{hD} p_{hS}\big\{ P(S \Rightarrow D)\alpha P_{DC\to U/S} \\
& + P(S \Rightarrow D)(1-\alpha P_{DC\to U/S})(1+D_{S_1}) \\
& + P(S \not\Rightarrow D)[q_C P_{S\to U} + q_C \bar{P}_{S\to U}(1+D_{S_1})] \\
& + P(S \not\Rightarrow D)\bar{q}_C D_{DC,0,S}\big\} \\
& + \bar{p}_{hD} \bar{p}_{hS}\big\{ P(S \Rightarrow D)\alpha P_{DC\to U/S} \\
& + P(S \Rightarrow D)(1-\alpha P_{DC\to U/S})(1+D_{DC}) \\
& + P(S \not\Rightarrow D)\alpha P_{DC\to U} \\
& + P(S \not\Rightarrow D)(1-\alpha P_{DC\to U})(1+D_{DC})\big\},
\end{aligned}
\tag{7}
$$

where D_{S_1} is the delay to receive the file from S given D misses it:

$$
D_{S_1} = q_C P_{S\to U} + q_C \bar{P}_{S\to U}(1+D_{S_1}) + \bar{q}_C D_{DC,0,S}, \tag{8}
$$

and D_{S_2} is the delay to receive the file from S given D caches it but does not attempt, i.e., $q_D = 0$, transmissions to U:

$$
D_{S_2} = q_C P_{S\to U} + (1 - q_C P_{S\to U})(1+D_D). \tag{9}
$$

We also need to compute delay caused by the data center DC:

$$
\begin{aligned}
D_{DC} = & P(S \Rightarrow D)\alpha P_{DC\to U/S} \\
& + P(S \Rightarrow D)(1-\alpha P_{DC\to U/S})(1+D_{DC}) \\
& + P(S \not\Rightarrow D)\alpha P_{DC\to U} \\
& + P(S \not\Rightarrow D)(1-\alpha P_{DC\to U})(1+D_{DC}).
\end{aligned}
\tag{10}
$$

Additionally, we need to calculate the following:

$$
D_{DC,0,S} = \alpha P_{DC\to U} + (1-\alpha P_{DC\to U})(1+D_{S_1}), \tag{11}
$$

$$
D_{DC,1,S} = \alpha P_{DC\to U/S} + (1-\alpha P_{DC\to U/S})(1+D_{S_1}), \tag{12}
$$

$$
D_{DC,0,D} = \alpha P_{DC\to U} + (1-\alpha P_{DC\to U})(1+D_D), \tag{13}
$$

$$
D_{DC,1,D} = \alpha P_{DC\to U/S} + (1-\alpha P_{DC\to U/S})(1+D_D), \tag{14}
$$

and:

$$
\begin{aligned}
D_D = & P(S \Rightarrow D) q_D P_{D\to U/S} \\
& + P(S \Rightarrow D) q_D \bar{P}_{D\to U/S}(1+D_D) \\
& + P(S \Rightarrow D)\bar{q}_D D_{DC,1,D} \\
& + P(S \not\Rightarrow D) q_D [P_{D\to U} + \bar{P}_{D\to U}(1+D_D)] \\
& + P(S \not\Rightarrow D)\bar{q}_D [p_{hS} D_{S_2} + \bar{p}_{hS} D_{DC,0,D}].
\end{aligned}
\tag{15}
$$

Table 2 Wireless links parameters

Parameter	Value	Parameter	Value
$P_{tx}(S)$	1 mW	$P_{S\to U}$	0.903
$P_{tx}(D)$	0.5 mW	$P_{D\to U}$	0.607
$P_{tx}(DC)$	10 mW	$P_{DC\to U}$	0.849
n	10^{-11} W	$P_{DC\to U/S}$	0.115
$r_{S\to D}$	50 m	$P_{S\to D}$	0.779
$r_{D\to U}$	50 m	$P_{S\to D/D}$	0.779
$r_{S\to U}$	40 m	$P_{S\to D/DC}$	0.223
$r_{DC\to U}$	80 m	$P_{D\to U/S}$	0.029
$r_{DC\to D}$	100 m	γ_1	0 dB
p	4	γ_2	0 dB

Table 3 Caches parameters and hit probabilities for different values of δ

Parameter	Value	Parameter	Value	
M_U	200	δ	0.5	1.2
M_D	1000	q_U	0.865	0.196
M_S	2000	p_{hD}	0.206	0.109
F	10000	p_{hS}	0.221	0.045

As one can observe, (7)–(15) are recursively defined. After some basic manipulations, (10) becomes:

$$D_{DC} = \frac{1/\alpha}{P(S \Rightarrow D)(P_{DC\to U/S} - P_{DC\to U}) + P_{DC\to U}}. \tag{16}$$

Assuming that $q_C P_{S\to U} - q_C \neq 1$, (8) becomes:

$$D_{S_1} = \left(q_C P_{S\to U} + \alpha(1-q_C)P_{DC\to U}\right)^{-1}. \tag{17}$$

Assuming that $q_C P_{S\to U} + (1-q_C)\alpha P_{DC\to U} \neq 0$ and using (17), (11) and (12) become:

$$D_{DC,0,S} = 1 + \frac{1-\alpha P_{DC\to U}}{q_C P_{S\to U} + (1-q_C)\alpha P_{DC\to U}}, \tag{18}$$

$$D_{DC,1,S} = 1 + \frac{1-\alpha P_{DC\to U/S}}{q_C P_{S\to U} + (1-q_C)\alpha P_{DC\to U}}. \tag{19}$$

Using (9), (13), (14) and applying the regenerative method [41], we get:

$$\begin{aligned} D_D = {} & \bar{q}_D P(S \not\Rightarrow D) q_C p_{hS} P_{S\to U} \\ & + \bar{q}_D \alpha P(S \Rightarrow D) P_{DC\to U/S} \\ & + \bar{q}_D \alpha P(S \not\Rightarrow D) \bar{p}_{hS} P_{DC\to U} \\ & + q_D[P_{D\to U} + P(S \Rightarrow D)(P_{D\to U/S} - P_{D\to U})]. \end{aligned} \tag{20}$$

Substituting (20) to (9), (13), and (14) yields expressions for D_{S_2}, $D_{DC,0,D}$, and $D_{DC,1,D}$, respectively, that are functions of link success probabilities (see Tables 1 and 2) and cache parameters (see Table 3) only.

5 Results and discussion

In this section, we present numerical evaluations of the analysis in the previous sections. The parameters we used for the wireless links between wireless nodes can be found in Table 2. The helpers apply the CMPC policy as described in Sect. 2.2. We consider a finite content library of files, $\mathcal{F} = \{f_1, ..., f_N\}$, to serve users requests. For the sake of simplicity, we assume that all files have equal size and that access to cached files happens instantaneously. The i-th most popular file is denoted as f_i, and the request probability of the i-th most popular file is given by: $p_i = \Omega/i^{\delta}$, where $\Omega = \left(\sum_{j=1}^{N} j^{-\delta}\right)^{-1}$ is the normalization factor and δ is the shape parameter of the Zipf law which determines the correlation of user requests. Consequently, the probability that user U requests a file that is not located in its cache is:

$$q_U = 1 - \sum_{i=1}^{M_U} p_i. \tag{21}$$

The cache hit probability at the caching helper D is given by:

$$p_{hD} = \sum_{i=M_U+1}^{M_U+M_D} p_i, \tag{22}$$

and the cache hit probability at the caching helper S is given by:

$$p_{hS} = \sum_{i=M_U+M_D+1}^{M_U+M_D+M_S} p_i. \tag{23}$$

In the following results, we study the maximum weighted sum throughput which is defined as $T_w = wT_S + (1-w)T_U$ or $T'_w = wT_S + (1-w)T'_U$ when the queue at S is stable or unstable, respectively. The expressions for T_S, T_U, and T'_U are given by (3)–(5) in Sect. 3. To maximize the weighted sum throughput, we solved the optimization problem (6a)–(6c) using the Gurobi optimization solver and report the results. To validate our theoretical results, we built a MATLAB-based behavioral simulator, which shows that the theoretical and the simulation results coincide after 50.000 time slots.

5.1 Maximum weighted sum throughput vs. average arrival rate λ

We consider a scenario where the wireless links parameters follow the values in Table 2. The cache sizes and cache hit probabilities are set as per Table 3 for two different values for the variable δ of the standard Zipf law for the popularity distribution that the cached files follow. In Fig. 5, the maximum weighted sum throughput versus the average arrival rate λ at helper S is presented for three different values of w when the queue at S is stable. We chose: (i) $w = 1/4$ as a representative case in which T_U is more important than T_S, (ii) $w = 2/4$ to equalize the importance of T_U and T_S, and (iii) $w = 3/4$ to put more emphasis on the importance of T_S versus T_U.

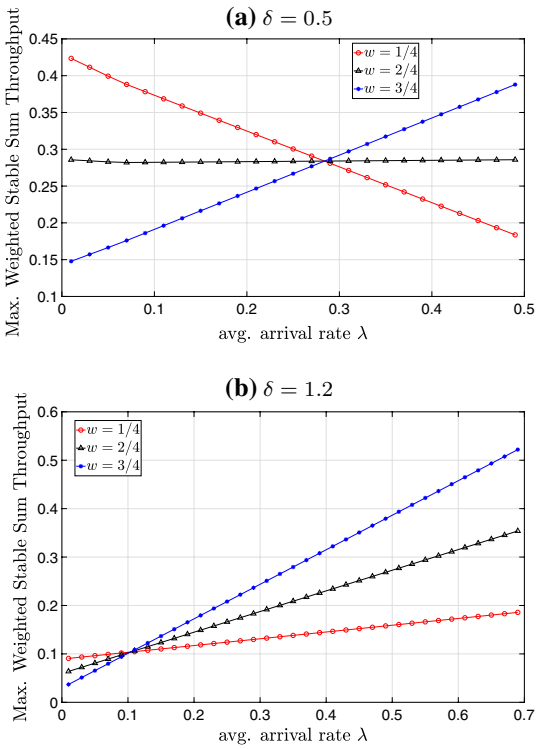

Fig. 5 The maximum weighted sum throughput vs. λ for $\alpha = 0.7$ and different values of w when the queue at S is stable for: **a** $\delta = 0.5$ and **b** $\delta = 1.2$

Table 4 The values of q_S^*, q_C^*, q_D^* for which the weighted sum throughput is maximized and the queue at S is stable for $\alpha = 0.7, M_U = 200, M_D = 1000$, and $M_S = 2000$

	$\delta = 0.5$				$\delta = 1.2$			
w	T_w	q_S^*	q_C^*	q_D^*	T_w	q_S^*	q_C^*	q_D^*
1/4	0.423	0.029	1	0	0.187	0.99	1	1
2/4	0.286	0.978	1	1	0.358	0.99	1	1
3/4	0.392	0.996	1	1	0.536	1	1	1

Table 5 The values of q_S^*, q_C^*, q_D^* for which the weighted sum throughput is maximized and the queue at S is unstable for $\alpha = 0.7, M_U = 200, M_D = 1000$, and $M_S = 2000$

	$\delta = 0.5$				$\delta = 1.2$			
w	T'_w	q_S^*	q_C^*	q_D^*	T'_w	q_S^*	q_C^*	q_D^*
1/4	0.430	0	1	0	0.189	1	0	1
2/4	0.286	0	1	0	0.363	1	0	1
3/4	0.399	1	0.024	1	0.537	1	0	1

In case $w = 1/4$, the maximum weighted sum throughput is a decreasing function of λ when $\delta = 0.5$ (see Fig. 5a), but increasing when $\delta = 1.2$ (see Fig. 5b). When $w = 2/4$, the maximum weighted sum throughput is almost constant for any value of λ when

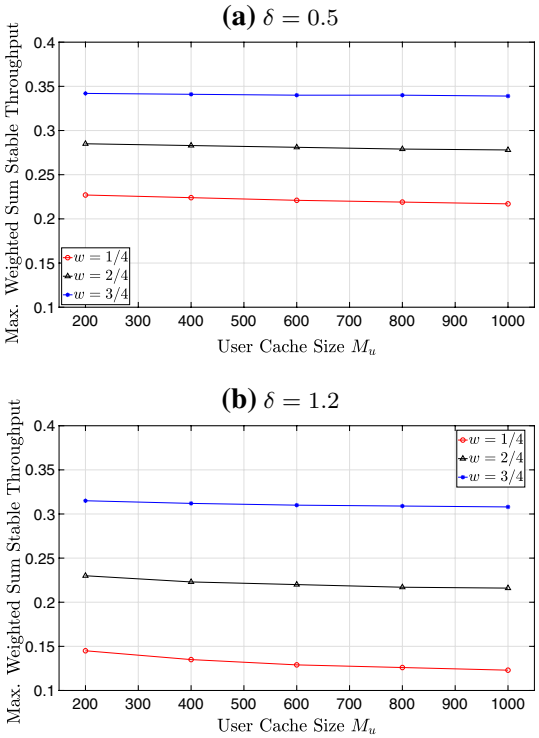

Fig. 6 The maximum weighted sum throughput vs. M_U when the queue at S is stable ($\lambda = 0.4$) and $\alpha = 0.7$ using different values of w for: (**a**) $\delta = 0.5$ and (**b**) $\delta = 1.2$

$\delta = 0.5$ and increases with λ for $\delta = 1.2$. Regarding $w = 3/4$, the maximum weighted sum throughput is an increasing function of λ for any δ value since T_S clearly dominates T_U in this case.

Furthermore, it is observed that the maximum weighted sum throughput is achieved when $q_C^* = 1$ for any value of w and λ when the queue at S is stable (see Table 4), but this is not the case when the queue is unstable, i.e., the average arrival rate λ is greater than the average service rate μ (see Table 5 for different values of δ). When queue at S is unstable, it is optimal for helper S to avoid transmissions, i.e., $q_C^* = 0$, to U when $\delta = 1.2$ for any values of w and λ. When $\delta = 0.5$, helper S must always attempt transmissions to U, i.e., $q_C^* = 1$, when $w \in \{1/4, 2/4\}$ to maximize the weighted sum throughput.

5.2 Maximum weighted sum throughput vs. cache size M_U

In this section, we study how the cache size M_U affects the maximum weighted sum throughput. Recall that q_U decreases as M_U increases. We consider two different values for δ, same as previously, to examine how δ affects maximum weighted sum throughput given different values for M_U.

In Fig. 6, the maximum weighted sum throughput versus M_U is presented for $\alpha = 0.7, M_D = 1000, M_S = 2000$ and $\lambda = 0.4$ for which the queue at S is stable. We observe that as the cache size at U increases, the maximum weighted sum throughput remains almost constant when $\delta = 0.5$ and slightly decreases when $\delta = 1.2$. This is expected since increasing cache size at U results in fewer requests for files from

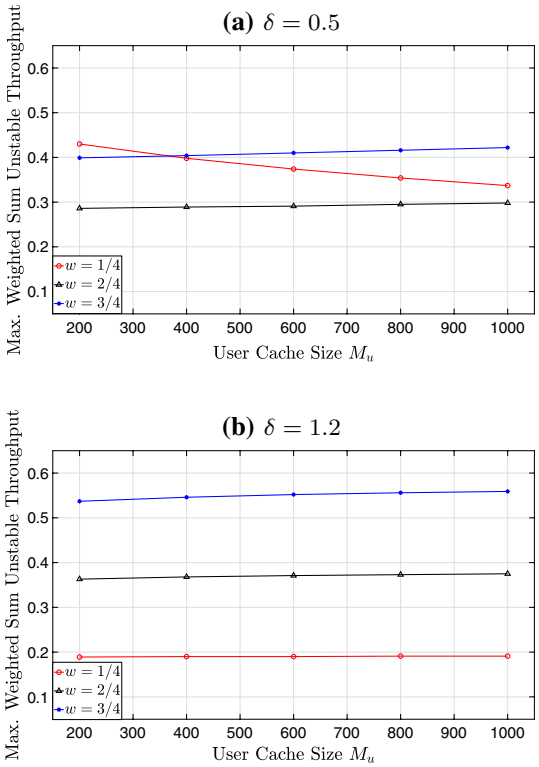

Fig. 7 The maximum weighted sum throughput vs. M_U when the queue at S is unstable and $\alpha = 0.7$ using different values of w for: (**a**) $\delta = 0.5$ and (**b**) $\delta = 1.2$

external results. Moreover, the maximum weighted sum throughput is higher when the value of δ is lower since, for a given cache size, e.g., $M_U = 200$, the probability of requesting content from external resources decreases as δ is increased.

In Fig. 7, the maximum weighted sum throughput versus M_U is presented for the same parameters as in the Fig. 6 but unstable queue at S. The maximum weighted sum throughput is an increasing function of M_U for every value of δ when $w \in \{2/4, 3/4\}$. This is expected since, for these values of w, the throughput achieved by D, i.e., T_S, dominates T_U, and T_S is increasing due to the decrease of requests to external content by U (recall that as M_U increases, q_U decreases). When $w = 1/4$, the maximum weighted sum throughput is almost constant ($\delta = 1.2$) or decreases ($\delta = 0.5$) as M_U increases. The latter decrease can be attributed to the fact that T_U ,i.e., the dominant term in the maximum weighted sum throughput, decreases as q_U decreases (and M_U increases).

The values of q_S^*, q_C^*, and q_D^* that achieve the maximum weighted sum throughput are given in Tables 6 and 7 when the queue at S is stable and unstable, respectively.

In case $\delta = 0.5$ and the queue at S is stable, the maximum weighted sum throughput T_w is achieved for $q_C^* = 1$ and $q_D^* = 1$ for every value of M_U and w. This means that, for the aforementioned parameters, user U should always be assisted by both S and D to achieve maximum weighted sum throughput T_w. This is not the case for $\delta = 1.2$ while the queue at S is stable and $M_U \geq 400$. For every value of $w \in \{1/4, 2/4, 3/4\}$, user U should only be assisted by S to achieve the maximum

Table 6 The values of (q_S^*, q_C^*, q_D^*) that maximize the weighted sum throughput T_w for different values of M_U when $\alpha = 0.7$ and the queue at S is stable

$\delta = 0.5$				
M_U		$w = 1/4$	$w = 2/4$	$w = 3/4$
200	max. T_w	0.227	0.285	0.342
	(q_S^*, q_C^*, q_D^*)	(0.999, 1, 1)	(0.999, 1, 1)	(0.999, 1, 1)
400	max. T_w	0.224	0.283	0.341
	(q_S^*, q_C^*, q_D^*)	(0.801, 1, 1)	(0.824, 1, 1)	(0.769, 1, 1)
600	max. T_w	0.221	0.281	0.340
	(q_S^*, q_C^*, q_D^*)	(0.756, 1, 1)	(1, 1, 1)	(1, 1, 1)
800	max. T_w	0.219	0.279	0.340
	(q_S^*, q_C^*, q_D^*)	(1, 1, 1)	(1, 1, 1)	(0.746, 1, 1)
1000	max. T_w	0.217	0.278	0.339
	(q_S^*, q_C^*, q_D^*)	(0.732, 1, 1)	(1, 1, 1)	(1, 1, 1)
$\delta = 1.2$				
200	max. T_w	0.145	0.230	0.315
	(q_S^*, q_C^*, q_D^*)	(0.713, 1, 1)	(0.713, 1, 1)	(0.713, 1, 1)
400	max. T_w	0.135	0.223	0.312
	(q_S^*, q_C^*, q_D^*)	(1, 1, 0)	(0.999, 1, 0)	(0.999, 1, 0)
600	max. T_w	0.129	0.220	0.310
	(q_S^*, q_C^*, q_D^*)	(1, 1, 0)	(1, 1, 0)	(1, 1, 0)
800	max. T_w	0.126	0.217	0.309
	(q_S^*, q_C^*, q_D^*)	(1, 1, 0)	(1, 1, 0)	(1, 1, 0)
1000	max. T_w	0.123	0.216	0.308
	(q_S^*, q_C^*, q_D^*)	(1, 1, 0)	(1, 1, 0)	(0.540, 1, 0)

weighted sum throughput T_w since $q_C^* = 1$ and $q_D^* = 0$. We also observe that, in this case, S should more frequently assist D since q_S^* has almost always a higher value compared to $\delta = 0.5$.

In case the queue at S is unstable, the values of (q_S^*, q_C^*, q_D^*) for which the maximum weighted sum throughput T_w' is achieved can be found in Table 7. We observe that neither helper S should serve helper D ($q_S^* = 0$) nor the latter should assist U ($q_D^* = 0$) to maximize T_w' when (i) $\delta = 0.5$ and $w = 1/4$ for any cache size M_U or (ii) $\delta = 0.5, w = 2/4$ and user U's cache can hold $M_U = 200$ files.

Moreover, when $\delta = 0.5, M_U \geq 400$ and $w \in \{2/4, 3/4\}$, helper S should only serve helper D and the latter should assist user U since $(q_S^*, q_D^*) = (1, 1)$. However, helper S should slightly assist U in some cases when, e.g., $M_U = 400$ or 600. When $\delta = 1.2$, helper S should only serve the destination helper D and the latter should assist user U for any value of M_U and w. Additionally, helper S should not assist user U for any cache size M_U but 400.

Furthermore, it should be noted that, for any value of M_U, the maximum weighted sum throughput is decreasing as δ increases when $w = 1/4$ and increases as δ increases when $w \in \{2/4, 3/4\}$.

5.3 Maximum weighted sum throughput vs. average arrival rate λ when $M_D = 0$

We consider a scenario where the system parameters are the same as in Sect. 5.1 (see Tables 2 and 3), but helper D cannot assist user U since its cache cannot hold any

Table 7 The values of (q_S^*, q_C^*, q_D^*) that maximize the weighted sum throughput T_w for different values of M_U when $\alpha = 0.7$ and the queue at S is unstable

$\delta = 0.5$				
M_U		$w = 1/4$	$w = 2/4$	$w = 3/4$
200	max. T'_w	0.430	0.286	0.399
	(q_S^*, q_C^*, q_D^*)	(0, 1, 0)	(0, 1, 0)	(1, 0.024, 1)
400	max. T'_w	0.398	0.289	0.404
	(q_S^*, q_C^*, q_D^*)	(0, 1, 0)	(1, 0.029, 1)	(1, 0.005, 1)
600	max. T'_w	0.374	0.295	0.410
	(q_S^*, q_C^*, q_D^*)	(0, 1, 0)	(1, 0, 1)	(1, 0.011, 1)
800	max. T'_w	0.354	0.295	0.416
	(q_S^*, q_C^*, q_D^*)	(0, 1, 0)	(1, 0, 1)	(1, 0, 1)
1000	max. T'_w	0.337	0.298	0.422
	(q_S^*, q_C^*, q_D^*)	(0, 1, 0)	(1, 0, 1)	(1, 0, 1)
$\delta = 1.2$				
200	max. T'_w	0.189	0.363	0.537
	(q_S^*, q_C^*, q_D^*)	(1, 0, 1)	(1, 0, 1)	(1, 0, 1)
400	max. T'_w	0.190	0.368	0.546
	(q_S^*, q_C^*, q_D^*)	(1, 0.046, 1)	(1, 0, 1)	(1, 0.484, 1)
600	max. T'_w	0.190	0.371	0.552
	(q_S^*, q_C^*, q_D^*)	(1, 0, 1)	(1, 0, 1)	(1, 0, 1)
800	max. T'_w	0.191	0.373	0.556
	(q_S^*, q_C^*, q_D^*)	(1, 0, 1)	(1, 0, 1)	(1, 0, 1)
1000	max. T'_w	0.191	0.375	0.559
	(q_S^*, q_C^*, q_D^*)	(1, 0, 1)	(1, 0, 1)	(1, 0, 1)

files, i.e., $M_D = 0$. Consequently, $q_D = 0$ and $p_{hD} = 0$ as well. This scenario will allow the study of the maximum weighted sum throughput versus λ when only one of the two helpers, the least powerful, is unable satisfy U's needs for content from external resources.

In Fig. 8, we plot the maximum weighted sum throughput versus λ when the queue at S is stable and $M_D = 0$. We observe that, when $\delta = 0.5$, the maximum weighted sum throughput (i) is a decreasing function of λ for $w = 1/4$, (ii) slightly decreases for $w = 2/4$, and (iii) increases for $w = 3/4$. Recall that, by definition, in the first case T_U dominates T_S, in the second case both throughput terms contribute equally, and in the third case T_S dominates T_U. When $\delta = 1.2$, the maximum weighted sum throughput is increasing with λ. We observe that higher values of w yield steeper increases in the maximum weighted sum throughput.

When the queue at S is stable, the maximum weighted sum throughput is always achieved when $q_C^* = 1$ for any w, δ, and λ using the system parameters we quoted before. However, in case $\delta = 0.5$, helper S should nearly always assist U since $q_S^* \in \{0.977, 0.999\}$. When $\delta = 1.2$, helper S should always assist U as Table 8 depicts.

On the other hand, when the queue at S is unstable and $\delta = 0.5$, helper S should only assist U when $w \in \{1/4, 2/4\}$ and only assist D when $w = 3/4$ (see Table 9). This

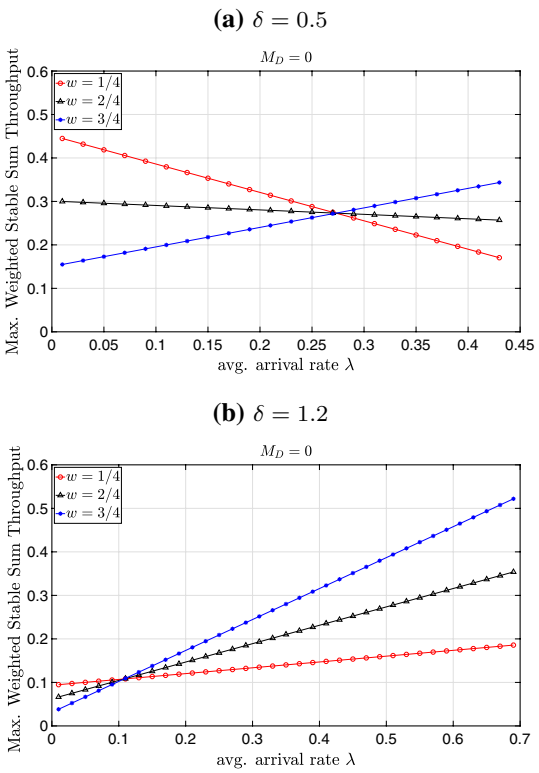

Fig. 8 The maximum weighted sum throughput vs the average arrival rate λ for which the queue at S is stable, $M_U = 200, M_D = 0, M_S = 2000$, and $\alpha = 0.7$ using different values of w for: (**a**) $\delta = 0.5$ and (**b**) $\delta = 1.2$

Table 8 The values of q_S^* and q_C^* for which the weighted sum throughput is maximized when the queue at S is stable, $\alpha = 0.7, M_U = 200, M_D = 0$, and $M_S = 2000$

$M_D = 0$	$\delta = 0.5$			$\delta = 1.2$		
w	max.T_w	q_S^*	q_C^*	max. T_w	q_S^*	q_C^*
1/4	0.445	0.994	1	0.187	1	1
2/4	0.300	0.977	1	0.358	1	1
3/4	0.348	0.999	1	0.529	1	1

Table 9 The values of q_S^* and q_C^* for which the weighted sum throughput is maximized when the queue at S is unstable, $\alpha = 0.7, M_U = 200, M_D = 0$, and $M_S = 2000$

$M_D = 0$	$\delta = 0.5$			$\delta = 1.2$		
w	max.T'_w	q_S^*	q_C^*	max. T'_w	q_S^*	q_C^*
1/4	0.451	0	1	0.187	1	0
2/4	0.301	0	1	0.359	1	0
3/4	0.349	1	0	0.531	1	0

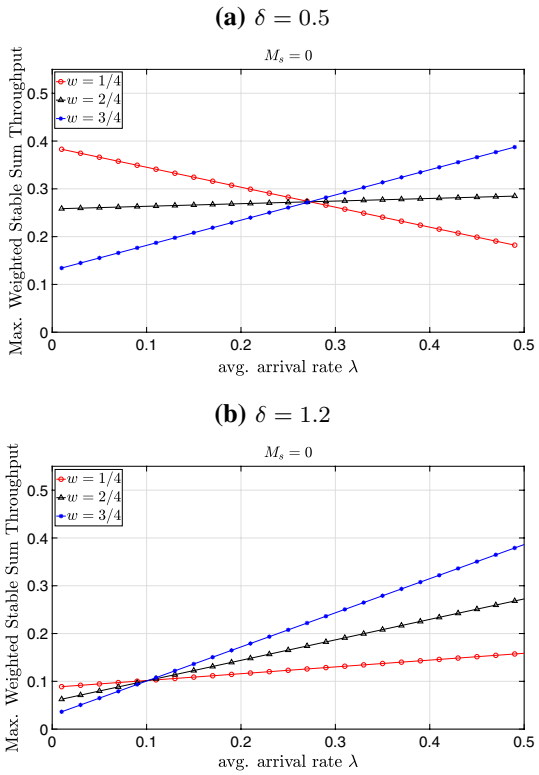

Fig. 9 The maximum weighted sum throughput vs. the average arrival rate λ for which the queue at S is stable, $M_U = 200, M_D = 1000, M_S = 0$ and $\alpha = 0.7$ using different values of w for: (**a**) $\delta = 0.5$ and (**b**) $\delta = 1.2$

Table 10 The values of q_S^* and q_D^* for which the weighted sum throughput is maximized when the queue at S is stable, $\alpha = 0.7, M_U = 200, M_D = 1000$, and $M_S = 0$

$M_S = 0$	$\delta = 0.5$			$\delta = 1.2$		
w	max.T_w	q_S^*	q_D^*	max.T_w	q_S^*	q_D^*
1/4	0.383	0.993	1	0.189	1	1
2/4	0.296	1	1	0.362	1	1
3/4	0.498	1	1	0.536	1	1

is expected since in the latter case, T_S dominates T_U and, hence, it is preferable that S always serves D to maximize the contribution of T_S. In this case, if user U requests content from external resources, it will be only served by the data center. Moreover, when $\delta = 1.2$, it is optimal that helper S serves only D for any value of w.

5.4 Maximum weighted sum throughput vs. average arrival rate λ when $M_S = 0$

Here, we study the maximum weighted sum throughput versus the average arrival rate λ when node S is not equipped with cache, i.e., $M_S = 0$, and, hence, $q_C = 0$ and $p_{hS} = 0$. The parameters of helper D's cache and the wireless links can be found in Tables 3 and 2, respectively.

Table 11 The values of q_S^* and q_D^* for which the weighted sum throughput is maximized when the queue at S is unstable, $\alpha = 0.7, M_U = 200, M_D = 1000$, and $M_S = 0$

$M_S = 0$	$\delta = 0.5$			$\delta = 1.2$		
w	max.T'_w	q_S^*	q_D^*	max.T'_w	q_S^*	q_D^*
1/4	0.387	0	1	0.189	1	1
2/4	0.286	1	1	0.363	1	1
3/4	0.399	1	1	0.537	1	1

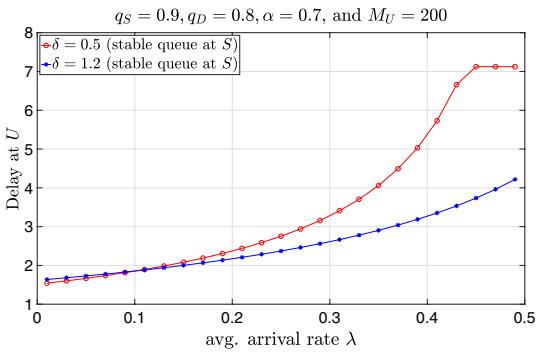

Fig. 10 The average delay at U vs. average arrival rate λ at S for $\delta \in \{0.5, 1.2\}$

In Fig. 9, we plot the maximum weighted sum throughput versus λ for which the queue at S is stable when $M_S = 0$ for different values of w. Regarding $\delta = 0.5$, when $w = 1/4$, the maximum weighted sum throughput is decreasing with λ. When $w \in \{2/4, 3/4\}$, and $\delta \in \{0.5, 1.2\}$ or $w = 1/4$ and $\delta = 1.2$, the maximum weighted sum throughput is an increasing function of λ.

In Table 10, we present the values of q_S^* and q_D^* that achieve maximum weighted sum throughput when the queue at S is stable for different values of w. Recall that, in this specific scenario, $q_C^* = 0$ since helper S has no cache and, thus, it cannot assist U. Therefore, S is only useful to helper D. We observe that the maximum weighted sum throughput is lowered compared to the case when $M_D = 0$ for $\delta = 0.5$ and slightly higher for $\delta = 1.2$ (compare with Table 8). Additionally, helper S should almost always serve D and the later should always assist user U to achieve the maximum weighted sum throughput.

In Table 11, we present the values of q_S^* and q_D^* that achieve maximum weighted sum throughput when the queue at S is unstable for different values of w. In order to maximize the weighted sum throughput, helper S should always serve D for any values of δ and w apart from the case in which $w = 1/4$ and $\delta = 0.5$ for which S should remain silent since $q_S^* = 0$. Furthermore, helper D should always assist U requests for every value of w and δ we used. The maximum weighted sum throughput is higher compared to the case in which $M_D = 0$ (compare with Table 9) for every value of w and δ apart from the cases in which $\delta = 0.5$ and $w \in \{1/4, 2/4\}$.

5.5 Average delay at user *U*

Here, we present the numerical results of the average delay experienced by user U to receive content from external sources. The delay analysis can be found in Sect. 4.

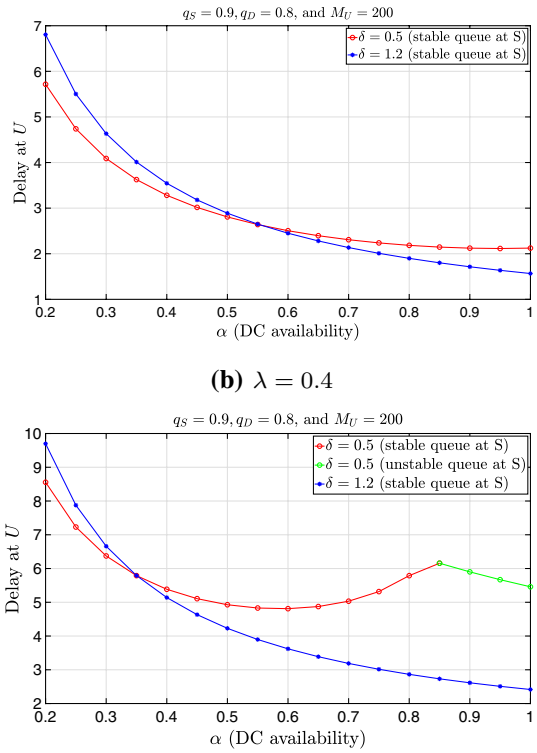

Fig. 11 The average delay at U vs. α for $\delta \in \{0.5, 1.2\}$

In the following plots, we study how the average arrival rate λ, the data center's random availability α, the probability that S attempts transmissions to D, q_S, the probability that D attempts transmissions to U, q_D, and the cache size at U, M_U affect the average delay at U. The wireless links characteristics can be found in Table 2. The cache sizes were set to hold $M_S = 2000$ and $M_D = 1000$ files at S and D, respectively, and we used two different values for δ to examine its effect on the realized average delay. Hence, the values of q_U, p_{hD}, and p_{hS} were given by (21)–(23) depending on δ. Also, we set $q_C = 0.5$.

In Fig. 10, the average delay versus the arrival rate at helper S is depicted for $q_S = 0.9, q_D = 0.8, \alpha = 0.7$ and $M_U = 200$. We observe that the delay increases with the arrival rate and the increase rate is steeper when $\delta = 0.5$ compared to $\delta = 1.2$. As we explained in Sect. 2.2, higher values of δ yield more requests for a few most popular files. Therefore, for a given M_U, the higher the δ, the lower the q_U, i.e., user U requests files from external sources with lower probability, as well as lower value for cache hits p_{hD} and p_{hS} (for given M_D and M_S). Fewer requests for files from external resources require fewer transmissions to U and, hence, less interference is realized. Consequently, less average delay is experienced at U.

In Fig. 11, we present the average delay at U versus data center's availability for two cases of arrival rate $\lambda = 0.2$ and $\lambda = 0.4$. We observe that the delay is lower when $\lambda = 0.2$ since a higher average arrival rate is more likely to create a congested queue at S and, consequently, a higher delay. In case $\lambda = 0.2$, the delay is decreased with the increase

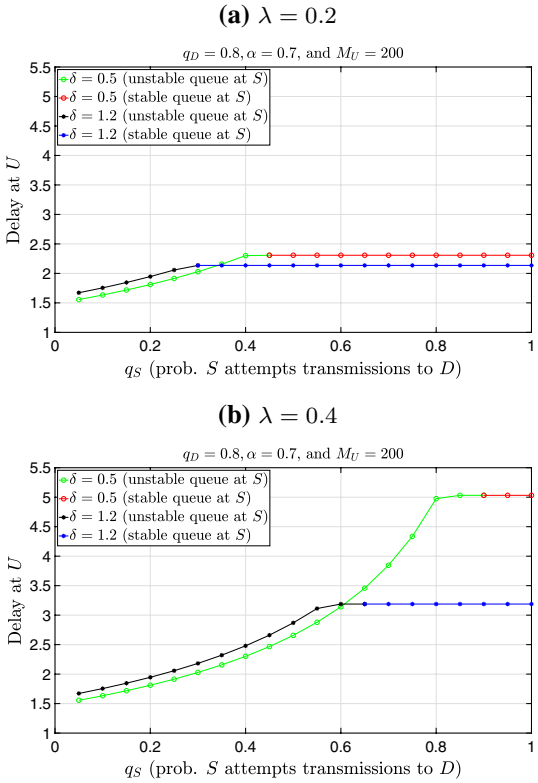

Fig. 12 The average delay at U vs. q_S for $\delta \in \{0.5, 1.2\}$

of α and the queue at helper S is stable for any $\alpha \in [0.2, 1]$. Additionally, the decrease is steeper with α when $\delta = 1.2$. When $\lambda = 0.4$ and $\delta = 0.5$, the queue at S remains stable for $\alpha \in [0.2, 0.8]$ and the delay has the non-monotonic behavior of Fig. 11b. For $\alpha \in [0.8, 1]$, the average delay starts decreasing with α and the queue at S is unstable. When $\delta = 1.2$, the queue at S is stable for every value of α and the delay is decreased with the increased availability of the data center.

In Fig. 12, we plot the average delay at U versus q_S for $\lambda = 0.2$ and $\lambda = 0.4$. We observe that as long as the queue at S is unstable, the delay increases with the q_S increase. This is expected since as q_S increases, helper S attempts more transmissions to helper D and, consequently, it is not only less likely to assist U but also U's probability to find an available helper is decreased (since the $S - D$ pair communicates more). Regarding the case in which the queue at S is stable, increasing q_S does not contribute to delay's improvement. Moreover, a lower value of q_S is required to achieve queue stability at S when $\lambda = 0.2$ compared to $\lambda = 0.4$. This is expected since a higher average arrival rate requires a higher average service rate to maintain queue stability.

In Fig. 13, we demonstrate the average delay at U versus q_D for $\lambda = 0.2$ and $\lambda = 0.4$. In the former case, the delay is slightly decreased with the increase of q_D This can be attributed to helper D's increased assistance that yields more transmissions to U and, hence, potentially decreased delay. When $\lambda = 0.4$, the average delay decreases considerably with q_D when $\delta = 0.5$ due to the increased assistance of helper D, but decreases slightly in case $\delta = 1.2$. This is expected, as we previously explained, since higher values

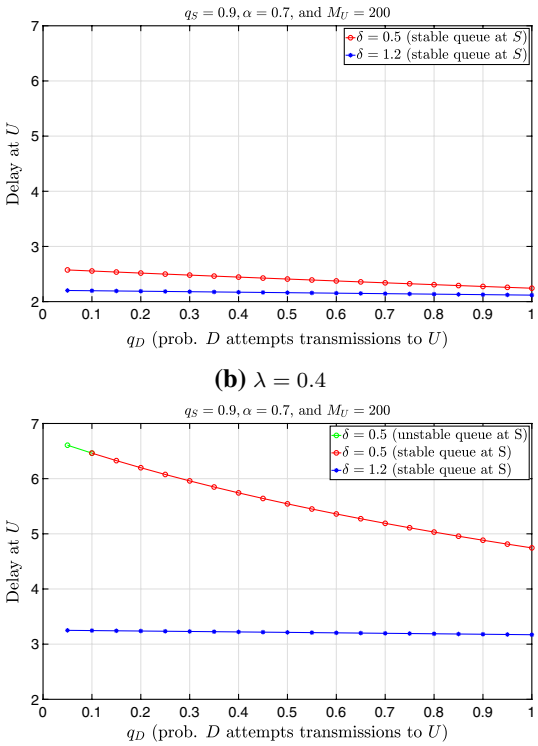

Fig. 13 The average delay at U vs. q_D for $\delta \in \{0.5, 1.2\}$

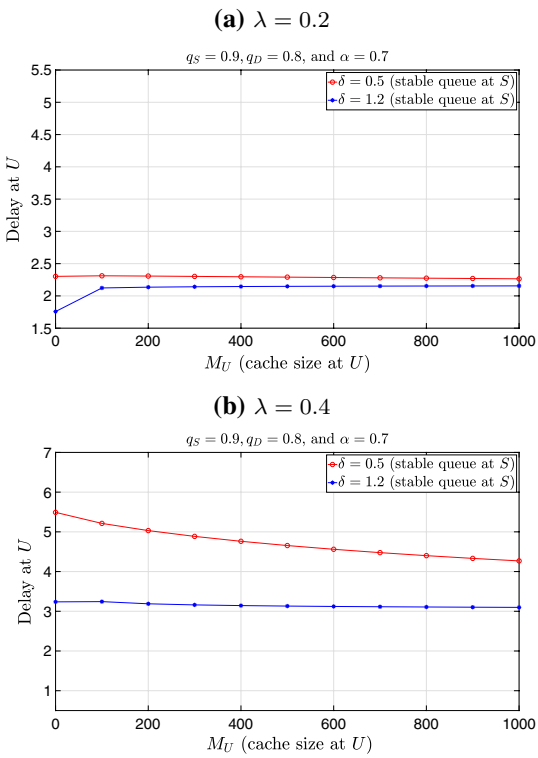

Fig. 14 The average delay at U vs. M_U for $\delta \in \{0.5, 1.2\}$

of δ create more requests for a few most popular files, and, thus, U's request for external content is decreased. As a result, the average delay at U is decreased compared to lower δ values.

In Fig. 14, we show the average delay at U versus the cache size at U for $\lambda = 0.2$ and $\lambda = 0.4$. The cache size M_U affects the request probability for external content, q_U, and as it increases, the q_U decreases. In any case, the queue at S is stable. When $\lambda = 0.2$, the effect of M_U on the average delay at U is minor. However, when average arrival rate λ is increased, then increasing cache size at U decreases the average delay especially when δ is lowered.

5.6 Summarized results

Here, we present a summary of the results in the previous parts of the manuscript. By denoting with T_w^* the maximum weighted sum throughput, the main observations are the following:

- If Q, i.e., the queue at S, is stable , T_w^* is achieved when the caching helper S is always available to serve the cached user U, i.e., $q_C = 1$, provided that every other parameter is fixed. Instead, if Q is unstable, it is not always optimal to set $q_C = 0$.
- T_w^* is almost constant or slightly decreasing with M_U, i.e., the cache size at U, when Q is stable no matter what value for δ is assumed, given that every other parameter is fixed. This is not the case when Q is unstable, though. For instance, T_w^* is decreased with M_U when $w = 1/4$ and $\delta = 0.5$.
- When one only caching helper (either S or D) is available to assist U, T_w^* is increasing with average arrival rate λ when $\delta = 1.2$, provided that every other parameter is fixed. On the other hand, when $\delta = 0.5$, the trend of T_w^* vs. λ varies with w.

Regarding D_U, i.e., the average delay realized by the cached user U given by (7), the main observations are the following:

- D_U increases with the average arrival rate λ and the increasing rate is steeper, when higher δ is assumed, given that every other parameter is fixed.
- D_U is decreased with α, i.e., the probability of DC being available to U, when the average arrival rate λ is relatively low, e.g., 0.2, provided that every other parameter is fixed. D_U will probably exhibit a different behavior with α for λ values that might cause an unstable queue at S.
- D_U increases with q_S, i.e., the probability of S being available for D, when the queue is unstable, given that every other parameter is fixed. On the other hand, when Q is stable, increasing q_S does not contribute to D_U improvement.
- D_U is slightly decreased with q_D, i.e., the probability of D being available for U, when the average arrival rate $\lambda = 0.2$, provided that every other parameter is fixed. For double λ, D_U decreases considerably with q_D when $\delta = 0.5$ is assumed, and slightly decreases with q_D when $\delta = 1.2$.

- D_U is not considerably affected when we vary the cache size at M_U from 100 to 1000 and $\lambda = 0.2$, given that every other parameter is fixed. If the average arrival rate is doubled, then increasing M_U is beneficial in terms of D_U especially when $\delta = 1.2$.

6 Conclusion

In this paper, we studied the effect of multiple randomly available caching helpers on a wireless system that serves cacheable and non-cacheable traffic. We derived the throughput for a system consisting of a user requesting cacheable content from a pair of caching helpers within its proximity or a data center. The helpers are assumed to exchange non-cacheable content as well as assisting the user's needs for cacheable content in a random manner. We optimized the probabilities by which the helpers assist the user's requests to maximize the system throughput. Moreover, we studied the average delay experienced by the user from the time it requested cacheable content till content reception.

Our theoretical and numerical results provide insights concerning the system throughput and the delay behavior of wireless systems serving both cacheable and non-cacheable content with assistance of multiple randomly available caching helpers.

Abbreviations

BS: Base station (see Section 2.1); C-RANs: Cloud RANs (see RANs below); CaaS: Caching as a service; CMPC: Collaborative most popular content (see Section 2.2); DC: Data center (see Section 2.1); FBS: Femto base station; IoT: Internet of Things; QoS: Quality of service; RANs: Random access networks; SINR: Signal to interference plus noise ratio (see Section 2.4).

Authors' contributions

This work was based on part of the Ph.D. research work of IA at Linköping University. The main idea was conceived by IA and NP. This research work was supervised by NP and VA. IA developed both the theoretical and simulation results as well as the corresponding scientific interpretations. VA helped in the final review of the paper. The authors read and approved the final manuscript.

Funding

Open access funding provided by Linköping University. This work was supported in part by CENIIT and ELLIIT.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Department of Computer and Information Science (IDA), Linköping University, 58183 Norrköping, Sweden. [2] Department of Science and Technology, Linköping University, 60174 Norrköping, Sweden.

Received: 26 March 2020 Accepted: 2 March 2021
Published online: 29 March 2021

References

1. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022 White Paper. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html
2. G. Paschos, E. Bastug, I. Land, G. Caire, M. Debbah, Wireless caching: technical misconceptions and business barriers. IEEE Commun. Mag. **54**(8), 16–22 (2016)
3. K. Shanmugam, N. Golrezaei, A.G. Dimakis, A.F. Molisch, G. Caire, FemtoCaching: wireless content delivery through distributed caching helpers. IEEE Trans. Inf. Theory **59**(12), 8402–8413 (2013)
4. G.S. Paschos, G. Iosifidis, M. Tao, D. Towsley, G. Caire, The role of caching in future communication systems and networks. IEEE J. Sel. Areas Commun. **36**(6), 1111–1125 (2018)

5. M.A. Maddah-Ali, U. Niesen, Fundamental limits of caching. IEEE Trans. Inf. Theory **60**(5), 2856–2867 (2014)
6. Z. Chen, N. Pappas, M. Kountouris, Probabilistic caching in wireless D2D networks: cache hit optimal versus throughput optimal. IEEE Commun. Lett. **21**(3), 584–587 (2017)
7. E. Baştuğ, M. Kountouris, M. Bennis, M. Debbah, On the delay of geographical caching methods in two-tiered heterogeneous networks, in *IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5 (2016)
8. N. Pappas, Z. Chen, I. Dimitriou, Throughput and delay analysis of wireless caching helper systems with random availability. IEEE Access **6**, 9667–9678 (2018)
9. J. Ma, J. Wang, P. Fan, A cooperation-based caching scheme for heterogeneous networks. IEEE Access **5**, 15013–15020 (2017)
10. Y. Wang, X. Tao, X. Zhang, Y. Gu, Cooperative caching placement in cache-enabled D2D underlaid cellular networks. IEEE Commun. Lett. **21**(5), 1151–1154 (2017)
11. Z. Chen, J. Lee, T.Q.S. Quek, M. Kountouris, Cooperative caching and transmission design in cluster-centric small cell networks. IEEE Trans. Wireless Commun. **16**(5), 3401–3415 (2017)
12. M. Naslcheraghi, M. Afshang, H.S. Dhillon, Modeling and performance analysis of full-duplex communications in cache-enabled D2D networks, in *IEEE International Conference on Communications (ICC)* pp. 1–6 (2018)
13. Z. Chen, M. Kountouris, D2D caching vs. small cell caching: where to cache content in a wireless network? in *IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–6 (2016)
14. D. Liu, C. Yang, Energy efficiency of downlink networks with caching at base stations. IEEE J. Sel. Areas Commun. **34**(4), 907–922 (2016)
15. S. Lin, D. Cheng, G. Zhao, Z. Chen, Energy-efficient wireless caching in device-to-device cooperative networks, in *IEEE 85th Vehicular Technology Conference (VTC Spring)* (2017)
16. B. Chen, C. Yang, A.F. Molisch, Cache-enabled device-to-device communications: offloading gain and energy cost. IEEE Trans. Wireless Commun. **16**(7), 4519–4536 (2017)
17. J. Rao, H. Feng, C. Yang, Z. Chen, B. Xia, Optimal caching placement for D2D assisted wireless caching networks, in *IEEE International Conference on Communications (ICC)* pp. 1–6 (2016)
18. D. Malak, M. Al-Shalash, J.G. Andrews, Spatially correlated content caching for device-to-device communications. IEEE Trans. Wireless Commun. **17**(1), 56–70 (2018)
19. W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, Z. Zhang, Edge caching at base stations with device-to-device offloading. IEEE Access **5**, 6399–6410 (2017)
20. B. Chen, C. Yang, Z. Xiong, Optimal caching and scheduling for cache-enabled D2D communications. IEEE Commun. Lett. **21**(5), 1155–1158 (2017)
21. Y. Chen, H. Zhang, Exploiting transmission and caching diversity in cache-enabled user-centric network: analysis and optimization. IEEE Access **7**, 65934–65943 (2019)
22. K. Thar, N.H. Tran, S. Ullah, T.Z. Oo, C.S. Hong, Online caching and cooperative forwarding in information centric networking. IEEE Access **6**, 59679–59694 (2018)
23. F. Rezaei, B.H. Khalaj, Stability, rate, and delay analysis of single bottleneck caching networks. IEEE Trans. Commun. **64**(1), 300–313 (2016)
24. D. Bethanabhotla, G. Caire, M.J. Neely, Adaptive video streaming for wireless networks with multiple users and helpers. IEEE Trans. Commun. **63**(1), 268–285 (2015)
25. X. Hong, J. Jiao, A. Peng, J. Shi, C. Wang, Cost optimization for on-demand content streaming in IoV networks with two service tiers. IEEE IoT J. **6**(1), 38–49 (2019)
26. L. Hou, L. Lei, K. Zheng, X. Wang, A Q-learning based proactive caching strategy for non-safety related services in vehicular networks. IEEE IoT J. **6**, 4512–4520 (2019)
27. S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, W. Wang, A survey on mobile edge networks: convergence of computing. IEEE Access Caching Commun. **5**, 6757–6779 (2017)
28. X. Liu, J. Zhang, X. Zhang, W. Wang, Mobility-aware coded probabilistic caching scheme for MEC-enabled small cell networks. IEEE Access **5**, 17824–17833 (2017)
29. Y. Wu, S. Yao, Y. Yang, T. Zhou, H. Qian, H. Hu, M. Hamalainen, Challenges of mobile social device caching. IEEE Access **4**, 8938–8947 (2016)
30. S. Lien, S. Hung, D. Deng, C. Lai, H. Tsai, low latency radio access in 3GPP local area data networks for V2X: stochastic optimization and learning. IEEE IoT J. **6**, 4867–4879 (2019)
31. Z. Piao, M. Peng, Y. Liu, M. Daneshmand, Recent advances of edge cache in radio access networks for internet of things: techniques, performances, and challenges. IEEE IoT J. **6**(1), 1010–1028 (2019)
32. S. Sardellitti, G. Scutari, S. Barbarossa, Joint optimization of radio and computational resources for multicell mobile-edge computing. IEEE Trans. Signal Inf. Process. Netw. **1**(2), 89–103 (2015)
33. S. Barbarossa, S. Sardellitti, P. Di Lorenzo, Joint allocation of computation and communication resources in multiuser mobile cloud computing, in: *IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 26–30 (2013)
34. Y. Wei, F.R. Yu, M. Song, Z. Han, Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning. IEEE IoT J. **6**(2), 2061–2073 (2019)
35. J. Yao, N. Ansari, Joint content placement and storage allocation in C-RANs for IoT sensing service. IEEE IoT J. **6**(1), 1060–1067 (2019)
36. X. Huang, N. Ansari, Content caching and distribution in smart grid enabled wireless networks. IEEE IoT J. **4**(2), 513–520 (2017)
37. X. Li, X. Wang, K. Li, V.C.M. Leung, CaaS: caching as a service for 5G networks. IEEE Access **5**, 5982–5993 (2017)

38. I. Avgouleas, N. Pappas, V. Angelakis, performance evaluation of wireless caching helper systems, in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–6 (2019)
39. N. Pappas, M. Kountouris, A. Ephremides, A. Traganitis, Relay-assisted multiple access with full-duplex multi-packet reception. IEEE Trans. Wireless Commun. **14**(7), 3544–3558 (2015)
40. R.M. Loynes, The stability of a queue with non-independent inter-arrival and service times. Math. Proc. Cambridge Philos. Soc. Cambridge Univ. Press **58**(3), 497–520 (1962)
41. J. Walrand, *Communication Networks: A First Course*, 2nd edn. (McGraw-Hill, New York, NY, 1998)

Publisher's Note

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

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com