Cache Selection in Dynamic D2D Multicast Networks Using Inhomogeneous Markov Model

Mansi Peer, Vivek Ashok Bohara, and Anand Srivastava

***Abstract*—This article presents a user spatio-temporal behavior aware cache selection framework to facilitate device-to-device multicast (D2MD) communication that minimizes the number of caches required while achieving a desired user load on the cellular network. Consequently, it alleviates the caching load on the cellular network. The optimization problem formulated to minimize the number of caches is combinatorial in nature with an exponential search space. Hence, a greedy algorithm for cache selection is proposed to reduce the search space. It has been shown that the greedy algorithm has a complexity $\mathcal{O}(K^2)$ where K is the number of users. A real-world location-based inhomogeneous Markov chain is presented to model the joint spatio-temporal behavior of the users. The diurnal variation of observable user load on the core network as well as sum-rate of non-caching users has been demonstrated for real-world location data traces. It has been shown that the proposed framework not only achieves the desired user load but also helps in improving the sum-rate of non-caching users as compared to the mobility-unaware selection of caches.**

***Index Terms*—Cache selection, d2d multicast, markov chain, real-world location information, social tagging, spatio-temporal behavior.**

I. Introduction

THE exponential increase in data traffic has led to considerable load on existing cellular networks. To alleviate the above, promising data offloading solutions such as device-to-device (D2D) communication and content caching at the network edge have been proposed recently. D2D involves communication among cellular users in proximity that paves the way for local data services [1], whereas content caching at the network edge refers to temporary storage of popular multimedia content at user devices and small-scale base stations (SBSs) to alleviate duplication of content requests at the core network [2]–[4]. It was shown in [5] that benefits of D2D can be complemented by incorporating content caching feature. However, caching at the user devices on the network edge requires the information of the mobility pattern of the users and the mobility pattern is characterized by the spatio-temporal behavior of the user [6], [7]. The recent literature on mobility-aware caching strategy for D2D network is based on the inter-contact time model [7], [8] for a pair of users. Specifically, authors in [7] solved a caching placement problem that maximizes the data offloading ratio through a greedy algorithm. They extended this work in [8] by modeling the pairwise contact pattern as an alternating renewal process and provided an improved cache placement strategy. An optimal caching policy based on user preferences was proposed in [9]. Similar to [7], [8], in [9] the problem of maximizing offloading probability was solved using a greedy algorithm. In [10], a caching strategy to minimize the cost was presented where caching took place at both SBSs and user devices. Further, a closed form expression for the average system cost was also derived. The work in [11] dealt with resource allocation for D2D multicast (D2MD) content sharing and utilized social and physical domain knowledge for D2D cluster formation for a given time instant. It was shown in [11] that exploiting D2MD at the user devices resulted in less consumption of resources at the content caching device. In [12], the authors proposed a novel approach to minimize the downloading latency and maximize the social welfare simultaneously for a socially aware D2D network. For achieving the above objectives, they efficiently selected the important users and matched the contents with users in a joint manner.

However, as evident from above, the existing works on caching strategies for D2D underlaying cellular networks do not take into account the joint spatio-temporal behavior of the users as well as fail to analyze the real-world interactions among multiple users for dynamic D2MD networks. Further, the mobility aware caching strategies that are discussed in [7], [8], [10] are primarily content placement strategies. They consider a large library of files and utilize file segmentation to store segments of files in a distributed manner. However, in scenarios where the library of files to be cached is small, for example, within a social group of users (explained later in detail), cache selection can be utilized where the complete file is stored on the selected caches. Optimal cache selection can alleviate the burden of content caching on the cellular network. Moreover, unlike file segmentation approach, caching the complete file at each cache does not require the proximity of all caches to the requesting users.

In addition to above, the authors in [7], [8] assumed the transmission rates to be the same for all D2D pairs. This is because the inter-contact time models used in these works only looked at the contact time duration, and neglected the spatial locations and spatio-temporal correlations in the user mobility pattern. However, the spatial preferences and temporal dependencies

Manuscript received November 28, 2019; revised May 7, 2020 and July 9, 2020; accepted August 21, 2020. Date of publication August 25, 2020; date of current version December 30, 2020. Recommended for acceptance by Dr. Atilla Eryilmaz. *(Corresponding author: Mansi Peer.)*

The authors are with the Wirocomm Research Group, Department of Electronics, and Communication Engineering, Indraprastha Institute of Information Technology (IIIT-Delhi), New Delhi 110020, India (e-mail: mansip@iiitd.ac.in; vivek.b@iiitd.ac.in; anand@iiitd.ac.in).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TNSE.2020.3019415

are needed to evaluate the holistic performance of content caching D2MD underlaid cellular network.

Motivated by the above, we propose a user spatio-temporal behavior aware cache selection framework for D2MD networks that minimizes the number of selected caches while achieving a desired user load on the cellular network. The selection of the minimum number of caches is formulated as an optimization problem. Further, minimizing the subset of users selected as caches can alleviate the burden of content caching on the cellular network. However, due to the unexpected occurrences of network congestion frequent optimizations will be required [13]. Assuming that the popular content needs to be updated on a daily basis.[1], an optimization is carried out at the beginning of each day to select the caches optimal on that day. In a realistic setting, the core network can equip the caches with the popular multimedia content, and on a daily basis update the content based on demand and popularity. For instance, a viral news article can be cached at selected users by the core network which can be downloaded by other requesting or non-caching users sometime later in the day. As the day progresses, more optimizations will be performed to manage the sudden variation in the user load constraint preset by the core network. However, frequent optimizations should not discard the previously selected caches. Otherwise, the cellular network will be burdened with content caching at the newly selected caching users. This will lead to a further increase in the caching load on the cellular network. In the formulated optimization problem, we take into consideration the above constraint and thus do not discard the previously selected caches. The problem is combinatorial in nature, and the complexity of the problem increases with an increase in the number of users. Hence, a greedy algorithm for cache selection is proposed that exploits the problem structure to reduce the search space. It is shown that the complexity of the greedy algorithm for K users is $\mathcal{O}(K^2)$ whereas the complexity of the widely applicable exhaustive search is $\mathcal{O}(2^{(K-1)})$. This has been discussed in detail in Section 3. To the best of our knowledge, the proposed work is the first of its kind that finds an optimal set of caches for dynamic D2MD networks that alleviates the caching load on the cellular network.

Leveraging the fact that a user prefers or requests few content files more over other files [9], the caching can be performed for a group of users who share common interests, i.e., the users are part of a social group and tend to be in each others proximity [15]. Consequently, the users in this social group may generate requests for common content files and it will be easier to accommodate a few commonly requested files at each selected cache. Hence, the proposed work also performs social tagging, i.e., a set of caches is selected from a social group and is assigned the task to serve the other members of the social group via D2MD communication. Base stations (BSs) that cover the spatial spread of the group of users can be informed a priori about the selection decision for that group. Hence, if a user of a particular social group requests for popular content file, BS will associate the user to one of the caches tagged to the user's social group. The above will be useful in a network assisted D2D peer discovery scenario as this will lead to fast peer (or cache) discovery since the search space for establishing a D2D connection is now limited to the optimal set of caches [15].

In order to capture the spatio-temporal behavior of the users, the proposed framework requires the joint mobility pattern of users. Consequently, an inhomogeneous discrete-time Markov chain [16], [17] is presented that utilizes the real-world location information to model the joint mobility pattern of users, wherein the time of the day is chosen as the line of reference. The location data was gathered from 9:00 to 19:00hrs for one semester (only working days) in a campus set-up where the user locations were broadly categorized into three buildings. Since the considered users are students, the location samples from a semester were collected to fully capture the mobility information. The duration of 9:00 to 19:00hrs was divided into 20 slots of 30 minutes each. The gathered real-world data is used as the training dataset to determine the transition probabilities of the Markov chain. The spatio-temporal correlations of each group of users are derived from the joint mobility pattern. This is explained in detail in Section 2.

Moreover, as mentioned before, the prior works [7], [8] assumed the transmission rates to be the same for all D2D pairs. However, generally, the transmission rates depend on the channel conditions of the users which may be present in different spatial locations. As 90% of the times users stay in an indoor environment [18], we use the indoor path loss models to evaluate the varying transmission rates.

A. Main Contributions

In this work, a novel cache selection framework is proposed that exploits the spatio-temporal behavior of the users to facilitate D2MD communication in a cellular network. The major contributions of this paper are as follows:

- The proposed framework minimizes the number of caches required to achieve a desired user load on the cellular network, thus alleviating the caching load on the cellular network.
- The selection of the minimum number of caches is formulated as an optimization problem. However, frequent optimizations due to the unexpected occurrences of network congestion result in frequent changes in selected caches. Consequently, the proposed framework does not discard the previously selected caches which further reduces the caching load.
- The formulated problem has an exponential search space. Hence, a greedy algorithm for cache selection with complexity $\mathcal{O}(K^2)$ is proposed that exploits the problem structure.
- An inhomogeneous discrete-time Markov chain model based on real-world location information[2] of users is presented to predict the spatio-temporal behavior.

[1] The cached content is flushed out at the end of the day, and optimization is again carried out at the beginning of the next day [14]

[2] The dataset is available at https://www.iiitd.edu.in/~wirocomm/resources /Social_Group_data.rar.

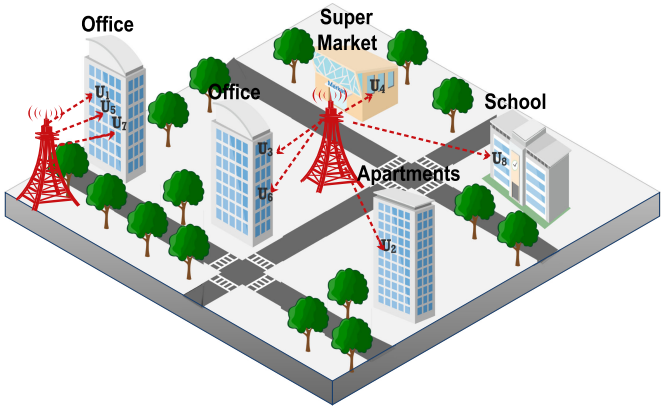


Fig. 1. Illustration of $K = 8$ users belonging to a social group spread across a geographical area consisting of two BSs.

- The proposed framework is compared to mobility-unaware cache selection. Our work exhibits that our framework outperforms mobility-unaware cache selection in terms of achievable sum-rate of non-caching users and user load on core network. It is shown that to achieve the desired user load with mobility-unaware cache selection the caching load on the cellular network should be increased.

The rest of the paper is organized as follows: Section 2 presents the system model for a D2MD underlaid cellular network and the details of the inhomogeneous Markov chain, Section 3 exhibits the formulated problem and discusses the algorithms for determining the solution, Section 4 demonstrates the simulation results with real data traces and Section 5 concludes the paper.

B. Terminologies

Below we have defined the key terms used throughout this paper:

- Cache: Cache or caching user refers to a user in the network that temporarily stores a multimedia file or content.
- Cache Selection: Cache selection refers to the process of selecting a set of caches where the multimedia content may be stored.
- User Load: It refers to the number of users that are served by the cellular network directly.
- User Mobility: User mobility describes the spatio-temporal variations in the location of a user.
- Mobility Unaware Cache Selection: It is the cache selection process where the user mobility pattern is not taken into account.

II. System Model

The system model considers a total of K users in a social group denoted by set $\mathcal{K}$ and each user is represented as $U_i, i \in [1, K]$. These users are distributed across m buildings namely $B_j, j \in [1, m]$. This scenario can depict either a single cell (BS) or a multi-cell (BSs) network depending on the geographical span of the user mobility patterns. In this paper, the terms users supported by the cellular network and user load are used interchangeably. Fig. 1 demonstrates a scenario where $K = 8$ users, belonging to a social group and accessing a popular multimedia file via unicast cellular links, are spread across a geographical area consisting of five buildings, i.e., $m = 5$ at a specific time of the day. Assuming the users supported by the cellular network that request the same popular file is equivalent to the user load, the user load for the above case will be 8. However, a more efficient approach would be to cache the popular multimedia content at some specific user (or users) such that the user load is less than 8.

In the proposed work, a set of caches is selected for a social group so that each cache is willing to share the cached content with others via D2D multicast. It is assumed that the users request the content at any time of the day but only once. Let L_p be the desired load of requesting users on the core network averaged over a day and mobility pattern of the users in $\mathcal{K}$. p is the instant of optimization and $p \in \{1, 2, 3 \ldots\}$. The set of caches has to be optimally selected to reduce the average user load to L_p. As illustrated in Fig. 1, it is considered that the members of a social group request for a popular multimedia file at the same time. In a realistic scenario, the number of users requesting the same content at a given time will be less than the total users in the group. However, the optimal selection assuming all users are requesting the same content ensures that an average user load of L_p or below is achieved irrespective of the number of requesting users. The information of the set of cache selected for the network will be shared among the BSs that span the m buildings by a central controller where the central controller is the entity where the cache selection decision is being taken.

Let us denote the average signal-to-interference and noise ratio (SINR) value as $SINR^j_{ik}$ for users U_i and U_k present in building B_j. The SINR values will be characterized by the layout of building B_j, users' relative positions, and interference. The interference will be due to the cellular user[3] C_R whose uplink (UL) resource block (RB) is being reused by a D2D multicast group. $SINR^j_{ik}$ can be defined as follows:

$$SINR^j_{ik} = \frac{P_t \Gamma}{I_{avg} + N_o BW}, \tag{1}$$

where P_t is the transmit power of U_i and $\Gamma = \mathbb{E}|h_{ik}|^2$. $h_{ik} \sim \mathcal{CN}(0, \beta^{-1}_{ik})$, β^{-1}_{ik} is the pathloss between U_i and U_k and $|h_{ik}|^2$ is the channel gain. $\mathbb{E}(.)$ denotes the expectation operator. Further, N_o is the noise spectral density and BW is the bandwidth of one RB. $I_{avg} = \mathbb{E}_d[P_t|g|^2]$ is the average interference at U_k due to transmission from C_R located at a distance d from user k. $g \sim \mathcal{CN}(0, \beta^{-1}_{C_R,k})$ where $\beta_{C_R,k}$ is the path loss between C_R and U_k. The pathloss values have been determined using WINNER II models [19]. Let us assume that the cellular network guarantees an average signal-to-noise ratio (SNR), SNR_{cell} to its users in order to maintain a pre-defined quality of service (QoS). Now, user U_i which is present in building B_j

[3] It is assumed that the RBs assigned to a cellular user in UL is reused by one D2D group.

TABLE I
FEASIBILITY SETS FOR EACH BUILDING

SNR_{cell}	B_1	B_2	B_3
20 dB	$U_1 - \{U_5\}$ $U_2 - \{U_5\}$ $U_3 - \{U_4, U_5\}$ $U_4 - \{U_3, U_5\}$ $U_5 - \{U_1, U_2, U_3, U_4\}$	$U_1 - \{U_2, U_3, U_4, U_5\}$ $U_2 - \{U_1, U_3, U_4, U_5\}$ $U_3 - \{U_1, U_2, U_4, U_5\}$ $U_4 - \{U_1, U_2, U_3, U_5\}$ $U_5 - \{U_1, U_2, U_3, U_4\}$	$U_1 - \{U_2, U_3, U_5\}$ $U_2 - \{U_1, U_3, U_4, U_5\}$ $U_3 - \{U_1, U_2, U_4, U_5\}$ $U_4 - \{U_2, U_3, U_5\}$ $U_5 - \{U_1, U_2, U_3, U_4\}$
25 dB	$U_1 - \{U_5\}$ $U_2 - \{\}$ $U_3 - \{U_5\}$ $U_4 - \{U_5\}$ $U_5 - \{U_1, U_3, U_4\}$	$U_1 - \{U_2, U_3, U_4, U_5\}$ $U_2 - \{U_1, U_3, U_4, U_5\}$ $U_3 - \{U_1, U_2, U_4, U_5\}$ $U_4 - \{U_1, U_2, U_3, U_5\}$ $U_5 - \{U_1, U_2, U_3, U_4\}$	$U_1 - \{U_2, U_5\}$ $U_2 - \{U_1, U_3, U_5\}$ $U_3 - \{U_2, U_4\}$ $U_4 - \{U_3, U_5\}$ $U_5 - \{U_1, U_2, U_4\}$

will be receiving content from U_k via D2MD only if $SINR^j_{ik} > SNR_{cell}$, i.e., QoS obtained through D2MD will be better than that obtained through cellular. Hence, in other words, we can state that if U_i belongs to the feasibility set of U_k, U_i can be served by U_k via D2MD. The users who are not served by D2MD will receive the content from the cellular network. It may be noted that each building can have a maximum of K feasibility sets depending on the geographical distribution of the K users in them.

For instance, the role of SNR_{cell} in the generation of these feasibility sets has been demonstrated for a real campus set-up where $m = 3$ and $K = 5$. The matrix $SINR^j$ contains the average pairwise SINRs of K = 5 for the j^{th} building based on P_t = 23 dBm, N_o = -173 dBm/Hz, BW = 180 kHz and d which is uniformly distributed in the range (10, 250) m. The SINR matrices are given below where $SINR^j_{ik}$ is the $(i,k)^{th}$ element of $SINR^j$:

$$SINR^1 = \begin{bmatrix} \text{NA} & 16.72 & 13.23 & 19.35 & 25.03 \\ 16.72 & \text{NA} & 15.46 & 15.36 & 22.95 \\ 13.23 & 15.46 & \text{NA} & 20.16 & 39.92 \\ 19.35 & 15.36 & 20.16 & \text{NA} & 34.34 \\ 25.03 & 22.95 & 39.92 & 34.34 & \text{NA} \end{bmatrix}, \quad (2)$$

$$SINR^2 = \begin{bmatrix} \text{NA} & 31.27 & 33.06 & 27.98 & 36.66 \\ 31.27 & \text{NA} & 25.96 & 40.78 & 29.45 \\ 33.06 & 25.96 & \text{NA} & 28.21 & 33.23 \\ 27.98 & 40.78 & 28.21 & \text{NA} & 40.70 \\ 36.66 & 29.45 & 33.23 & 40.70 & \text{NA} \end{bmatrix}, \quad (3)$$

$$SINR^3 = \begin{bmatrix} \text{NA} & 33 & 24 & 17.73 & 31.72 \\ 33 & \text{NA} & 26.35 & 24.74 & 27.50 \\ 24 & 26.35 & \text{NA} & 29.77 & 20.44 \\ 17.73 & 24.74 & 29.77 & \text{NA} & 29.29 \\ 31.72 & 27.50 & 20.44 & 29.29 & \text{NA} \end{bmatrix}, \quad (4)$$

where NA is used to denote the SINR value for the link from user U_i to itself. For instance, the SINR corresponding to the link between U_1 and U_5 in building B_1 is $SINR^1_{15}$ = $SINR^1_{51}$ = 25.03dB. Using (2)-(4), the feasibility sets observed for each building at SNR_{cell} = 20dB, 25dB for the campus set-up are presented in Table I. It can be seen with the increase in SNR_{cell}, the number of elements in feasibility sets reduces. This is because as the threshold SNR_{cell} increases, some pairwise SINR values will fail to meet the increased threshold.

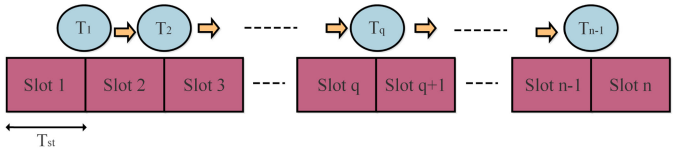


Fig. 2. Illustration of dependency of transition matrices on slot boundaries.

A. Inhomogeneous Discrete-Time Markov Chain

As stated before, for optimal cache selection, one of the key elements is to deduce the mobility pattern of the users. However, as the proposed work involves D2MD where multiple users are involved, there is a need to analyze the joint mobility pattern of the users. This will help in analyzing the interaction of multiple users at the same time. We present a discrete-time Markov chain to model the joint mobility pattern of the users. The state of the network is denoted by a vector of size $1 \times K$ where element i of the state vector is the building visited by U_i and can have m possible values, i.e., $B_1, \ldots, B_m$. Due to the mobility of the users, the network will transit from one state to another. The network is assumed to stay in any state 'a' for time μT_{st}, where T_{st} is the minimum time the network will stay in any state 'a' and μ is an integer that depends on the user mobility. As the mobility pattern is modeled using a Markov chain, the next state will only depend on the current state of the network. A complete day is divided into n slots of T_{st} duration each. The transition probability from one state to another has been derived from the observed joint mobility pattern of the users. There are inherent spatio-temporal correlations in the observed joint mobility pattern, i.e., given a time slot, certain buildings will be more preferred by users than others. Since the building preferences are time-dependent, each slot will observe a different set of states, and each slot boundary will have a different transition matrix. Consequently, the Markov chain is inhomogeneous [17]. The transition matrix corresponding to q^{th} slot boundary is denoted as $T_q, q \in [1, n-1]$ and $(n-1)$ are the number of slot boundaries as illustrated in Fig. 2. Further, each element $p_{a,b}$ in a transition matrix denotes the transition probability from state 'a' in q^{th} slot to 'b' in $(q+1)^{th}$ slot. The joint location information must be gathered for a large number of days to capture mobility information completely. The state vectors are obtained for each time slot. For instance, if there are K users in the network and m buildings which means in a given slot there are m^K possible state vectors per slot. However, because of the regularity in the human mobility pattern the set of observed state vectors in q^{th} slot will be considerably small and will depend on the time of the day. Let us denote the observed state vectors in the q^{th} slot as X_q. To better illustrate the user proximity with the change in state vectors, each observed state vector can be represented as m contact graphs where each graph corresponds to each building. The connection between one user to another will be decided by observing its feasibility set in that building. If user U_i lies in the feasibility set of U_k for building B_j there will be an edge in the graph of B_j connecting nodes U_i and U_k. These graphs will be dynamic in nature depending on the time of the day. Fig. 3 plots the graphs corresponding to the

Algorithm 1. Constructing Transition Matrix, T_q

Step 1: First record the observed sets X_q and X_{q+1}.
Step 2: Let $Y_q = (X_q \cup X_{q+1})$ denote the set of states at q^{th} slot boundary and $M_q = |Y_q|$.
Step 3: Determine the transition probability of each state a to b, i.e., $p_{a,b}\ \forall a, b \in Y_q$. These transition probabilities will constitute T_q.

unique observed state vectors for slot 1 with T_{st} = 30 minutes, $K = 5, m = 3$ with SNR_{cell} = 20dB for the real-world campus set-up considered in this work. It can be seen that $|X_1|$ = 8, which is significantly less than 3^5 = 243. Due to the small value of $|X_n|$ for any social group of users, the number of transitions possible at a slot boundary will also be low. As the states in the Markov chain increases, complexity increases. Therefore, to reduce the complexity, we have restricted the size of transition matrix T_q to $M_q \times M_q$ where M_q is the number of states observed in $X_q \cup X_{q+1}$ [20]. The complexity can be reduced further by the methods suggested in [17], which is out of the scope of this work. Further, it may be noted that certain entries in the transition matrices will be zero due to the spatio-temporal preferences of the users.

Algorithm 1 describes the steps for constructing the transition matrix T_q. At first, the observed state vectors in the q^{th} and $(q+1)^{th}$ slots are recorded. The state vectors in $X_q \cup X_{q+1}$ are the observable states of the Markov chain at the q^{th} boundary. This is followed by evaluation of transition probabilities from one state to another. The steps are repeated for all the slot boundaries.

III. Problem Formulation And Solution

Once the transition matrices have been constructed, the next task is the optimal selection of set of caches taking into account spatio-temporal behavior of the users. The optimal selection minimizes the number of selected caches subject to user load L_p or below on the core cellular network and must not discard the previous cache selections. Let v_0 be the state of the network at the instant of optimization and $v = (v_1, v_2, \ldots, v_r, \ldots, v_{n_0})$ be the observable sequence of states that occurs after T_{st} minutes of v_0, where a state vector in v can be present more than once and n_0 is the number of slots remaining that day (i.e., future time slots). For instance, when the optimization is carried out in slot 1, v_0 is the state of network in slot 1, $n_0 = (n-1)$ and all observable sequences v can be obtained from $Y_1, \ldots, Y_{n-1}$. Let $\rho(v|v_0)$ denote the probability of sequence v given v_0 has occurred and can be given as:

$$\begin{aligned}\rho(v|v_0) &= \rho[(v_1, v_2, \ldots, v_{n_0})|v_0], \\ &= P[v_1|v_0]P[v_2|(v_0, v_1)]\ldots...P[v_r|(v_0, v_1, v_{r-1})] \\ &\ldots P[v_{n_0}|(v_0, v_1, v_{n_0-1})],\end{aligned} \tag{5}$$

where $P[v_r|(v_0, v_1, \ldots, v_{r-1})]$ is the probability of state v_r given sequence (v_0, v_1,..., v_{r-1}). As Markov chain model is assumed, $\rho(v|v_0)$ can be written as follows:

$$\begin{aligned}\rho(v|v_0) &= P[v_1|v_0]P[v_2|v_1]\ldots...P[v_{n_0}|(v_{n_0-1})], \\ &= \prod_{r=1}^{n_0} p_{v_{r-1}, v_r}.\end{aligned} \tag{6}$$

The optimization problem formulated to minimize the number of selected caches at p^{th} ($p \in \{1, 2, 3\ldots\}$) instant of optimization is given in (7) where C_p is the set of selected caches and $|C_p|$ denotes cardinality of C_p. Let state $v_l \in \{v_0\} \cup v$ where l ranges from 0 to n_0. $U_{C_p}^{v_l}$ are the non-caching users in v_l state served by users in C_p via D2D multicast. $U_{C_p}^{v_l}$ will be determined by the feasibility sets of the users in C_p. For $K = 5$, $C_p = \{U_1, U_2\}$, $v_l = (B_1, B_1, B_1, B_1, B_1)$ and feasibility sets as given in Table I, $U_{C_p}^{v_l} = 1$. L.H.S of (7b) is the expected user load when set C_p is selected. As the content request at the cache can be generated at any time of the given day, while selecting the set of caches, we are averaging over the user load corresponding to current time slot and future time slots. Given a observable sequence v, first the user load is averaged over each state $v_l \in \{v_0\} \cup v$. Then, the user load is averaged over all the observable sequences. Constraint (7b) assures that the expected user load is less than L_p on the core network. As mentioned in Section 1, frequent optimizations are required in the network to tackle with the sudden occurrences of network congestion. In such scenarios, to assure that the previously selected caches are not discarded, a constraint needs to be applied. In other words, for a given K, the optimal solution with cardinality, let us say, $|C_{p-1}^*|$ at desired user load L_{p-1} must be a subset of optimal solution with cardinality $|C_p^*|$ at desired user load L_p where $L_{p-1} > L_p$. Constraint (7c) takes care of the above requirement. It may be noted that the cardinality of optimal solution, C_p^* cannot go beyond $\lfloor K/2 \rfloor$ because at a given time no more than $\lfloor K/2 \rfloor$ D2MD groups can exist and has been accounted for using (7d). After slot 1 of the day, more optimizations will only be required when there is a decrease in the load constraint, because only then more caches will have to be selected.

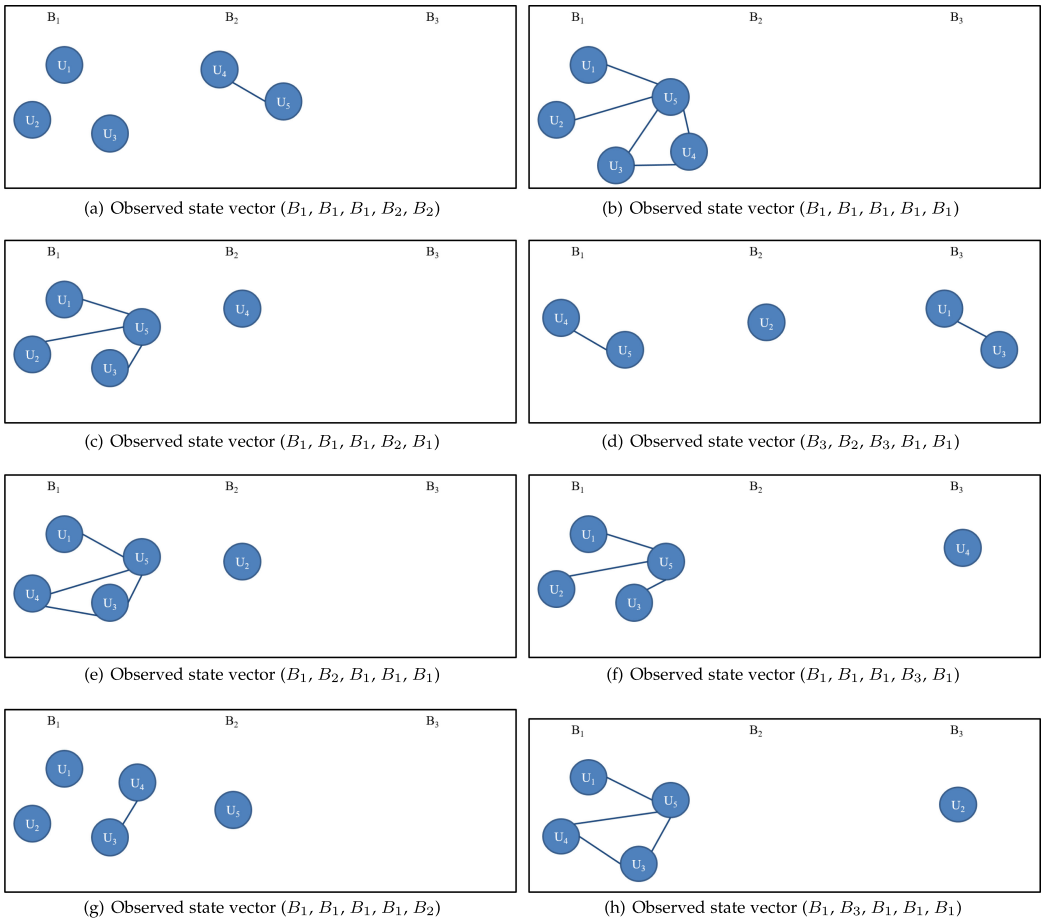
$$\text{minimize} \qquad \big|C_p\big|, \tag{7a}$$

$$\text{subject to} \qquad \mathbb{E}_v\Big[\ \mathbb{E}_{v_l}\Big[(K - |C_p|) - U_{C_p}^{v_l}\Big]\Big] \leq L_p, \tag{7b}$$

$$C_{p-1}^* \subseteq C_p,\ C_0 = \emptyset \tag{7c}$$

$$|C_p| \leq \lfloor K/2 \rfloor \tag{7d}$$

The problem in (7a)-(7d) qualifies to be a combinatorial problem. The optimization will done in two stages: (1) Stage 1: optimization at the beginning of each day, i.e., p = 1, (2) Stage 2: optimization due to sudden network congestion during the day, i.e., $p > 1$. For stage 1, constraint (7c) will not hold as there will be no previously selected caches at the beginning of the day. One approach to solve the optimization problem is to perform an exhaustive search. However, exhaustive search has a complexity of $\mathcal{O}(2^{(K-1)})$ (see Appendix A in the supplementary document) due to the exponential search space. As a consequence, in the proposed work, a greedy algorithm for cache selection is proposed that exploits the problem structure and reduces the search space.



(a) Observed state vector $(B_1, B_1, B_1, B_2, B_2)$ (b) Observed state vector $(B_1, B_1, B_1, B_1, B_1)$

(c) Observed state vector $(B_1, B_1, B_1, B_2, B_1)$ (d) Observed state vector $(B_3, B_2, B_3, B_1, B_1)$

(e) Observed state vector $(B_1, B_2, B_1, B_1, B_1)$ (f) Observed state vector $(B_1, B_1, B_1, B_3, B_1)$

(g) Observed state vector $(B_1, B_1, B_1, B_1, B_2)$ (h) Observed state vector $(B_1, B_3, B_1, B_1, B_1)$

Fig. 3. Contact graphs for illustrating the user proximity in a real campus set-up where $K = 5$, $m = 3$.

A. Greedy Algorithm for Cache Selection

As the cardinality of the selected set of caches needs to be minimized such that the constraints are met, the search will start from the lowest cardinality i.e., 1. If the load constraint (7b) is not met at a given cardinality of set C_p then the cardinality of the set C_p has to be increased to search for the optimal solution. Exploiting this behavior, we have proposed a greedy algorithm for cache selection that takes into account only those candidate sets with cardinality c which have $\mathcal{Q}$, with cardinality $c-1$, as a subset. $\mathcal{Q}$ is the set that fails to be optimal but has a minimum expected user load among the candidate sets at cardinality $c-1$. The greedy algorithm has a search space of the order $\mathcal{O}(K^2)$ (see Appendix A in the supplementary document) whereas the exhaustive search is of the order $\mathcal{O}(2^{(K-1)})$. Hence, the greedy algorithm reduces the search space of the formulated problem.

B. Illustration of Exhaustive Search and Greedy Algorithm with Real Data

In this subsection, we have compared the performance of the exhaustive search with the greedy algorithm. We have considered a campus set-up with m = 3 and K users in a social group. The joint mobility pattern of these K users is modeled as a discrete-time inhomogeneous Markov chain. As discussed in Section 2, transition matrices are constructed using the real-world location data of the users in the campus set-up.

1) Stage 1 - Optimization at the Beginning of Each Day: Algorithm 2 presents the exhaustive search to compute the optimal set of caches. Let $|C_1|$ be the cardinality of the selected set of caches. At cardinality $|C_1|$, there will be $\binom{K}{|C_1|}$ candidate sets. Algorithm 2 evaluates the load corresponding to all the candidate sets. Then, it is checked whether there is a candidate set that has the load less than the load requirement, i.e., L_1. If there is no such candidate set then the value of $|C_1|$ is incremented by 1 and the above process is repeated exhaustively. However, if there are candidate sets that meet the load requirement then optimal solution is obtained by selecting that candidate set that has the minimum load and the algorithm stops. Further, while incrementing $|C_1|$, $|C_1| \leq \lfloor K/2 \rfloor$ condition should always be met. If the load requirement is not met until $|C_1| = \lfloor K/2 \rfloor$ then it implies that no optimal solution is feasible and algorithm stops. Further, Algorithm 3 presents the proposed greedy algorithm for cache selection. Similar to Algorithm 2, $|C_1|$ is the cardinality of the selected set of caches. If a solution is not obtained at cardinality $|C_1|$ then $\mathcal{Q}$ will be updated with the candidate set having minimum user load and $|C_1|$ will be incremented by 1. The above process will be repeated until $|C_1| = \lfloor K/2 \rfloor$ or optimal solution is obtained.

Algorithm 2. Stage 1: Exhaustive Search

Step 1: Initialize $|C_1| = 0$.
Step 2: Increment value of $|C_1|$.
if $|C_1| > \lfloor K/2 \rfloor$ **then**
  Exit search. No optimal solution.
else
  Calculate observable user load for all $\binom{K}{|C_1|}$ sets of $|C_1|$ caches.
  if Observable Load $\leq L_1$ for at least one set **then**
    Optimal solution obtained
  else
    Go to Step 2
  end if
end if

Algorithm 3. Stage 1: Greedy Algorithm

Step 1: Initialize $\mathcal{Q} = \emptyset$ and $|C_1| = 0$.
Step 2: Increment value of $|C_1|$.
if $|C_1| > \lfloor K/2 \rfloor$ **then**
  Exit search.
else
  Consider only those candidates in Step 3 which have $\mathcal{Q}$ as a subset.
end if
Step 3: Calculate observable user load for candidate sets of $|C_1|$ caches.
if Observable Load $\leq L_1$ for at least one set **then**
  Solution obtained
else
  Update $\mathcal{Q}$ with the candidate having minimum user load. Go to Step 2.
end if

TABLE II
TABLE FOR $|C_1|$= 1 AND K = 6

Candidate Sets	User Load
$\{U_1\}$	3.476
$\{U_2\}$	3.618
$\{U_3\}$	3.644
$\{U_4\}$	3.518
$\{U_5\}$	2.784
$\{U_6\}$	2.726

TABLE III
TABLE FOR $|C_1|$ = 2 AND K = 6

Candidate Sets	Load	Candidate Sets	Load
$\{U_1, U_2\}$	2.037	$\{U_2, U_6\}$	1.590
$\{U_1, U_3\}$	1.845	$\{U_3, U_4\}$	1.996
$\{U_1, U_4\}$	1.809	$\{U_3, U_5\}$	1.553
$\{U_1, U_5\}$	1.765	$\{U_3, U_6\}$	1.568
$\{U_1, U_6\}$	1.671	$\{U_4, U_5\}$	1.584
$\{U_2, U_3\}$	1.921	$\{U_4, U_6\}$	1.506
$\{U_2, U_4\}$	1.696	$\{U_5, U_6\}$	1.265
$\{U_2, U_5\}$	1.671		

TABLE IV
TABLE FOR $|C_1|$= 1 AND K = 8

Candidate Sets	User Load
$\{U_1\}$	4.943
$\{U_2\}$	4.906
$\{U_3\}$	4.812
$\{U_4\}$	5.325
$\{U_5\}$	4.128
$\{U_6\}$	4.537
$\{U_7\}$	5.203
$\{U_8\}$	4.275

Let $K = 6$, p = 1, $L_1 = 2$, $|C_0|$ = 0 and $C_0 = \emptyset$. Using Algorithm 2, C_1 is initialized as 0. At step 2, $|C_1|$ is incremented to 1. Using Table II, load corresponding to all the candidate sets is evaluated. However, all candidate sets have load greater than L_1. Hence, $|C_1|$ is incremented to 2. It can be observed from Table III that there are multiple candidate sets that have load lower than L_1. Consequently, we select the candidate set with minimum load equal to 1.265 as the optimal set of caches, i.e., C_1^* = $\{U_5, U_6\}$.

Further, on applying the greedy algorithm, $|C_1|$ is initialized as 0 and $\mathcal{Q} = \emptyset$. At step 2 of Algorithm 3, $|C_1|$ = 1. On incrementing $|C_1|$, only candidate sets with $\mathcal{Q}$ as a subset will be considered. It can be observed from Table II, no candidate set has load lower than L_1. Hence, C_1 is incremented to 2. It is shown in Table II, set $\{U_6\}$ has minimum value of user load. Therefore, $\mathcal{Q} = \{U_6\}$. Considering only those candidate sets in Table III that have $\mathcal{Q}$ as a subset, there are multiple sets that have load less than L_1. The candidate set with the minimum load equal to 1.265 is the solution of the greedy algorithm, $C_1^{prop} = \{U_5, U_6\}$. In this example, $C_1^* = C_1^{prop} = \{U_5, U_6\}$.

Similarly, for $K = 8$ and $L_1 = 2$, the optimal solution is evaluated as $\{U_5, U_6, U_7\}$, using Tables IV, V and VI, for exhaustive search as well as greedy algorithm. Further, for $K = 10$ and L_1 = 1, the solutions for the exhaustive search and greedy algorithm are obtained as $C_1^* = \{U_3, U_6, U_7, U_9\}$ and $C_1^{prop} = \{U_5, U_6, U_7, U_9\}$.[4] Hence, the cardinality of the solutions obtained using exhaustive search and greedy algorithm are same, however, the set of selected caches is different. For $K = 10$ and $L_1 = 1.8$, the optimal solution using exhaustive search is obtained as $C_1^* = \{U_1, U_4, U_6\}$ with load 1.77. However, the minimum load at cardinality three using greedy algorithm is 1.82. As the gap between the minimum load at cardinality three using exhaustive search and greedy algorithm is small, the solution using greedy algorithm is obtained at cardinality four with $C_1^{prop} = \{U_5, U_6, U_7, U_9\}$. However, the exhaustive algorithm has a search space of $\left(\binom{K}{1} + \binom{K}{2} + \binom{K}{3}\right) = 175$ candidate sets whereas the greedy algorithm has $(K + (K-1) + (K-2) + (K-3)) = 34$ candidate sets. The gap between the search spaces of the exhaustive search and greedy algorithm widens with K.

In order to validate the efficacy of the proposed algorithm for even higher number of users, we have done a detailed comparison of exhaustive search and proposed cache selection algorithm for $K = 20$. The comparison is also presented for different load constraints. We have tabulated the solutions[5] for $K = 20$ using

[4] See Appendix B for the load values of the candidate sets.
[5] User load values corresponding to each of the candidate sets for Stage 1 are available at https://www.iiitd.edu.in/∼wirocomm/resources/Social_Group_data.rar.

TABLE V
TABLE FOR $|C_1|$= 2 AND K = 8

Candidate Sets	Load	Candidate Sets	Load
$\{U_1, U_2\}$	2.981	$\{U_3, U_5\}$	2.315
$\{U_1, U_3\}$	2.946	$\{U_3, U_6\}$	2.328
$\{U_1, U_4\}$	2.540	$\{U_3, U_7\}$	2.625
$\{U_1, U_5\}$	2.596	$\{U_3, U_8\}$	2.609
$\{U_1, U_6\}$	2.446	$\{U_4, U_5\}$	2.490
$\{U_1, U_7\}$	2.890	$\{U_4, U_6\}$	2.787
$\{U_1, U_8\}$	3.131	$\{U_4, U_7\}$	3.534
$\{U_2, U_3\}$	2.553	$\{U_4, U_8\}$	2.334
$\{U_2, U_4\}$	2.681	$\{U_5, U_6\}$	2.175
$\{U_2, U_5\}$	2.506	$\{U_5, U_7\}$	2.465
$\{U_2, U_6\}$	2.634	$\{U_5, U_8\}$	2.296
$\{U_2, U_7\}$	2.706	$\{U_6, U_7\}$	2.753
$\{U_2, U_8\}$	2.690	$\{U_6, U_8\}$	2.484
$\{U_3, U_4\}$	2.693	$\{U_7, U_8\}$	2.440

TABLE VI
TABLE FOR $|C_1|$= 3 AND K = 8

Candidate Sets	Load	Candidate Sets	Load	Candidate Sets	Load
$\{U_1, U_2, U_3\}$	1.715	$\{U_1, U_6, U_8\}$	1.787	$\{U_3, U_4, U_7\}$	1.956
$\{U_1, U_2, U_4\}$	1.421	$\{U_1, U_7, U_8\}$	1.712	$\{U_3, U_4, U_8\}$	1.493
$\{U_1, U_2, U_5\}$	1.828	$\{U_2, U_3, U_4\}$	1.578	$\{U_3, U_5, U_6\}$	1.350
$\{U_1, U_2, U_6\}$	1.693	$\{U_2, U_3, U_5\}$	1.640	$\{U_3, U_5, U_7\}$	1.509
$\{U_1, U_2, U_7\}$	1.596	$\{U_2, U_3, U_6\}$	1.575	$\{U_3, U_5, U_8\}$	1.550
$\{U_1, U_2, U_8\}$	2.081	$\{U_2, U_3, U_7\}$	1.403	$\{U_3, U_6, U_7\}$	1.456
$\{U_1, U_3, U_4\}$	1.668	$\{U_2, U_3, U_8\}$	1.712	$\{U_3, U_6, U_8\}$	1.618
$\{U_1, U_3, U_5\}$	1.734	$\{U_2, U_4, U_5\}$	1.609	$\{U_3, U_7, U_8\}$	1.531
$\{U_1, U_3, U_6\}$	1.606	$\{U_2, U_4, U_6\}$	1.784	$\{U_4, U_5, U_6\}$	1.559
$\{U_1, U_3, U_7\}$	1.793	$\{U_2, U_4, U_7\}$	1.831	$\{U_4, U_5, U_7\}$	1.778
$\{U_1, U_3, U_8\}$	2.028	$\{U_2, U_4, U_8\}$	1.453	$\{U_4, U_5, U_8\}$	1.446
$\{U_1, U_4, U_5\}$	1.562	$\{U_2, U_5, U_6\}$	1.637	$\{U_4, U_6, U_7\}$	2.084
$\{U_1, U_4, U_6\}$	1.393	$\{U_2, U_5, U_7\}$	1.459	$\{U_4, U_6, U_8\}$	1.565
$\{U_1, U_4, U_7\}$	1.818	$\{U_2, U_5, U_8\}$	1.678	$\{U_4, U_7, U_8\}$	1.593
$\{U_1, U_4, U_8\}$	1.465	$\{U_2, U_6, U_7\}$	1.534	$\{U_5, U_6, U_7\}$	1.293
$\{U_1, U_5, U_6\}$	1.453	$\{U_2, U_6, U_8\}$	1.703	$\{U_5, U_6, U_8\}$	1.471
$\{U_1, U_5, U_7\}$	1.631	$\{U_2, U_7, U_8\}$	1.453	$\{U_5, U_7, U_8\}$	1.393
$\{U_1, U_5, U_8\}$	1.753	$\{U_3, U_4, U_5\}$	1.618	$\{U_6, U_7, U_8\}$	1.503
$\{U_1, U_6, U_7\}$	1.431	$\{U_3, U_4, U_6\}$	1.596		

exhaustive search and proposed algorithm in Table VII. Let us analyze one of the scenarios given in Table VII. Suppose at Stage 1 at a given day, $L_1 = 8$. On one hand, using the exhaustive search, $C_1^* = \{U_{16}, U_{19}\}$ and $|C_1^*| = 2$. On the other hand, proposed algorithm provides $C_1^{prop} = \{U_{15}, U_{16}\}$ and $|C_1^{prop}| = 2$. Hence, for this scenario, the cardinality of the selected set of caches is equal for both the approaches. Similarly, observations were also made for other scenarios in Table VII. It exhibits that the load requirement also has an impact on the efficacy of the proposed algorithm.

In the exhaustive search, at a cardinality c of candidate sets, the candidate set with minimum load that meets the load constraint is selected as optimal solution. We observed that the minimum load at cardinality c obtained using the greedy algorithm is sometimes equal to that of exhaustive search (e.g., $K = 6$, $L_1 = 2$ and $K = 8$, $L_1 = 2$) or slightly higher (e.g., $K = 10$, L_1 = 1, 1.8). In cases where the minimum load is same, exhaustive and greedy algorithm will give the same solution. However, when there is a small difference in the minimum load values, the cardinality in greedy algorithm needs at most to be incremented by 1. Hence, based on our observations, the solution using the greedy algorithm will be $|C_p^{prop}| \in \{|C_p^*|, |C_p^*| + 1\}$. Further, we have demonstrated

Algorithm 4. Stage 2: Exhaustive Search

Step 1: Initialize $|C_p| = |C_{p-1}^*|$ and $C_p = C_{p-1}^*$.
Calculate user load for set C_p.
if Observable load $\leq L_p$ **then**
  Optimal C_p^* is achieved.
else
  Go to Step 2.
end if
Step 2: Increment value of $|C_p|$.
if $|C_p| > \lfloor K/2 \rfloor$ **then**
  No optimal solution.
else
  Calculate observable user load for all $\binom{K}{|C_p|}$ sets of $|C_p|$ caches.
  if Observable load $\leq L_p$ for a set in $Select_1$ **then**
    if C_{p-1}^* is a subset for a set in $Select_2$ **then**
      Optimal solution C_p^*.
    else
      Go to Step 2
    end if
  else
    Go to Step 2
  end if
end if

the search space reduction that can be achieved by the greedy algorithm. Consequently, in Section 4, we utilize greedy algorithm for the analysis of the cache selection framework.

2) Stage 2 - Optimization Due to Sudden Network Congestion: In a scenario where sudden network congestion is detected, the desired load, L_p on the network due to the social group will be lowered in the optimization problem at the p^{th} instant. Algorithm 4 presents the exhaustive search for Stage 2. Unlike Algorithm 2, Algorithm 4 initializes $|C_p| = |C_{p-1}^*|$ and $C_p = C_{p-1}^*$ such that the previously selected set of caches is not discarded. Then, the algorithm checks whether $|C_p| = |C_{p-1}^*|$ will meet the load constraint. If the load constraint is met, the algorithm will stop else $|C_p|$ will be incremented by 1. After this, similar to Algorithm 2, Algorithm 4 will carry out exhaustive search for the optimal solution. In Algorithm 4, $Select_1$ consists of all the candidate sets at a specific cardinality. $Select_2$ contains those sets, present in $Select_1$, that meet the load constraint. Further, the proposed algorithm for Stage 2 is illustrated in Algorithm 5. Initially, $|C_p| = |C_{p-1}^{prop}|$ and $C_p = C_{p-1}^{prop}$ such that the previously selected set of caches is not discarded. Then, the algorithm checks whether $|C_p| = |C_{p-1}^{prop}|$ will meet the load constraint. If the load constraint is met, the algorithm will stop else $|C_p|$ will be incremented by 1. After this, similar to Algorithm 3, Algorithm 5 will determine the solution.

Now, let us say, for $K = 6$, due to sudden network congestion there will be a second instant of optimization with $L_2 = 1$ at the beginning of the 10^{th} slot of the specific day under consideration. As mentioned above, another optimization needs to be performed, and v_0 will now be the state at the 10^{th} slot. From the example above, C_1^*= C_1^{prop}= $\{U_5, U_6\}$. For the exhaustive search, Algorithm 4 is used. $|C_2|$ is initialized as $|C_1^*|$ = 2. As the load constraint is not met at $|C_2|$ = 2, $|C_2|$ is

TABLE VII
RESULTS FOR $K = 20$

Scenario	Load Requirements	Exhaustive Search	Proposed Algorithm
1.	$L_1 = 11$	$C_1^* = \{U_{15}\}, \lvert C_1^*\rvert = 1$	$C_1^{prop} = \{U_{15}\}, \lvert C_1^{prop}\rvert = 1$
	$L_2 = 9$	$C_2^* = \{U_{15}, U_{16}\}, \lvert C_2^*\rvert = 2$	$C_1^{prop} = \{U_{15}, U_{16}\}, \lvert C_1^{prop}\rvert = 2$
2.	$L_1 = 8$	$C_1^* = \{U_{16}, U_{19}\}, \lvert C_1^*\rvert = 2$	$C_1^{prop} = \{U_{15}, U_{16}\}, \lvert C_1^{prop}\rvert = 2$
	$L_2 = 6$	$C_2^* = \{U_{16}, U_{19}\}, \lvert C_2^*\rvert = 2$	$C_2^{prop} = \{U_{15}, U_{16}, U_{19}\}, \lvert C_2^{prop}\rvert = 3$
3.	$L_1 = 5$	$C_1^* = \{U_6, U_7, U_{19}\}, \lvert C_1^*\rvert = 3$	$C_1^{prop} = \{U_{15}, U_{16}, U_{19}\}, \lvert C_1^{prop}\rvert = 3$
	$L_2 = 3$	$C_2^* = \{U_6, U_7, U_{19}, U_5\}, \lvert C_2^*\rvert = 4$	$C_1^{prop} = \{U_{15}, U_{16}, U_{19}, U_{17}\}, \lvert C_1^{prop}\rvert = 4$

TABLE VIII
TABLE FOR $|C_2| = 3$ AND $K = 6$ IN CASE OF RE-OPTIMIZATION AT 10^{th} SLOT

Candidate Sets	User Load	Candidate Sets	User Load
$\{U_1, U_2, U_3\}$	0.887273	$\{U_2, U_3, U_4\}$	0.996364
$\{U_1, U_2, U_4\}$	0.898182	$\{U_2, U_3, U_5\}$	0.985455
$\{U_1, U_2, U_5\}$	1.058182	$\{U_2, U_3, U_6\}$	0.949091
$\{U_1, U_2, U_6\}$	1.134545	$\{U_2, U_4, U_5\}$	1.025455
$\{U_1, U_3, U_4\}$	1.014545	$\{U_2, U_4, U_6\}$	0.781818
$\{U_1, U_3, U_5\}$	0.978182	$\{U_2, U_5, U_6\}$	1
$\{U_1, U_3, U_6\}$	1.087273	$\{U_3, U_4, U_5\}$	1.087273
$\{U_1, U_4, U_5\}$	1.029091	$\{U_3, U_4, U_6\}$	0.890909
$\{U_1, U_4, U_6\}$	0.876364	$\{U_3, U_5, U_6\}$	0.785455
$\{U_1, U_5, U_6\}$	1.021818	$\{U_4, U_5, U_6\}$	0.82901

incremented to 3. The candidate sets at $|C_2|$ = 3 are presented in Table VIII and are stored in $Select_1$. Out of the sets in $Select_2$, the set that has C_1^* as subset and has minimum load is obtained as $\{U_3, U_5, U_6\}$. Further, using Algorithm 5 for the greedy cache selection, C_1^{prop}= $\{U_3, U_5, U_6\}$.

Further, let there be a change in the load requirement from L_1 =8 to $L_2 = 6$ for $K = 20$ as shown in Table VII. On using the exhaustive search the solution will be the same as before, i.e., $C_2^* = \{U_{16}, U_{19}\}$ and $|C_2^*| = 2$. However, proposed algorithm will result in $C_2^{prop} = \{U_{15}, U_{16}, U_{19}\}$ and $|C_2^{prop}| = 3$. Let there be a change in the load requirement to $L_2 = 6$. This will trigger a new optimization. On using the exhaustive search the solution will be the same as before, i.e., $C_2^* = \{U_{15}, U_{16}\}$ and $|C_2^*| = 2$. However, proposed algorithm will result in $C_2^{prop} = \{U_{15}, U_{16}, U_{19}\}$ and $|C_2^{prop}| = 3$. Therefore, the cardinality of selected set of caches differs by 1.

In general, the BSs present within the spatial spread of the social group of the K users can be informed about the selected caches. In this work, we have not directed the algorithm to the scenario where the transition matrix at a given slot boundary is also dynamic in nature. However, the work can easily be extended to such a scenario.

IV. RESULTS

In this section, we demonstrate the results of the proposed cache selection framework for a specific working day at the campus using the greedy algorithm. Further, we have compared the performance of the mobility-unaware cache selection approach to the proposed cache selection framework. For a fair comparison, in mobility-unaware cache selection, the number of selected caches is kept equal to $|C_p^{prop}|$; however, any set can be randomly selected out of all the candidate sets [21]. We have considered social groups of size K = 5, 10, 20 to demonstrate the impact of the number of users in a social group on the selection of caches. The cardinality of the set of selected caches quantifies the caching load on the cellular network. The optimization has been performed in slot 1 of the specific day under consideration unless otherwise stated.

Algorithm 5. Stage 2: Greedy Algorithm

Step 1: Initialize $|C_p| = |C_{p-1}^{prop}|$, $\mathcal{Q} = C_{p-1}^{prop}$ and $C_p = C_{p-1}^{prop}$.
Calculate user load for set C_p.
if Observable load $\leq L_p$ **then**
  Solution C_p^{prop} is achieved.
else
  Go to Step 2.
end if
Step 2: Increment value of $|C_p|$.
if $|C_p| > \lfloor K/2 \rfloor$ **then**
  Exit search.
else
  Consider only those candidates in Step 3 which have $\mathcal{Q}$ as a subset.
end if
Step 3: Calculate observable user load for candidate sets of $|C_p|$ caches.
if Observable Load $\leq L_p$ for at least one set **then**
  Solution obtained
else
  Update $\mathcal{Q}$ with the candidate having minimum user load. Go to Step 2.
end if

Table IX exhibits the effect of SNR_{cell} on the user load on the cellular network for $K = 5, 10, 20$ when $L_1 = 2, 4, 4$ respectively. It can be observed that with the increase in SNR threshold from 20dB to 25dB either user load on the core network increases or cardinality of the selected set of caches increases. This is because with increased threshold the size of feasibility sets reduces as explained in Section 2, and the load on the network will increase, or the network will require more caches to achieve the predefined user load constraint. As a result, for $K = 5$, U_5 is selected as the cache when threshold SNR_{cell} = 20dB whereas U_2 and U_5 are selected as caches at SNR_{cell} = 25dB.

Fig. 4 shows the achievable sum-rate of the non-caching users using the proposed spatio-temporal user behavior aware caching framework and mobility-unaware cache selection approach for $K = 5$ at SNR_{cell} = 20dB. The sum-rate expression for each slot is given as follows:

TABLE IX
TABLE SHOWING IMPACT OF SNR_{cell}

K	L_1	SNR_{cell}	Caches Selected	Load
5	2	20 dB	1 (U_5)	1.956
		25 dB	2 (U_2, U_5)	1.334
10	4	20 dB	2 (U_5, U_6)	2.781
		25 dB	2 (U_5, U_6)	3.204
20	4	20 dB	2 (U_{15}, U_{16}, U_{19})	3.765
		25 dB	2 (U_{15}, U_{16}, U_{19})	3.92

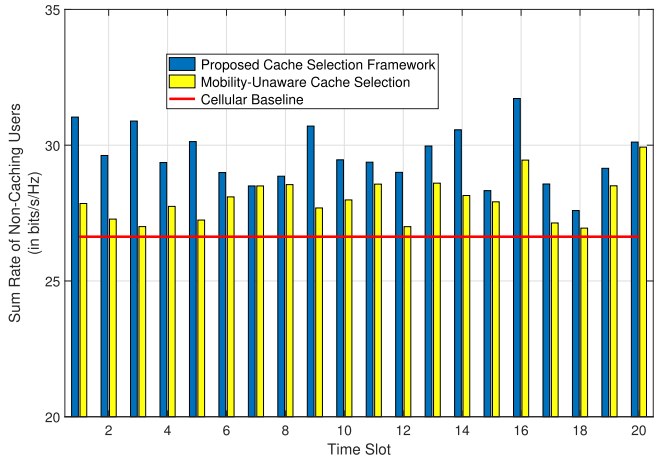


Fig. 4. Sum-rate of non-caching users w.r.t. time slot when SNR_{cell} = 20dB.

$$S_{rate} = \mathbb{E}_{v_l}\left[\sum_{i=1}^{(K-|C_1^{prop}|)} R_i\right], \tag{8}$$

where R_i is the rate achievable at the i^{th} user which may be served by one of the caches in C_1^{prop} via D2MD or cellular network. The sum rate is averaged over the observable states v_l in a given slot. From Table IX, it is known that U_5 should be selected as the cache at $K = 5$ and $SNR_{cell} = 20$ dB, i.e., there are 4 non-caching users. In a conventional cellular communication with unicast links per user, the baseline cellular sum-rate performance will be $4\times\log_2(1+100)$ = 26.632 bits/s/Hz. It is obvious from Fig. 4 that the former performs worse than the proposed cache selection framework. This is because mobility-unaware cache selection fails to take into account the effect of spatio-temporal behavior and other constraints. Lesser sum-rate in mobility-unaware cache selection means more users are opting for cellular communication. Hence, the user load on the cellular network is higher in mobility-unaware cache selection. To achieve the desired user load on the cellular network, more caches have to be selected in the mobility-unaware cache selection approach. However, this will increase the caching load on the cellular network.

Fig. 5 presents the set of caches required for $L_1 = 2$. It can be seen that as the number of users in the social group increases, the number of caches required will also increase. An interesting insight can be obtained through Fig. 5, i.e., even though the number of caches needed for K = 8 and 9 are equal, however, the caching users are not the same. This is because the total number

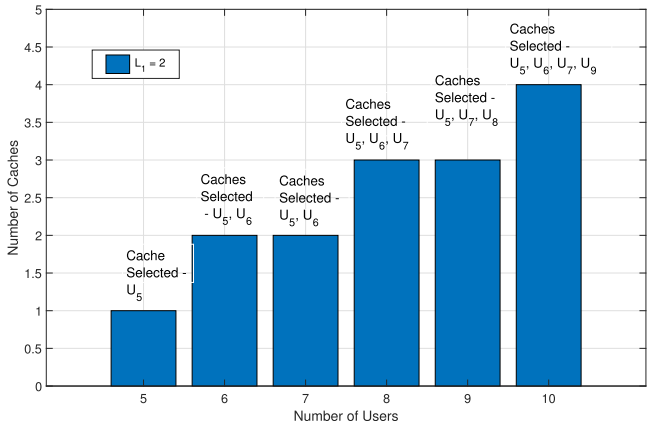


Fig. 5. Number of Caches w.r.t. number of users to achieve L_1 = 2.

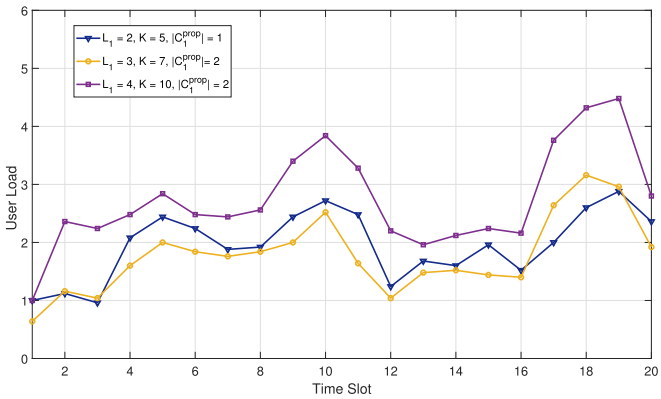


Fig. 6. User load on cellular network w.r.t. time slot for K = 5, 7, 10.

of caches as well as the users that are to be selected as cache are the function of the joint mobility patterns and pairwise SINR values of users. Hence, for K = 8 and 9 sets $\{U_3, U_6, U_7\}$ and $\{U_5, U_7, U_8\}$ are selected as caches respectively.

Fig. 6 shows the per time slot user load on the cellular network at $K = 5, 7, 10$ and $L_1 = 2, 3, 4$ respectively when set C_1^{prop} of caches is utilized. The user load on the network varies with the time of the day as in each time slot different set of state vectors are observable depending on the transition matrices. This is also evident from the discussion in Section 2. Further, with a change in K and the desired user load L_1, the user load on the network for each slot also shows a variation. This is because C_1^{prop} obtained using the greedy algorithm varies with the value of K and L_1. Fig. 7 shows the impact of K = 20 and the number of caches selected on the minimum user load on the cellular network. It can be observed that as the number of caches increases, the minimum user load on cellular network decreases. However, an increase in the number of caches will strain the cellular network at the time of content caching.

Now, let us say, for $K = 6$, due to sudden network congestion, another optimization needs to be performed and v_0 is now the state of the network at the 10^{th} slot. $L_1 = 2$ is reduced to $L_2 = 1$ (example with $K = 6$ and $L_1 = 2$ illustrated in Section 3) at the beginning of the 10^{th} slot of the specific day other consideration. In case the constraint (7c) introduced in Section 3 is not applied

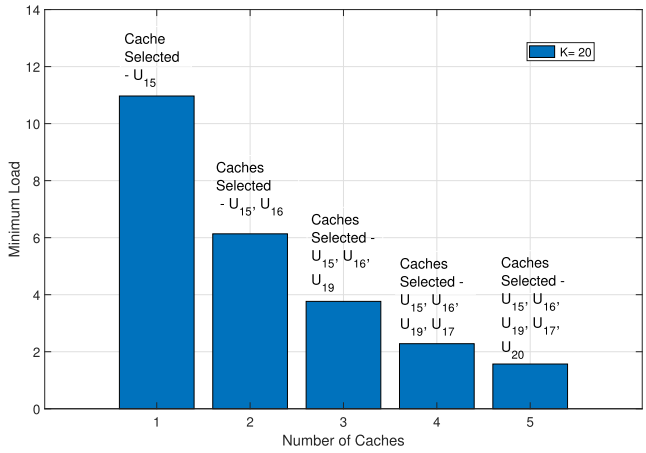


Fig. 7. Minimum user load on cellular network w.r.t. number of caches for $K = 20$.

to our proposed framework, using Table VIII, set $\{U_2, U_4, U_6\}$ with minimum user load less than L_2 will be selected as caches. Thus, two new caches are added to the network. On the other hand, on incorporating the constraint (7c) and using $\mathcal{Q} = C_1^{prop} = \{U_5, U_6\}$, $\{U_3, U_5, U_6\}$ will be selected as the set of caches. Consequently, only one new cache, i.e., U_3 needs to be added. This implies that our proposed framework also helps in alleviating the caching load due to frequent optimizations.

V. Conclusion

In this work, a greedy algorithm for cache selection is proposed to solve the combinatorial optimization problem of selecting a set of the minimum number of caches to achieve the desired user load for a D2MD network. The set of caches is selected by utilizing real-world location information to obtain the spatio-temporal behavior of the users. The proposed work has been shown to alleviate the caching load on the cellular network. Moreover, the proposed framework does not discard the previously selected caches. Further, a discrete-time inhomogeneous Markov chain is presented to model the joint mobility pattern for the users in the D2MD network is developed. The selected caches are tagged to their social group and are responsible for doing D2D multicast to disseminate the popular multimedia files to the non-caching users. It is observed that the proposed algorithm is computationally less intensive with complexity $\mathcal{O}(K^2)$ as compared to complexity $\mathcal{O}(2^{(K-1)})$ of the exhaustive search. Further, the presented optimization has been shown to perform better than mobility-unaware cache selection.

Acknowledgment

The authors would like to thank Visvesvaraya research fellowship, Department of Electronics and Information Technology, Ministry of Communication and IT, Government of India, for providing financial support for this article.

References

[1] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surv. Tut.*, vol. 20, no. 3, pp. 2133–2168, Third Quarter 2018.

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

[3] S. T.ul Hassan, M. Bennis, P. H. J. Nardelli, and M. Latva-Aho, "Modeling and analysis of content caching in wireless small cell networks," in *Proc. Int. Symp. Wireless Commun. Syst.*, Aug. 2015, pp. 765–769.

[4] W.Wen, Y. Cui, F. Zheng, and S. Jin, "Random caching based cooperative transmission in heterogeneous wireless networks," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.

[5] M. Ji, G. Caire, and A. F. Molisch, "Optimal throughput-outage tradeoff in wireless one-hop caching networks," *CoRR*, vol. abs/1302.2168, 2013. [Online]. Available: http://arxiv.org/abs/1302.2168

[6] R. Amer, M. M. Butt, M. Bennis, and N. Marchetti, "Inter-cluster cooperation for wireless D2D caching networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6108–6121, Sept. 2018.

[7] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug. 2017.

[8] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Exploiting mobility in cache-assisted D2D networks: Performance analysis and optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5592–5605, Aug. 2018.

[9] B. Chen and C. Yang, "Caching policy optimization for D2D communications by learning user preference," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–6.

[10] I. Pappalardo, G. Quer, B. D. Rao, and M. Zorzi, "Caching strategies in heterogeneous networks with D2D, small BS and macro BS communications," in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 1–6.

[11] L. Feng *et al.*, "Resource allocation for 5G D2D multicast content sharing in social-aware cellular networks," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 112–118, Mar. 2018.

[12] J. Li *et al.*, "On social-aware content caching for D2D-enabled cellular networks with matching theory," *IEEE Int. Things J.*, vol. 6, no. 1, pp. 297–310, Feb. 2019.

[13] A. Orsino *et al.*, "Time-dependent energy and resource management in mobility-aware D2D-empowered 5G systems," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 14–22, Aug. 2017.

[14] S. Hosny, A. Eryilmaz, and H. E. Gamal, "Impact of user mobility on D2D caching networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[15] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, "A survey on socially-aware device-to-device communications," *IEEE Commun. Surv. Tut.*, vol. 20, no. 3, pp. 2169–2197, Third Quarter 2018.

[16] A. Konrad, B. Y. Zhao, A. D. Joseph, and R. Ludwig, "A markov-based channel model algorithm for wireless networks," *Wireless Networks*, vol. 9, no. 3, pp. 189–199, May 2003. [Online]. Available: https://doi.org/10.1023/A:1022869025953

[17] E. B. Iversen, J. K. Møller, J. M. Morales, and H. Madsen, "Inhomogeneous Markov models for describing driving patterns," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 581–588, Mar. 2017.

[18] Machine learning, "Navigating the great indoors and safer vehicles," Jan. 2017. [Online]. Available: https://www2.deloitte.com/au/en/pages/media-releases/articles/what-the-future-holds-tmt-predictions-240117.html

[19] M. Peer, V. A. Bohara, and A. Srivastava, "On the performance of network-assisted indoor device-to-device communication using location awareness and realistic path loss models," in *Proc. IEEE 28th Annu. Int. Symp. Personal, Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–7.

[20] V. L. Erickson, M. A. Carreira-Perpiñán, and A. E. Cerpa, "Occupancy modeling and prediction for building energy management," *ACM Trans. Sen. Netw.*, vol. 10, no. 3, pp. 42:1–42:28, May 2014. [Online]. Available: http://doi.acm.org/10.1145/2594771

[21] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, G. Pei, and A. Srinivasan, "Cellular traffic offloading through opportunistic communications: A case study," in *Proc. 5th ACM Workshop Challenged Networks*, ser. CHANTS '10. New York, NY, USA: ACM, 2010, pp. 31–38. [Online]. Available: http://doi.acm.org/10.1145/1859934.1859943