Caching Video on Demand in Radio Clusters

Furkan Murat, Ertan Onur
Department of Computer Engineering
Middle East Technical University, Ankara Turkey
fmurat@ceng.metu.edu.tr, eronur@metu.edu.tr

***Abstract*—Densification of mobile networks leads to some problems such as overloaded backhaul and increased latency. Base stations may not always have fiber backhauls. For such cases, radio links are employed between base stations towards a base station that has a high capacity backhaul link. Those set of base stations that are connected to the hub over radio links are called radio clusters. We propose caching in radio clusters for decreasing the video streaming load on radio links. We define an optimization problem to determine the cache locations for video-on-demand segments considering user requirements. We tested our model with different radio cluster configurations concentrating on the size of the radio cluster, the number of users, and the number of demanded videos. Significant gains can be attained by employing the proposed cache placement and replication solution in radio clusters.**

I. Introduction

Densification of base stations helps mobile operators provide higher capacities to end users by employing frequency reuse and decreasing user-to-base station distances. However, densification reflects itself as a difficulty in backhauling user traffic. Backhaul is the set of links from base stations towards the core of the network. Since deploying high-capacity fiber cables to backhaul traffic from all base stations is costly, mobile operators tend to employ micro- or mm-wave radio links for backhauling from base stations that do not have any high capacity connection to the core. Network operators consider various topologies for establishing paths from base stations to a concentration site with a high capacity (e.g., fiber) connection [1]. We will refer to the concentration site as the hub in this paper. Network operators consider many factors such as whether or not a line of sight (LOS) exists between sites, wireless propagation characteristics, or environmental conditions when designing the topology of radio clusters. Multi-hopping is also common when there is no line-of-sight between a base station and the hub. A set of base stations connected to the same hub over one or more radio links is called a radio cluster.

Caching contents close to end-users reduces latency and enhances the response time of applications. TCP applications may attain further benefit by acquiring a larger portion of the bandwidth since TCP employs ACK clocking in its congestion control mechanisms. Reducing latency helps TCP increase its congestion window faster. Multimedia streaming applications that employ rate adaption such as DASH [2], which works on HTTP over TCP, benefit considerably from such latency reductions. Therefore, network operators prefer caching video segments close to users instead of letting each user fetch the content from distant content providers. Predicting segments that will be subsequently consumed may further facilitate prefetching them in advance to improve the user experience.

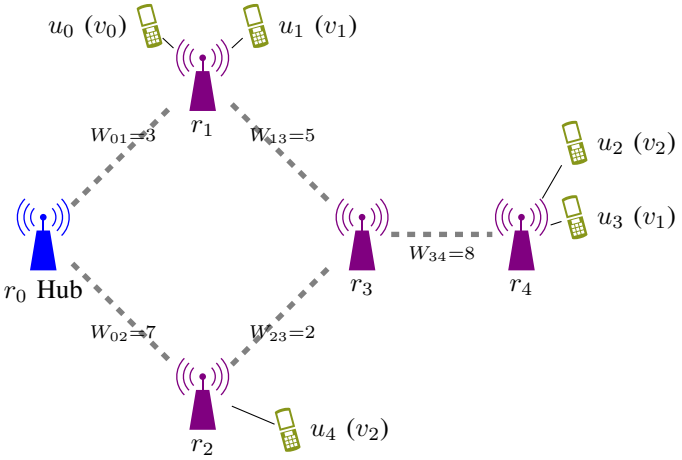

Fig. 1. A simple radio cluster that consists of 5 base station sites where r_0 is the hub. There are 5 users requesting 3 different videos. Base stations (aka routers) are equipped with radio links represented as dashed lines.

In this paper, we focus on caching in radio clusters of mobile networks to exploit the latency enhancements mentioned above. As shown in Fig.1, a base station site has multiple radio transceivers for providing access service to end-users and establishing radio links for backhauling. In a multi-hop topology, sites establish radio links to each other towards the hub. This work assumes that the sites have multi-access edge computing cloudlets and storage that facilitates caching video-on-demand segments. The main problem we address is where to locate the to-be-cached segments to minimize the total latency for video-on-demand streaming to users inside a radio cluster.

Previously in the literature, Wang et al. focused on the caching placement problem by asking the what to cache and how to cache questions and discussing two caching techniques [3]. Femtocaching [4] is another study that proposed architecture for wireless video distribution and distributed caching of the content in femto-basestations. As in our study, [5] also worked on a network where all the base stations are connected to a central controller via backhaul links and tried to optimize caching placement strategy. Differently, their model considers base station cooperation in the radio access. Dai et al. [6] has also worked on collaborative caching strategy employing resource auctions.

978-3-903176-36-2 ©2021 IFIP

To consider the effect of user-mobility on caching, Chen et al. proposed a mobility-aware caching technique in 5G ultra-dense cellular networks [7] while Wang et al. studied in content-centric wireless networks to succeed mobility-aware caching [8]. Edge caching strategy has received much attention in recent years even though this is slightly different area. Ma et al. [9] and Vu et al. [10] have studied edge caching. While Ma et al. have proposed dynamic cache content update scheduling algorithms by considering the age of information optimization of the cache content, Vu et al. [10] have considered the caching capability when designing the signal transmission and investigated multi-layer caching. Different from these studies, we do not concentrate on caching at user equipment deemed as a subdomain of edge caching.

As the main contribution of this paper, we present a solution for caching in radio clusters of mobile networks. We solve the cache placement problem inside a radio cluster. We refer to the problem as Caching in Radio Cluster Backhaul Problem (CRCBP). We formulate CRCBP as an integer problem (IP) in Section II. The typical radio cluster sizes are not large. Therefore, hubs may employ IP solvers to determine cache locations and configure the cloudlets and cache applications at the base station sites within a time budget. We present the results of numerical evaluations and discussions in Section III and then conclude the paper.

II. Caching in Radio Cluster Backhaul (CRCBP)

In this section, we present the Caching in Radio Cluster Backhaul Problem (CRCBP). We firstly define notation and assumptions. After discussing base station site model, radio link channel models and the caching model, we will present the integer linear program formulation of CRCBP.

A. Base Station Site Model

We focus on a single radio cluster that has a specific topology modelled as a graph $G{=}(\mathcal{R},\mathcal{L})$ where $\mathcal{R}$ represents the set of base station sites and $\mathcal{L}$ represents the set of links between the base stations. Base station sites consist of base transceivers and radio link transceivers. Base transceivers (BTS in 2G, NodeB in 3G or eNodeB in 4G or gNB in 5G) provide access links to users and radio link transceivers implement backhaul links. Although it is somewhat misleading, we will refer to base station sites as routers in the rest of the paper since radio link transceivers also act as routers in a multi-hop backhaul topology; this naming simplifies the presentation of CRCBP. We will assume point-to-point (PtP) links unless otherwise is stated. There is a single designated router $R_0{\in}\mathcal{R}$ in the graph that represents the hub which is the aggregation router having a large backhaul capacity. The hub is also assumed to be equipped with some computation power that will facilitate solving CRCBP.

Let u,r index the users, and the base station sites (routers) in the radio cluster, respectively. We assume that the hub has the knowledge of user-to-base station association matrix and continuously updates it. $\mathbf{A}_{ur}{=}1$ if user u is associated with router r, otherwise, it is zero. Note that, a user can be connected to at most one router ; i.e., $\sum_r \mathbf{A}_{ur}{\leqslant}1$. We do not employ this constraint in CRCBP since we assume this matrix is given and is already known by the hub.

The topology of the radio cluster is represented as the matrix $\mathbf{L}_{rs}{=}1$ if router r is connected to router s, otherwise, it is zero. Here, we do not impose any constraint on the rows of this matrix, since we can model any topology type. We consider directed graphs.

B. Link Capacities

Nodes may have one or more radio interfaces for establishing radio links. A single interface may establish multiple links shared in time using steerable antennas. Access links employ sub-6 GHz links to serve users. Although the capacity of the radio and access links are not deterministic and subject to change depending on environmental conditions, we assume the network state does not change significantly over the period in which the optimization problem run and the hub configures the network. We do not claim that the capacity of downstream radio links will be employed with maximum utilization since aggregated traffic towards the hub may overwhelm upstream capacities. We assume that the hub continuously keeps track of the user-to-base station associations inside the radio cluster, the empty space in caches of routers and the capacities of the radio links that impact the weights of the edges in the graph.

C. Caching Model

We assume that each router (r) of the radio cluster contains some storage space (C_r) employed for caching. The hub (router R_0) will solve CRCBP when a new user request comes, and then it proactively and continuously prefetches contents considering the state of the radio cluster while monitoring the subsequent requests. In this work, we assume users consume video-on-demand streams from a content provider. For bounding the solution space to a single radio cluster we set $C_0{=}\infty$, and do not model what is going on beyond the hub.

Two approaches may be employed for caching. The first approach is exploiting the programmable data plane. In this case, we assume that the router R_0 supports P4 language for altering the routing pipeline stages. It actively monitors the video streaming traffic and takes actions for caching. After solving CRCBP, it prefetches subsequent segments and places the video segments in the caches of the proper routers in the radio cluster. The second approach is the classical proxy-based caching where each router is configured as a proxy that collects the user requests and then conveys this information to the hub for solving the CRCBP. In both approaches, we assume that the hub employs software-defined networking. It is able to configure the routing or flow tables of base stations sites such that video requests are served from the closest replica. We do not consider the cache replacement strategies.

We do not concentrate on all videos stored by the content provider but the finite list of the requested ones. Therefore, the hub manages a fixed number of videos requested by users in the coverage of the radio cluster. We represent user video demands with the matrix $\mathbf{V}_{uv}$. When user u initiates the

TABLE I
NOMENCLATURE AND THE DEFAULT VALUES EMPLOYED IN THE SIMULATIONS. 1 NU REPRESENTS THE AVERAGE SIZE OF A VIDEO WHICH IS USED TO NORMALIZE CAPACITY, DELAY AND STORAGE SIZES.

Symbol	Description	Defaults
$\mathcal{R}$	Set of routers indexed with r or s	35
$\mathcal{L}$	Set of links between pairs of routers in $\mathcal{R}$	random
$\mathcal{U}$	Set of users indexed with u	50
$\mathcal{V}$	Set of requested videos indexed with v	10
T	The requested number of replicas	3
$\mathbf{A}_{ur}\in\{0,1\}$	User-to-base station association indicating whether user u is connected BS r	random
$\mathbf{V}_{uv}\in\{0,1\}$	Matrix indicating user u asks video v	random
$\mathbf{W}_{rs}$	Weight Matrix representing delays (cost) on the radio link between router r and its neighbor s	random
$\mathbf{D}_{rs}$	Total delay matrix experienced transmitting video (segment) over the radio link between router r and s	random
C_r	Cache storage capacity at router r	7 NU
s_r	Size of the video file (segment)	1 NU
$x_{rv}\in\{0,1\}$	Decision variable indicating whether video v stored on router r	
$y_{stij}\in\{0,1\}$	Decision variable set to 1 if $(i,j)\in\mathcal{R}$ link resides on the shortest path for any $(s,t)\in\mathcal{R}$ pairs of start and termination routers	

request for streaming the video v, we set $\mathbf{V}_{uv}=1$ and it is zero otherwise. We normalize storage and video sizes with respect to the size of a fixed-size segment (NU bytes) and consider unit-less normalized values for storage and video sizes.

D. ILP Formulation of CRCBP

In this formulation, we assume that all-pairs shortest paths are computed first. Given a weight matrix ($\mathbf{W}_{rs}$), an algorithm such as Dijkstra, Bellman-Ford, Floyd-Warshall or A* can be run to compute the all-pairs shortest paths. For the sake of completeness, the all-pairs shortest path formulation assuming a (weakly-)connected directed-graph is as follows.

$$\underset{y_{stij}}{\text{minimize}} \quad \sum_{s}\sum_{t}\sum_{(i,j)\in\mathcal{L}} \mathbf{W}_{ij}y_{stij} \tag{1a}$$

subject to

$$\sum_{j\in\delta^O(i)} y_{stij} - \sum_{j\in\delta^I(i)} y_{stji} = \begin{cases} -1 & \text{if } i=t \\ 1 & \text{if } i=s \\ 0 & \text{if } i\neq s, i\neq t \end{cases} \quad \forall s,t,i\in\mathcal{R} \tag{1b}$$

$$y_{stij}\in\{0,1\}, \quad \forall s,t,i,j\in\mathcal{R} \tag{1c}$$

where the objective is to minimize the total distance between any pairs of nodes. We define $\delta^O(i)=\{j\in\mathcal{R} : (i,j)\in\mathcal{L}\}$ and $\delta^I(i)=\{j\in\mathcal{R} : (j,i)\in\mathcal{L}\}$ as the set of routers to which router i has a link and the set of routers that have a link to router i, respectively. Constraint (1b) keeps track of the difference between in- and out-degrees of all destinations, all sources and all interim nodes for all source-destination pairs. The total distance matrix ($\mathbf{D}_{rs}$) can then be computed using the weight matrix $\mathbf{W}_{rs}$ and the shortest paths collated in y_{stij} where $y_{stij}=1$ if $(i,j)\in\mathcal{R}$ link resides on the shortest path for any $(s,t)\in\mathcal{R}$ pairs of source and destination routers and it is not distance but the end-to-end delays. Note that the total distance matrix shows the summation of all link weights on the path from a source to a destination. The Caching in Radio Cluster Backhaul Problem (CRCBP) aims at minimizing the video delivery latency to users. The decision variable x_{rv} is set to 1 if the video (segment) v is stored on router r, otherwise, it will be zero. Using the total distance matrix ($\mathbf{D}_{rs}$), the overall formulation of the CRCBP problem is as follows.

$$J(x^*_{rv}) = \underset{x_{rv}}{\text{minimize}} \quad \sum_{u}\sum_{v}\sum_{r}\sum_{s} \mathbf{A}_{ur}\mathbf{D}_{rs}\mathbf{V}_{uv}x_{rv} \tag{2a}$$

subject to

$$\sum_{v} x_{rv}s_v \leqslant C_r, \quad \forall r\in\mathcal{R} \tag{2b}$$

$$\sum_{r} x_{rv} \geqslant T, \quad \forall v\in\mathcal{V} \tag{2c}$$

$$x_{rv}\in\{0,1\}, \quad \forall r\in\mathcal{R}, \forall v\in\mathcal{V} \tag{2d}$$

where the objective value $J(x^*_{rv})$ if the sum of the total distances (delays) between the routers to which users are connected and the routers on which the caches are located for all requested videos v of users u. The input $\mathbf{A}_{ur}$ and $\mathbf{V}_{uv}$ matrices represent user-to-router associations and user-video demands, respectively. Constraint (2b) puts a limit on the number of videos stored on caches for not violating the storage capacity limit on routers. Constraint (2c) insists on storing the videos somewhere in the radio cluster and $T\in\mathbb{Z}^+$ is the requested number of replicas in the radio cluster. Note that, we set $C_0=\infty$, to indicate that the content will be fetched from further way caches outside the radio cluster. The overall objective of CRCBP is to reduce the latency by minimizing the total delay of users accessing videos. In CRCBP, we decide only on cache placement. Setting x_{rv} to 1 for more than one router for a video file will increase the total cost (objective value). Therefore, this formulation implicitly minimizes the number of routers where videos will be stored. Here, T plays a significant role in performance. $T=1$ is the smallest value pushing the base stations sites to keep a replica. Increasing its value will decrease the latency of individual users since a replica over a shorter path may be possible for users who request an already cached video.

E. A Toy Example

A toy configuration for CRCBP is given in Fig. 1. Assume all videos are of size 3 units (e.g., normalized with respect to NU) and storage capacity at all routers is 5 units. Note that we assume $C_0=\infty$ in the formulation. When this configuration (topology, weight matrix, users, and their video requests) are given as input to CRCBP, we first run an all-pairs shortest path algorithm and find

$$\mathbf{D}_{rs}=\begin{pmatrix} 0 & 3 & 7 & 8 & 16 \\ 3 & 0 & 7 & 5 & 13 \\ 7 & 7 & 0 & 2 & 10 \\ 8 & 5 & 2 & 0 & 8 \\ 16 & 13 & 10 & 8 & 0 \end{pmatrix}$$

TABLE II
ANALYSIS OF THE INDIVIDUAL DELAYS IN THE TOY EXAMPLE.

		T=1		T=2	
User	Video	Replica	Cost	Replica	Cost
u_0	v_0	r_1	0	r_1	0
u_1	v_1	r_3	5	r_0	3
u_2	v_2	r_4	0	r_4	0
u_3	v_1	r_3	8	r_3	8
u_4	v_2	r_4	10	r_2	0
	total		23		11
	$J(x^*_{rv})$		23		55

Then, CRCBP finds

$$x^*_{rv}=\begin{pmatrix}0&0&0\\1&0&0\\0&0&0\\0&1&0\\0&0&1\end{pmatrix}$$

When T=1, video v_0,v_1,v_2 are placed at r_1,r_3,r_4 respectively yielding a total cost of $J(x^*_{rv})$=23 units. The optimization are carried out using Gurobi Optimizer version 9.1.1, build v9.1.1rc0 on a MAC computer with 3.2 GHz Quad-Core Intel Core i5 CPU and 24 GB 1867 MHz DDR3 RAM. It takes around 1.5 msec to find the optimal solution to this toy example. When we set T=2, v_0 is stored in r_0,r_1, and v_1 is stored in r_0,r_3 and v_2 is stored in r_2,r_4. Although the objective value increases to 55 from 23, the delays experienced by some users is significantly improved as can be seen in Table II. While u_1 slightly reduces the delay, u_4's gain is significant.

III. NUMERIC EVALUATIONS AND DISCUSSIONS

In this section, we present the Monte-Carlo simulator and the numeric results and discuss them.

A. Simulator

We developed the CRCBP problem in Python (3.7.4) using Gurobi version 9.1.1 [11]. A full factorial experiment is carried out for various values of the parameters presented in Table I. We define NU as the normalization unit for simplifying the capacity, link weight and storage calculations. NU can be selected any value; we used it as the size of a video segment in our simulations. In a Monte-Carlo simulation run, a random directed graph consisting of routers in $\mathcal{R}$ is generated. Link weights of this graph are generated uniform randomly between 1 and 10 msec. Here we assume that these weights represent the delays experienced for transmitting a video segment of 1 NU over links with uniform randomly generated capacities. Any pair of routers in the radio cluster may have a directed link (asymmetric) with some edge probability p_e that takes on values 0.8, 0.9 or 1. When p_e=1 we consider full mesh topology. After generating the weighted directed graph, the shortest path between all pairs of routers is computed using the above formulation. We uniform randomly generate users in $\mathcal{U}$ and their video requests. The cache storage capacities of routers C_r are uniform randomly determined between 3 and 6 NUs; i.e., a typical cache may keep 3-6 video segments concurrently. User to base station (router) assignments are determined uniform randomly satisfying $\sum_r \mathbf{A}_{ur} \leqslant 1$, a user is connected to only one base station.

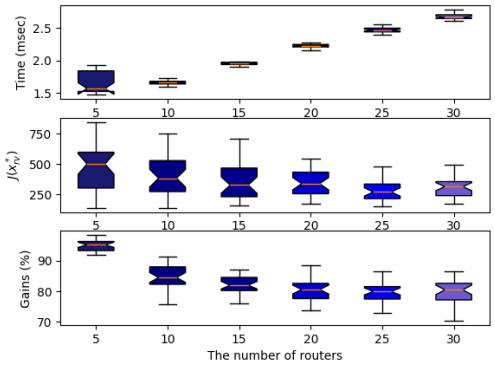

Fig. 2. Impact of the size of the radio cluster on run time (msec), objective values and the individual gain percentage.

The CRCBP problem will be solved on the hub. When users initiate streaming video content from a content distribution network, firstly they fetch the metadata description of the content. By analyzing the request by user u using one of the aforementioned caching techniques and the response metadata file describing the information about segments of the video v, the hub will set V_{uv}=1. After generating the problem instance, Gurobi is employed to solve the CRCBP by satisfying the replica count T constraint. Each parameter set is simulated 30 times and the box plots are presented in the sequel. Each figure shows the average time to solve the problem instances, the objective value of the CRCBP and the gains. The gain is computed as the percentage of reduction in the total delay for streaming the video from the caches when the same video is cached at multiple routers; i.e., T>1. If T=1 then the objective value is equal to the sum of delays experienced by users for streaming the video from the cached router on the backhaul links in the radio cluster.

B. Results and Discussions

The time to solve the CRCBP is in the order of a couple of milliseconds as can be seen in Figures 2 to 5. As the problem instance size (number of routers, users, videos, ...) grows, it takes some more time for Gurobi to find the optimal solution. As the size of the radio cluster becomes larger, CRCBP determines better alternatives for caching the videos closer to the user who demanded those videos and the average total delay (the objective value) drops from 500 msec to 200 msec. The gains with respect to caching only on one cache drops from 95% to 80% as the number of routers in the radio cluster is increased from 5 to 30 as shown in Fig.2.

In this work, we only consider users that stream video. As their number increases, the total delay increases. However, as we have already discussed in Section II-E, the delays experienced by individual users decrease significantly. This phenomenon can be observed in Fig.3. The rate of increase in

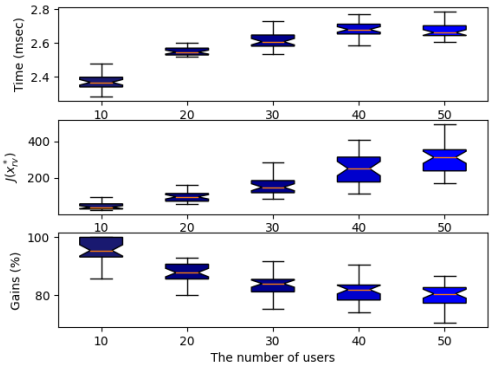

Fig. 3. Impact of the number of users on run time (msec), objective values and the individual gain percentage.

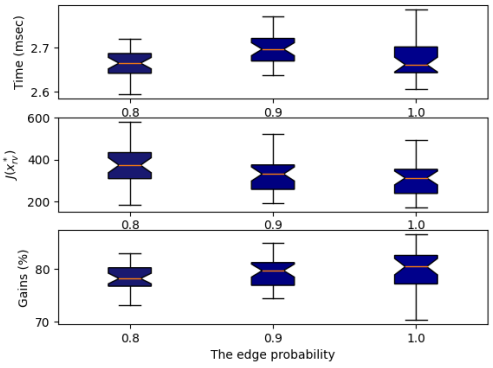

Fig. 4. Impact of the edge probability on run time (msec), objective values and the individual gain percentage.

the objective value is slower than the rate of increase in the number of users. As more and more users demand videos from the same set, the gains from replication reduce since users are uniform randomly connected to the radio cluster through different base stations.

The edge probability p_e determines the existence of a link between any pairs of routers in the topology. As p_e becomes larger, the number of links in the radio cluster increases as shown in Fig.4; the solution time does not change significantly. Because of better connectivity, shorter paths become possible which reduces the total average delays. Replication gains do not seem to be impacted because of higher link density.

As the cached video is replicated in multiple routers, the replication gains increase significantly. Note that the increase in the gains decreases as T becomes larger and larger since the storage capacity of the routers become the bottleneck as shown in Fig.5.

IV. Conclusion

When each base station at a location cannot be backhauled with a fiber cable, radio links are employed. A cluster of base stations that are connected to a hub base station through the radio links is called a radio cluster. Since the capacity of radio links are considerably lower than the fiber cables, the waste

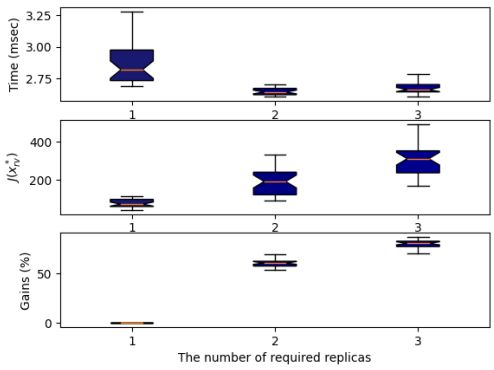

Fig. 5. Impact of the required number of replicas on run time (msec), objective values and the individual gain percentage.

of capacity due to redundant streaming of video content from the core network is not desired. Caching and cache replication can significantly improve the video streaming experience of users and the link utilization is radio clusters by bringing the content closer to users. In this paper, we presented a technique for caching on-demand video in radio clusters. We defined Caching in Radio Cluster Backhaul Problem (CRCBP), a linear program that determines the optimal cache placement considering the current state of the radio cluster, users and the video demands. The hub is expected to solve this problem, place the videos in caches and configure the routing tables. Around 80% reduction in the total average video streaming delay can be achieved.

References

[1] R. Nadiv and T. Naveh, "Wireless backhaul topologies: Analyzing backhaul topology strategies," *Ceragon White Paper*, pp. 1–15, 2010.

[2] J. Kua, G. Armitage, and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over http," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1842–1866, 2017.

[3] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[4] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, 2013.

[5] X. Peng, J.-C. Shen, J. Zhang, and K. B. Letaief, "Backhaul-aware caching placement for wireless networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[6] J. Dai, F. Liu, B. Li, B. Li, and J. Liu, "Collaborative caching in wireless video streaming through resource auctions," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 458–466, 2012.

[7] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, "Mobility-aware caching and computation offloading in 5g ultra-dense cellular networks," *Sensors*, vol. 16, no. 7, p. 974, 2016.

[8] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 77–83, 2016.

[9] M. Ma and V. W. Wong, "Age of information driven cache content update scheduling for dynamic contents in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8427–8441, 2020.

[10] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2827–2839, 2018.

[11] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com