OPTIMAL MOTILITY AWARE CACHING POLICY IN DEVICE TO DEVICE NETWORKS

[1] Smt. D.Lalitha Kumari, [2]P. Vennela.

[1] Assistant Professor, [2] M Tech Student,
[1] Dept. of ECE,
[1] JNTUA College of Engineering, Ananthapuramu, India.

Abstract- In this paper we use gadget to gadget communication by using caching, so that the cellular traffic will be decreased and the burden on the backhaul links will be reduced. However, user motility in existing systems is not included, so the data can be lost during transmission when the user is on movement. So in this paper, user motility is taken in to consideration and we implement the optimal caching placement policy in order to improve the information offloading proportion in the D2D network rather than the cellular network. In any case, here issue falls in the class of monotone submodular amplification over a matroid limitation, so we use a greedy algorithm which makes the choices that are best at that movement. Simulation results will give the performance of the implemented caching placement policy which include motility and we observe how the data transferring rate will change for various parameters like number of users, velocity of the users and capacity of the cache memory.

Index Terms — caching, gadget–to-gadget communication, client motility, matroid constraint, submodular function.

I. INTRODUCTION

Now a days every body prefer smart gadgets like cell phones, laptops etc., which are used to send the high data e.g., videos, HD photos so due to that the traffic in the cellular system is increasing and burden on backhaul links are increasing, which connect base stations (BSs) with the core network [1]. To decrease the backhaul load, one promising methodology is to reserve well known substance at BSs and client gadgets , [2] so portable clients can get the required substance from neighborhood BSs or close-by client gadgets without using backhaul joins.

Most past examinations on remote storing systems expected settled system topologies [3], [4]. Be that as it may, client portability is a characteristic element of remote systems, which changes the system topologies after some time. In this manner, it is basic to consider the client versatility design. Then again, client versatility can likewise be a valuable element to misuse, as it will build the correspondence chances of moving clients.. In this, we implement a powerful motility caching policy in gadget to-gadget (D2D) storing systems to offload data from base stations.

II. SYSTEM MODEL

In this paper we use the gadget to gadget transmission network and communication can be done between the gadgets only if the gadgets are in the transmission range of the transmitter gadget. Here the connection time for two gadgets is characterized as the time that they can contact with each other, i.e., they may trade files in that connection time. Then the inter connection time is the time of connection between two connection times. Here the take N_g number of gadgets, and the set of the gadgets is represented as G={1 ,2, 3,…, N_g} and the connection rate between two users let us say u and v as $\lambda_{u,v}$.

Let us take a library of N_{doc} documents, whose set is signified as D ={ 1, 2, 3, …, N_{doc}}. Here Rateless Fountain coding is used, where each document is encoded into an extensive number of various fragments, and it very well may be recouped by gathering a specific number of encoded sections. Likewise, it tends to be ensured that there is no redundant encoded portion in the system. In particular, we expect that each document is encoded into different fragments, each with measure i bytes, and record d can be recuperated by gathering I_d encoded sections. Note that the estimation of I_d relies upon the extent of document d. It is expected that every versatile gadget holds a specific measure of capacity limit with regards to reserving, which can store at most C encoded portions. The quantity of encoded portions of document d reserved in client u is indicated as $y_{u,d}$.

Portable clients will ask for records in the document library in view of their requests. For straightforwardness, we accept that the solicitations of the considerable number of clients take after a similar dissemination, and document d is asked for by one client with probability Pd , with the overall probability of all documents is $\sum_{d \in D} \text{Pd} = 1$.

At the point when a client u asks for a document d, it will begin to download encoded fragments of record d from the experienced clients, and furthermore check its own reserve.

We expect that the length of each connection of clients u and v is $t_{u,v}$ seconds, j is total bytes and the rate of transmission is $r_{u,v}$ bps of client v to client u. we consider that fragments that can be transmitted inside one connection from client v to client I is $A_{u,v}$ = [($t_{u,v}$ * $r_{u,v}$) / (j)] fragments. We additionally expect that there is a postpone requirement, signified as Td. In the event that client u can't gather in any event I_d diverse encoded fragments of document d inside Td, it will ask for the rest of the sections from the BS. For instance, as appeared in Fig (1), gadget 3 asks for some fragments of document from gadget 1 and after that remaining fragments from the document is collected from gadget 2 ,here gadget 3 is in moving so that it will collect fragments from both gadget 1 and gadget 2. Where the other gadgets 5 and 6 will collect from gadget 2 , gadget 4 will collect from gadget 1 and it should be done in time t where t < T^d otherwise required document will be collected by the cell tower.

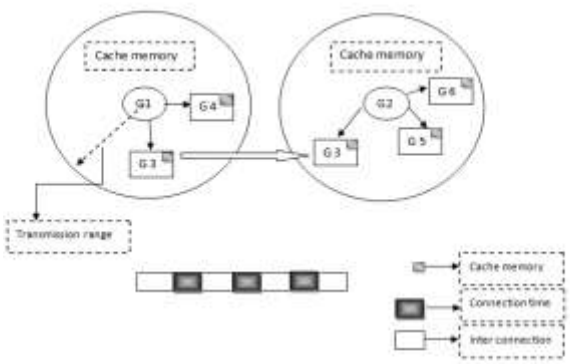


Fig (1): system model with $N_g = 6$, $N_{doc} = 5$, $C = 5$, $A = 1$, $I_d = 4$ for $d \in D$ and G for gadgets

III. PROBLEM FORMULATION

In this paper, to boost the information offloading proportion, i.e., the level of the asked for information that can be conveyed through D2D joins, as opposed to by means of the BS, we examine the reserving arrangement system. In the event that the D2D connections can offload more information from the BS, it will prompt higher spatial reuse proficiency and furthermore fundamentally lessen the backhaul trouble. In particular, for client I, the information offloading proportion is characterized as, in the event that the D2D connections can offload more information from the BS, it will prompt higher spatial reuse proficiency and furthermore fundamentally lessen the backhaul trouble. In particular, for client I, the information offloading proportion is characterized as

$$F_u = \mathbb{E}_{d\epsilon D}\left[\mathbb{E}_{w_{u,d}}\left[\frac{\min(w_{u,d}, I_d)}{I_d}\middle| gadget\ u\ requests\ document\ d\right]\right]$$

$$= \sum_{d\epsilon D} pd\left\{\frac{\mathbb{E}_{w_{u,d}}[\min(w_{u,d}, I_d)]}{I_d}\right\} \quad (1)$$

Where $w_{u,d} = \sum_{v\epsilon G} w_{u,d}^{v}$ indicates the quantity of encoded fragments of record d that can be gathered by client u inside time Td, of which $w_{u,d}^{v}$ portions can be gathered from client v. Let $P_{u,v}$, with $u \epsilon G$ and $v \epsilon G$, signify the quantity of contact times for client u and v inside time Td. Since $A_{u,v}$ sections can be transmitted from client v in one connection time, client u can maximally download $A_{u,v}P_{u,v}$ portions from client v inside Td. Accordingly, amid time Td, client u can gather $w_{u,d} = \sum_{v\epsilon G}\min(A_{u,v}P_{u,v}, y_{u,d})$, encoded portions of document f from all clients. At that point, the information offloading proportion for client u is

$$F_u = \sum_{f\epsilon \mathcal{F}} \frac{pd}{I_d}\left\{\mathbb{E}\left[min\left(\sum_{v\epsilon G}\min(A_{u,v}P_{u,v}, y_{u,d}), I_d\right)\right]\right\} \quad (2)$$

Likewise, the general normal information offloading proportion of the considerable number of clients is $\frac{1}{N_g}\sum_{u\epsilon G} F_u$,and the versatility mindful reserving arrangement issue is figured as

$$\max_{X} \frac{1}{N_g}\sum_{u\epsilon G} F_u, \quad (3)$$

$$s.t. \quad \sum_{d\epsilon D} y_{v,d} \le C, \forall\, v \in G \quad (3a)$$

$$y_{v,d} \in N, \forall\, v \in G\ and\ d \in D \quad (3b)$$

$$y_{v,d} \le I_d, \forall\, v \in G\ and\ d \in D \quad (3c)$$

where limitation (3a) infers that every client gadget can't store more than C fragments, (3b) ensures that each encoded section is either completely put away or not put away at all in one client gadget, and (3c) infers that, since document f can be recouped by I_d encoded portions, it is excess to store more than I_d fragments of record d at every client.

In the accompanying, we will reformulated first issue (3) as a monotone submodular augmentation issue over a matroid requirement. Right off the bat, the ground set O is characterized as $O = \{l_{v,d,i} | v \in G, d \in D\ and\ 1 \le i \le I_d\}$, and each storing situation L is a subset of O. On the off chance that the component $l_{v,d,i}$ is in L , then it implies that the i-th portion of record d is reserved at client v. In particular, the connection between $y_{v,d}$, with $v \in G$ and $d \in D$, and the storing situation L ⊆ O is

$$y_{v,d} = |L \cap O_{v,d}| \quad (4)$$

where $O_{v,d}$ is an arrangement of all I_d sections of record d that might be reserved at client v characterized as

$$O_{v,d} = \{l_{v,d,i} | 1 \le i \le I_d\} \quad (5)$$

Next, we revise the target capacity of issue (3) as a component of subsets of O as

$$g(L) = \frac{1}{N_g}\sum_{u\epsilon G}\sum_{d\epsilon D}\frac{p_d}{I_d}$$

$$\times E\left[\left(\sum_{v\epsilon G}\min(A_{u,v}P_{u,v}, |L \cap O_{v,d}|), I_d\right)\right] \quad (6)$$

The formula g(L) characterized in (6) is a monotone submodular work , the primary trouble is to determine the gain of including one component into L and the limitation in the first issue is ended up being a matroid imperative is explained in [5]. To total up, issue (3) can be reformulated as

$$\max_{L\in K}\frac{1}{N_g}\sum_{u\epsilon G}\sum_{d\epsilon D}\frac{p_d}{I_d}$$

$$\times E\left[\left(\sum_{v\epsilon G}\min(A_{u,v}P_{u,v}, |L \cap O_{v,d}|), I_d\right)\right] \quad (7)$$

which is a monotone submodular expansion issue over a matroid requirement.

IV. IMPLEMENTATION METHOD

Greedy Algorithm for Motility Caching

For issue (7), the algorithm calculation gives a compelling arrangement, and has been demonstrated to file a 1/2-estimate [6]. A randomized calculation was implemented in [7], which accomplishes a higher guess proportion. In the recreation, we will demonstrate that the algorithm calculation performs near the ideal reserving system.

The algorithm calculation right off the bat sets the storing arrangement L as a vacant set, and afterward, it adds a component as per the need esteems while fulfilling the matroid requirement. The procedure proceeds until the point that no more components can be included. The need estimation of a component $l_{v,d,i}$, signified as $g_{v,d,i}^{d}$, is characterized as the gain of including $l_{v,d,i}$ into the storing arrangement L , given in [10], when, $L \in K, l_{v,d,i} \in O - L\ and\ L \cup \{l_{v,d,i}\} \in K$.

The trouble in the ravenous calculation is to refresh the need esteem productively. In outline, the implemented method calculation is depicted in Algorithm, where O^r means the staying set, including the components that might be included into the reserving position L. At the point when the component $\{l_{v^*,d^*,i^*}\}$ is added to L, it ought to be erased from O^r. At the point when client v^* has just put away C fragments, no greater section can be put away in client v^*, and in this way, every one of the components in O_{v^*} ought to be erased from O^r. Also, in the wake of adding another component $\{l_{v^*,d^*,i^*}\}$ to L , the need esteems ought to be refreshed.

Algorithm The Greedy Algorithm

1) $Set\ L = \emptyset \leftrightarrow Set\ y_{v,d} = 0, \forall v \in G\ and\ d \in D.$
2) $O^r = O.$
3) $Intialize\ the\ priority\ values$

$$\{g^d_{v,d,i} | v \in G, d \in D, 1 \le i \le I_d\}$$

4) while $|L| < N_g \times C\ do$
5) $l_{v^*,d^*,i^*} = arg \max_{l_{v,d,i} \in O^r} g^d_{v,d,i^*}$
6) $Set\ L = L \cup \{l_{v^*,d^*,i^*}\} \leftrightarrow add\ y_{v^*,d^*}\ by\ 1.$
7) $O^r = O^r - \{l_{v^*,d^*,i^*}\}.$
8) $if\ |L \cap O_{v^*}| = C\ then$
9) $O^r = O^r - O_{v^*}$
10) end if
11) $Update\ the\ priority\ values$

$$\{g^d_{v,d^*,i} | v \in G, d \in D, 1 \le i \le I_{d^*}, l_{v,d^*,i} \in O^r\}$$

by using divide and conquer method in [10]
12) end while

V. SIMULATION RESULTS

In this area, reenactment results will be given to assess the execution of the implemented motility caching procedure.

1) Greedy motility caching technique: the imperfect arrangement of issue (3) utilizing the greedy calculation.

2) Random storing methodology: the irregular reserving procedure [5], where the probabilities of every client to store fragments of various records are relative to the document ask probabilities.

In the accompanying, we accept that the document ask for likelihood takes after a Zipf appropriation with parameter gamma, i.e., $P_d = \frac{d^{-gamma}}{\sum_{d \in D} d^{-gamma}}$ [3], [8], [9]. The quantity of encoded sections to recuperate each document, i.e., I_d, is haphazardly chosen in [1, I_{max}]

Here the performance of implemented method mainly depends on number of gadgets used caching memory size as shown in Fig (2) as number of gadgets increases the information offloading proportion also increases, and in the Fig (3) as the memory size of the cache memory increases the information offloading proportion also increases with gamma as 1.2 in Fig (2) and gamma as 0.6 and 1.2 in Fig (3).

The runtime for the proposed method with the number of gadgets in the network is shown in the Fig (4) where we can see that runtime will be increased when the gadgets in the network increases with gamma as 0.6.

Analayis of implemented method according to gamma values is shown in Fig(5) and Fig(6) with different time delays 120 sec and 600 sec respectively in order to observe the improvement in the information offloading proportion with gamma values from 0.6 to 1.2 and we can oberve that as gamma value increases information offloading proportion improves.

Here we are taking gadget motility into consideration so we will also calculate how the information offloading proportion varies according to gadget velocity and the average network connection rate as shown in fig(7) and fig (8).

So from Fig (7) and Fig (8) we can say that information offloading proportion improves when gadget velocity and connection rate increases.

Hence we found how to improve the information offloading proportion by using the implemented method with gamma values as gamma values increases most of the information is collected.

.

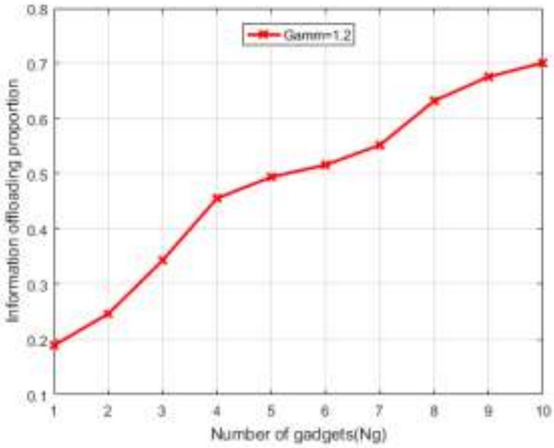


Fig (2): Number of gadgets Vs Information offloading proportion with . $N_g = 10, N_{doc} = 20, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

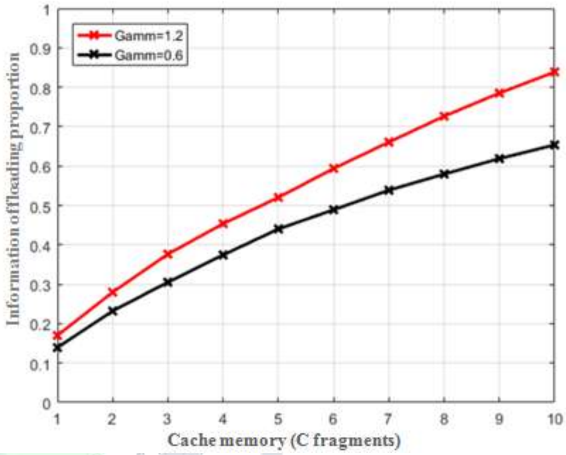


Fig (3): Cache memory (C) Vs Information offloading proportion with. $N_g = 10, N_{doc} = 20, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

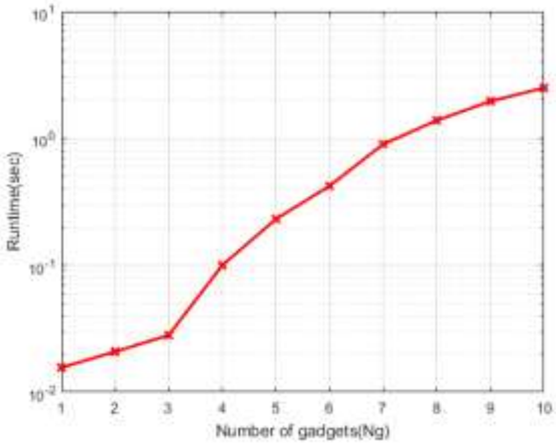


Fig (4): Number of gadgets (N_g) Vs Runtime(sec) with $N_g = 10, N_{doc} = 20, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

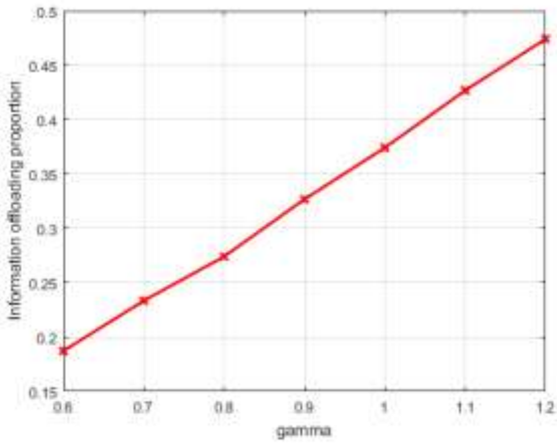


Fig (5): Gamma Vs Information offloading proportion with $N_g = 10, N_{doc} = 500, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

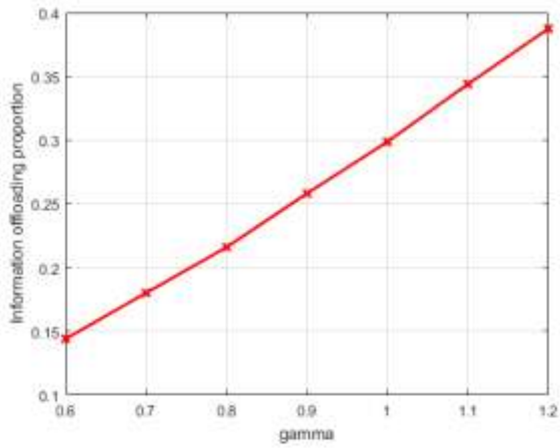
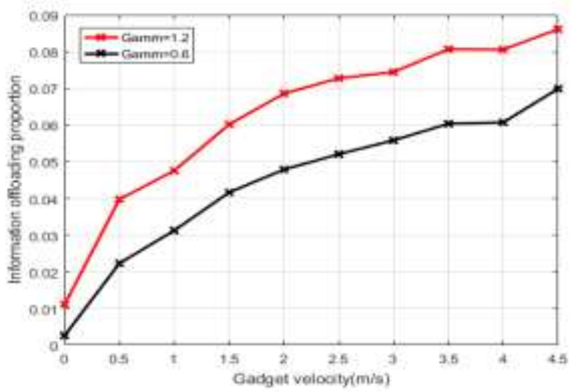


Fig (6): Gamma Vs Information offloading proportion with $N_g = 10, N_{doc} = 500, T^d = 600\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

VI. CONCLUSION

In this paper, we implement the gadget to gadget network and use the method which takes motility of clients into consideration so that we analyze the performance of the method by considering various factors like number of gadgets in the system, cache memory size, gamma value, gadget velocity, we get optimum performance when those factors are higher. In future work we can increase the factors mentioned above and we can apply it for the autonomous vehicle communication, also we can increase the range of the gadget for better communication.



.

Fig (7): Gadget velocity Vs Information offloading proportion with. $N_g = 10, N_{doc} = 20, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

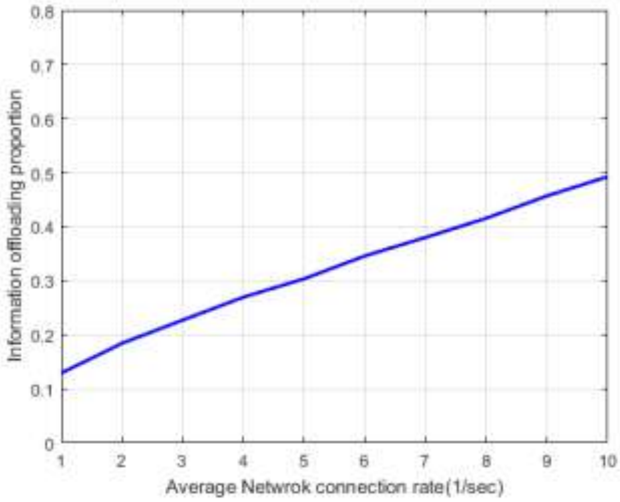


Fig (8): Average network connection rate Vs Information offloading proportion with $N_g = 10, N_{doc} = 500, T^d = 120\ s, A_{u,v} = 2, I_{max} = 5\ and\ C = 10.$

REFERENCES

[1] V. Chandrasekhar, J. Andrews, and A. Gatherer, “Femtocell networks: A survey,” IEEE Commun. Mag., vol. 46, no. 9, pp. 59–67, Sep. 2008.

[2] 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 Commun. Mag., vol. 51, no. 4, pp. 142–149, Apr. 2013.

[3] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A.F. Molisch, and G. Caire, “Femtocaching: Wireless content delivery through distributed caching helpers,” IEEE Trans. Inf. Theory, vol. 59, no. 12, pp. 8402–8413, Sep. 2013.

[4] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, “Femtocaching: Wireless video content delivery through distributed caching helpers,” in Proc. of IEEE Int. Conf. on Computer
Commun. (INFOCOM), Orlando, FL, Mar. 2012.

[5] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey, “An analysis of approximations for maximizing submodular set functions-I,” Math. Prog., vol. 14, no. 1, pp. 265–294, Dec. 1978.

[6] A. Schrijver, Combinatorial optimization: polyhedra and efficiency. Springer Science & Business Media, 2003.

[7] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák, “Maximizing a monotone submodular function subject to a matroid constraint,” SIAM
J. Comput., vol. 40, no. 6, pp. 1740–1766, 2011.

[8] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, “Scaling behavior for device-to-device communications with
distributed caching,” IEEE Trans. Inf. Theory, vol. 60, no. 7, pp. 4286–4298, Apr. 2014.

[9] B. Blaszczyszyn and A. Giovanidis, “Optimal geographic caching in cellular networks,” in Proc. IEEE Int. Conf. on Commun. (ICC), London, UK, Jun. 2015.

[10] Rui Wang, Jnu Zhang, S.H.Song, Khaled B. Letaief, “Mobility-Aware Caching in D2D Networks” IEEE Transactions on Wireless Communications, Volume: 16, Issue: 8, Aug. 2017.