International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.5, Sep-Oct. 2012 pp-3841-3849 ISSN: 2249-6645

Data Caching Between Mobile Nodes in Wireless Adhoc Networks

Y.Tulasi Rami Reddy 1, K.Gopinath 2

[1.] *M.Tech Student In CSE Department, Kottam College Of Engineering, Kurnool (India) 518218,*
[2.] *Associate Professor in IT Department, KSRM College Engineering, Kadapa (India) 516003*

Abstract: *we are introduced the cooperative caching in wireless networks ,where the nodes may be mobile and exchange information in apeer-to-peer fashion. We consider both cases of nodes with large-and small-sized caches. For large-sized caches, we devise a strategy where nodes, independent of each other, decide whether to cache some content and for how long. In the case of small-sized caches, we aim to design a content replacement strategy that allows nodes to successfully store newly received information while maintaining the good performance of the content distribution system. Under both conditions, each node takes decisions according to its per-ception of what nearby users may store in their caches and with the aim of differentiating its own cache content from the other nodes'. The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the de-sired information nearby. We simulate our caching algorithms indifferent ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient*
Information access.

Index Terms *– Data cashing, mobile Nodes, mobile ad hoc networks.*

I. Introduction

The necessary information to users on the move is one of the most promising directions of the infotainment business, which rapidly becomes a market reality, because infotainment modules are deployed on cars and handheld devices. The ubiq-uity and ease of access of third- and fourth-generation (3Gor 4G) networks will encourage users to constantly look for content that matches their interests. However, by exclusivelyrelying on downloading from the infrastructure, novel appli-cations such as mobile multimedia are likely to overload thewireless network (as recently happened to AT&T followingthe introduction of the iPhone [1]). It is thus conceivable that a peer-to-peer system could come in handy, if used in conjunction with cellular networks, to promote content sharing using ad hoc networking among mobile users [2]. For highlypopularcontent, peer-to-peer distribution can, indeed, remove bottlenecks by pushing the distribution from the core to the edge of the network In such an environment, however, a cache-all-you-see ap-proach is unfeasible, because it would swamp node storage capacity with needless data that were picked up on the go. Thus, several techniques of efficiently caching information in wireless ad hoc networks have been investigated in the literature; for example, see the surveys in [3] and [4] and the related work discussed in Section II. The solution that we propose, called Hamlet, aims at creating content diversity within the node neighborhood so that users likely find a copy of the different information items nearby (regardless of the content popularity level) and avoid flooding the network with query messages. Although a similar concept has been put forward in [5]–[8], the novelty in our proposal resides in the probabilistic estimate, run by each node, of the information presence (i.e., of the cached content) in the node

proximity. The estimate is performed in a cross-layer fashion by overhearing content query and information reply messages due to the broadcast nature of the wireless channel. By lever- aging such a local estimate, nodes autonomously decide what information to keep and for how long, resulting in a distributed scheme that does not require additional control messages. The Hamlet approach applies to the following cases.

1.1 Large-sized caches. In this case, nodes can potentially store a large portion (i.e., up to 50%) of the available information items. Reduced memory usage is a desirable (if not required) condition, because the same memory may be shared by different services and applications that run at nodes.

1.2 Small-sized caches. In this case, nodes have a dedicated but limited amount of memory where to store a small percentage (i.e., up to 10%) of the data that they retrieve. The caching decision translates into a cache replacement strategy that selects the information items to be dropped among the information items just received and

the infor- mation items that already fill up the dedicated memory.

We evaluate the performance of Hamlet in different mobile network scenarios, where nodes communicate through ad hoc connectivity. The results show that our solution ensures a high query resolution ratio while maintaining the traffic load very low, even for scarcely popular content, and consistently along different network connectivity and mobility scenarios.

II. RELATED WORK

Several papers have addressed content caching and content replacement in wireless networks. In the following sections, we review the works that are most related to this paper, highlighting the differences with respect to the Hamlet framework that we propose.

2.1. Cooperative Caching

In [9], distributed caching strategies for ad hoc networks are presented according to which nodes may cache highly popular content that passes by or record the data path and use it to redirect future requests. Among the schemes presented in [9], the approach called HybridCache best matches the operation it as a benchmark for Hamlet in our comparative evaluation. In [10], a cooperative caching technique is presented and shown to provide better performance than HybridCache. However, the solution that was proposed is based on the formation of an over-lay network composed of "mediator" nodes, and it is only fitted to static connected networks with stable links among nodes.These assumptions, along with the significant communication overhead needed to elect "mediator" nodes, make this scheme unsuitable for the mobile environments that we address. The work in [11] proposes a complete framework for information retrieval and caching in mobile ad hoc networks, and it is built on an underlying routing protocol and requires the manual setting of a network wide "cooperation zone" parameter. Note that assuming the presence of a routing protocol can prevent the adoption of the scheme in [11] in highly mobile networks, where maintaining network connectivity is either impossible or more communication expensive than the querying/caching process. Furthermore, the need of a manual calibration of the "cooperation zone" makes the scheme hard to configure, because different environments are considered. Conversely, Hamlet is self contained and is designed to self adapt to network environments with different mobility and connectivity features.

2.2. Content Diversity

Similar to Hamlet, in [6], mobile nodes cache data items other than their neighbors to improve data accessibility. In particular, the solution in [6] aims at caching copies of the same content farther than a given number of hops. Such a scheme, however, requires the maintenance of a consistent state among nodes and is unsuitable for mobile network topologies. The concept of caching different content within a neighborhood is also exploited in [7], where nodes with similar interests and mobility patterns are grouped together to improve the cache hit rate, and in [8], where neighboring mobile nodes implement a cooperative cache replacement strategy. In both works, the caching management is based on instantaneous feedback from the neighboring nodes, which requires additional messages. The estimation of the content presence that we propose, instead, avoids such communication overhead.

2.3. Caching With Limited Storage Capability

In the presence of small-sized caches, a cache replacement technique needs to be implemented. Aside from the scheme in [8], centralized and distributed solutions to the cache place- ment problem, which aim at minimizing data access costs when network nodes have limited storage capacity, are presented in [14]. Although centralized solutions are not feasible in ad hoc environments, the distributed scheme in [14] makes use of cache tables, which, in mobile networks, need to be maintained similar to routing tables. A content replacement scheme that addresses storage limita-tions is also proposed in [16]. It employs a variant of the last recently used (LRU) technique, which favors the storage of the most popular items instead of the uniform content distribution targeted by Hamlet. In addition, it exploits the cached item IDs provided by the middleware to decide on whether to reply to passing-by queries at the network layer, as well as link-layer traffic monitoring to trigger prefetching and caching. In [17], the popularity of content is taken into account, along with its update rate, so that items that are more frequently updated are more likely to be discarded. Similarly, in [18], cache replace-ment is driven by several factors, including access probability, update frequency, and retrieval delay.

2.4. Data Replication

Although addressing a different problem, some approaches to data replication are relevant to the data caching solution that we propose. One technique of eliminating information replicas among neighboring nodes

is introduced in [21], which, unlike Hamlet, requires knowledge of the information access

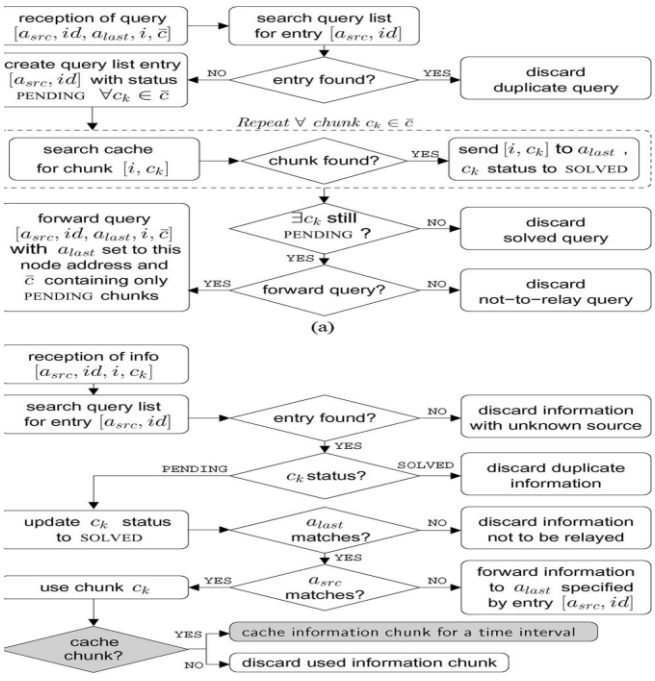
frequency and periodic transmission of control messages to coordinate the nodes' caching decisions. In [5], the authors propose a replication scheme that aims at having every node close to a copy of the information and analyze its convergence time.

III. System Outline And Assumptions

Hamlet is a fully distributed caching strategy for wireless ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. In particular, we address a mobile ad hoc network whose nodes may be resource-constrained devices, pedestrian users, or vehicles on city roads. Each node runs an application to request and, possibly, cache desired information items. Nodes in the network retrieve information items from other users that temporarily cache (part of) the requested items or from one or more gateway nodes, which can store content or quickly fetch it from the Internet. We assume a content distribution system where the following assumptions hold: 1) A number I of information items is available to the users, with each item divided into a number C of chunks; 2) user nodes can overhear queries for content and relative responses within their radio proximity by exploiting the broadcast nature of the wireless medium; and 3) user nodes can estimate their distance in hops from the query source and the responding node due to a hop-count field in the messages.

1

Fig: 1. Flow Charts of the processing of a) query and b) information messages at user nodes



Although Hamlet can work with any system that satisfies theafore mentioned three generic assumptions, for concreteness ,we detail the features of the specific content retrieval system that we will consider in the remainder of this paper. The reference system that we assume allows user applica-tions to request an information item i ($1 \le i \le I$) that is notin their cache. Upon a request generation, the node broadcasts a query message for the C chunks of the information item. Queries for still missing chunks are periodically issued until either the information item is fully retrieved or a timeout expires. If a node receives a fresh query that contains a request for information i's chunks and it caches a copy of one or more of the requested chunks, it sends them back to the requesting node through information messages. If the node does not cache (all of) the requested chunks, it can rebroadcast a query for the missing chunks, thus acting as a forwarder. Once created, an information message is sent back to the query source. To avoid a proliferation of information copies along the path, the only node that is entitled to cache a new copy of the information is the node that issued the query. Information messages are transmitted back to the source of the request in a unicast fashion, along the same path from which the request came.

A node that receives the requested information has the option to cache the received content and thus become a provider for that content to the other nodes.

IV. Simulation Scenarios And Metrics

We tested the performance of Hamlet through ns2 simulations under the following three different wireless scenarios:

1) a network of vehicles that travel in a city section (referred to as City); 2) a network of portable devices carried by customers who walk in a mall (Mall); and 3) a network of densely and randomly deployed nodes with memory limitations (memory constrained nodes). The three scenarios are characterized by different levels of node mobility and network connectivity.In the City scenario, as depicted in Fig. 4, vehicle movement is modeled by the intelligent driver model with intersection management (IDM-IM), which takes into account car-to-car interactions and stop signs or traffic lights [27]. We simulated a rather sparse traffic, with an average vehicle density of 15 veh/km over a neighborhood of 6.25 km2. The mobility model settings, forcing vehicles to stop and queue at intersections, led to an average vehicle speed of about 7 m/s (i.e.,25 km/h). We set the radio range to 100 m in the vehicular scenario, and by analyzing the network topology during the simulations, we observed an average link duration of 24.7 s

and a mean of 45 disconnected node clusters concurrently present over the road topology. The City scenario is thus characterized by scattered connectivity and high node mobility. The Mall scenario is represented in Fig. 4 as a large L-shaped open space of 400 m of length on the long side, where pedestrian users can freely walk. In this scenario, we record an average of 128 users who walk at an average speed of 0.5 m/s according to the random-direction mobility model with reflections [28]. The node radio range is set to 25 m, leading to an average link duration equal to 43 s, with a mean of ten disconnected clusters of users present at the same time in the network. The connectivity level in the Mall is thus significantly higher than in the City, whereas node mobility is much lower.The memory-constrained scenario is similar to the scenario employed for the performance evaluation of the cache The information-sharing application lies on top of a User Datagram Protocol (UDP)-like transport protocol, whereas,

at the media access control (MAC) layer, the IEEE 802.11 standard in the promiscuous mode is employed. No routing algorithm is implemented: queries use a MAC-layer broadcast transmission, and information messages find their way back to the requesting node following a unicast path. Information messages exploit the request to send/clear to send (RTS/CTS) mechanism and MAC-level retransmissions, whereas query messages of broadcast nature do not use RTS/CTS and are never retransmitted. The channel operates at 11 Mb/s, and signal propagation is reproduced by a two-ray ground model. Simulations were run for 10 000 s.

In the aforementioned scenarios, our performance evaluation hinges upon the following quite-comprehensive set of metrics that are aimed at highlighting the benefits of using Hamlet in a distributed scenario:

1) The ratio between solved and generated queries, called solved-queries ratio;
2) The communication overhead;
3) The time needed to solve a query;
4) The cache occupancy.

V. Evaluation With Large-Sized Caches

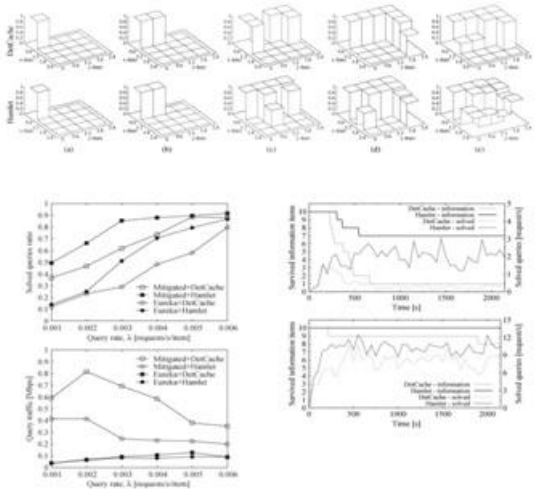
Here, we evaluate the performance of Hamlet in a network of nodes with large storage capabilities, i.e., with caches that can store up to 50% of all information items. Because such characteristics are most likely found in vehicular communication devices, tablets, or smartphones, the network environments under study are the City and Mall scenarios. As discussed in Section IV, in this case, the Hamlet framework is employed to compute the caching time for information chunks retrieved by nodes, with the goal of improving the content distribution in the network while keeping the resource consumption low. We first compare Hamlet's performance to the results obtained with a deterministic caching strategy, called DetCache, which simply drops cached chunks after a fixed amount of time.

Then, we demonstrate the effectiveness of Hamlet in the specific task of information survival. In all tests, we assume I = 10 items, each comprising C = 30 chunks. All items have identical popularity, i.e., all items are requested with the same rate $\lambda = \Lambda/I$ by all network nodes. The choice of equal request rates derives from the observation that, in the presence of nodes with a large-sized memory, caching an information item does not imply discarding another information item; thus, the caching dynamics of the different items are independent of each other and only depend on the absolute value of the query rate. It follows that considering a larger set of items would not change the results but only lead to more time-consuming simulations.

Each query includes 20 B plus 1 B for each chunk request, whereas information messages include a 20-B header and carry a 1024-B information chunk. The maximum caching time MC is set to 100 s, unless otherwise specified. Queries for chunks that are still missing are periodically issued every 5 s until either the information is fully retrieved or a timeout that is set to 25 s expires.

5.1 Benchmarking Hamlet

We set the deterministic caching time in DetCache to 40 s, and we couple DetCache and Hamlet with both the mitigated flooding and Eureka techniques for query propagation. We are interested in the following two fundamental metrics: 1) the ratio of queries that were successfully solved by the system and 2) the amount of query traffic that was generated. The latter metric, in particular, provides an indication of the system effectiveness in preserving locally rich information content: if queries hit upon the sought information in one or two hops, then the query traffic is obviously low. However, whether such a wealth of information is the result of a resource-inefficient cache-all-you-see strategy or a sensible cooperative strategy, e.g., the approach fostered by Hamlet, remains to be seen. Thus, additional metrics that are related to cache occupancy Fig: 5.1 Mall: Solved –queries ratio (top) and query traffic (bottom) with different schemes versus content request rate.



5.2 Information Survival

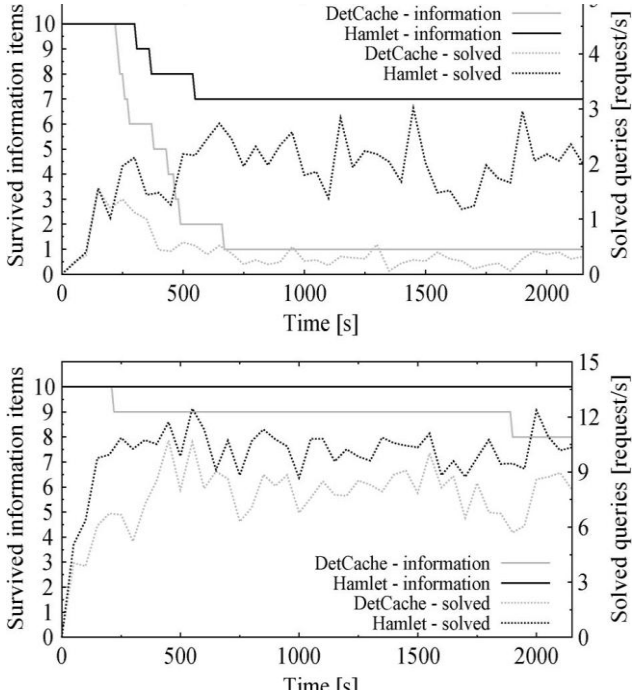


Fig5.3 Mall: Information survival for different gateway switch-off times. The smaller numbers on the x- axis indicate the landmark time (in seconds) to which the number of survived items refers.(computed from the start of the simulation). Clearly, the laterthe gateways are shut down, the higher the probability of information survival, because the information has more time to spread through the network. We observe that Hamlet can maintain information presence equal to 100% if the information is given enough time to spread, i.e., gateways are disabled after 600 s or more, whereas DetCache loses half the items within the first 2000 s of simulation. We could wonder whether caching times give the edge to either Hamlet or DetCache. However, the average caching time in Hamlet ranges from 37 s to 45 s, depending on the gateway switch-off times and on the specific information item considered. These values are very close to the DetCache caching time of 40 s, showing that Hamlet improves information survival by better distributing content in the network and not by simply caching them for longer periods of time.

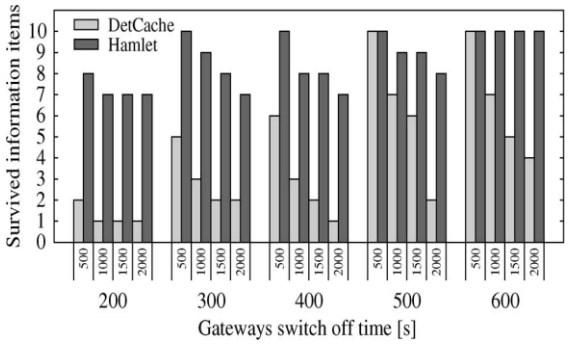


Fig 5.2 Information survival in the Mall (top) and

VI. Evaluation With Small-Sized Caches

We now evaluate the performance of Hamlet in a network where a node cache can accommodate only a small portion of the data that can be retrieved in the network. As an example, consider a network of low-cost robots that are equipped with sensor devices, where maps that represent the spatial and temporal behavior of different phenomena may be needed by the nodes and have to be cached in the network. We thus consider the memory-constrained scenario introduced in Section V and employ the Hamlet framework to define a cache replacement strategy. In such a scenario, the caching dynamics of the different information items become strongly intertwined. Indeed, caching an item often implies discarding different previously stored content, and as a consequence, the availability of one item in the proximity of a node may imply the absence of another item in the same area. Thus, in our evaluation, it is important to consider a large number of items, as well as to differentiate among these items in terms of popularity. We consider an overall pernode query rate $\Lambda = 0.1$ and sets of several hundreds of items. We assume that popularity levels qi are distributed according to the Zipf law, which has been shown to fit popularity curves of content in different kinds of networks [29]. When not stated otherwise, the Zipf distribution exponent is set to 0.5. Such a value was selected, because it is close to the values observed in the real world [29], and the skewness that it introduces in the City (bottom) scenarios. The temporal behavior of the survived information and solved queries when the gateway nodes are switched off at t = 200 s.

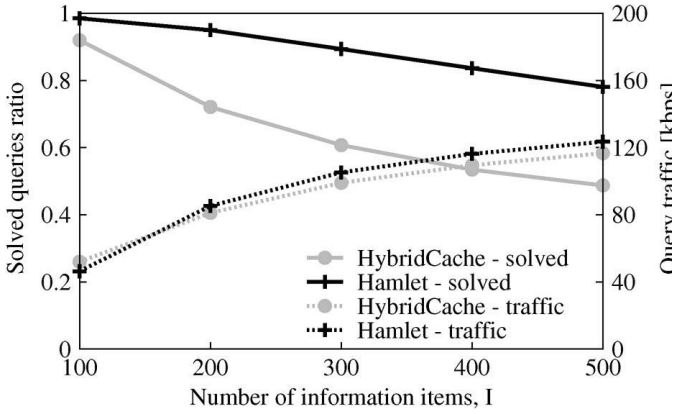


Fig 6.1 Static memory-constrained nodes: Solved-queries ratio and query traffic as the information set size varies, with Hybrid Cache and Hamlet.

popularity distribution is already sufficient to make differences emerge between the caching schemes that we study. In any case, we provide an analysis of the impact of the Zipf exponent at the end of this section. We assume that nodes can cache at most ten items, which correspond to a percentage between 2% and 10% of the entire information set, depending on the considered value of I. In addition, we set C = 1 to account for the smaller size of information items typically exchanged by memory-constrained nodes and MC to 300 s, because the increased network connectivity prolongs the reliability of information presence estimation.

6.1. Benchmarking Hamlet

Let us first focus on the memory-constrained scenariooutlined in Section V with static nodes. Fig. 6.1 solved queries ratio and the overall query traffic versus the information set size. We observe that Hamlet reacts better to the growth of the number of items than HybridCache, without incurring any

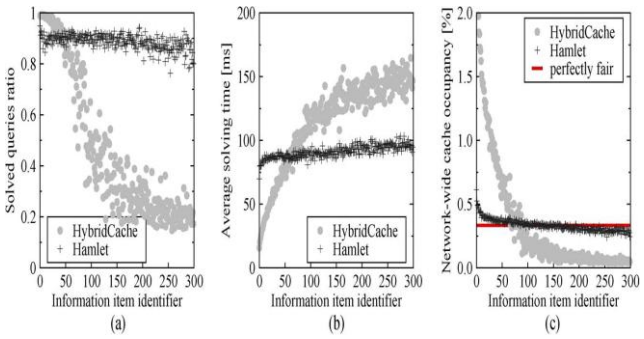


Fig. 6.2. Static memory-constrained nodes.
(a) Query-solving ratio, (b) time, and (c) average networkwide cache occupancy for each item when usingHybridCache and Hamlet, with I = 300. In (c), the red horizontal line represents perfect fairness in cache occupancy among different items.

penalty in terms of network load, as shown by the similar query traffic generated by the two schemes.

Observing the performance of Hamlet and HybridCache on a per-item basis allows a deeper understanding of the results.

In Fig. 6.2 (a), we show the solving ratio of the queries for each item when I = 300. Along the x-axis, items are ordered in decreasing order of popularity, with item 1 representing the most sought-after information and item 300 the least requested information. Unlike Hamlet, HybridCache yields extremely skewed query solving ratios for the different content; a similar observation also applies to the time needed to solve queries,

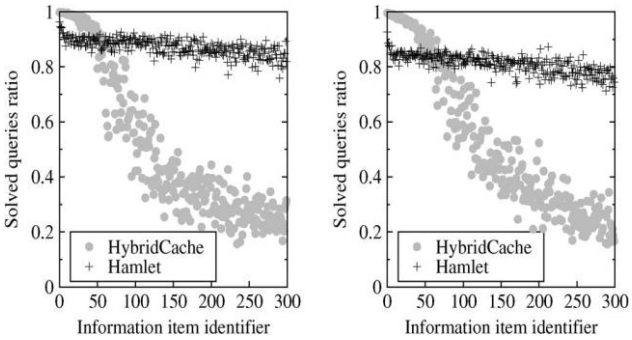


Fig. 6.3. Memory-constrained mobile nodes: Query-solving ratio for eachinformation item when using HybridCache and Hamlet, with I = 300. Theplots refer to vm that is equal to 1 m/s (left) and 15 m/s (right).

We now compare the performance of HybridCache and Hamlet in the scenario with memory-constrained mobile nodes. We test the two schemes when I = 300 and for an average node speed vm equal to 1 and 15 m/s.

The solved-queries ratio recorded with HybridCache and Hamlet on a per-item basis are shown in Fig. 13. Comparing the left and right plots, we note that the node mobility, even at high speed, does not seem to significantly affect the results due to the high network connectivity level. The spatial redistribution of content induced by node movements negatively affects the accuracy of Hamlet's estimation process, which explains the slight reduction in the solved query ratio at 15 m/s. That same movement favors HybridCache, at least at low speed, because it allows unpopular information to reach areas that are far from the gateway. However, the difference between the two schemes is evident, with Hamlet solving an average of 20% requests more than HybridCache, when nodes move at 15 m/s. even if it represents two thirds of the whole information set.

Instead, Hamlet achieves, in a completely distributed manner, a balanced networkwide utilization of node caches. Indeed, the results of Hamlet are very close to

the most even cache occupancy that we can have, represented by the horizontal red line in the plot and corresponding to the case where the total network storage capacity is equally shared among the I items

VII. Results

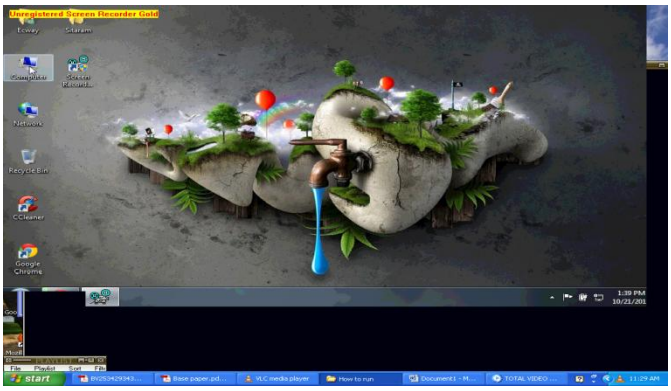


Fig 1. Out Put Screen for Data Caching in Mobile nodes

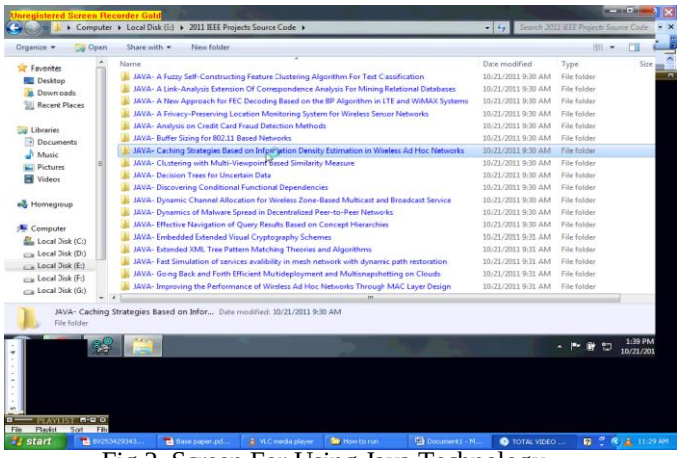


Fig 2. Screen For Using Java Technology

Fig 3. Mobile nodes in wireless adhoc-networks

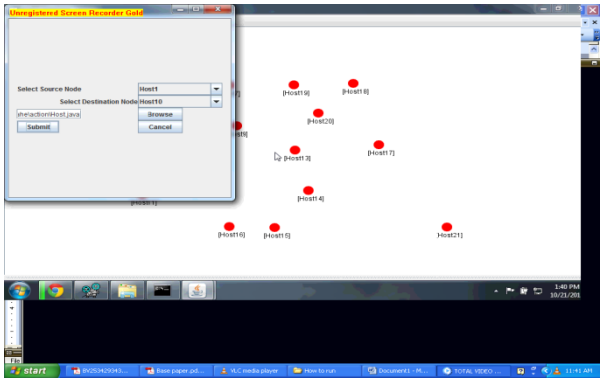


Fig 4. Screen from Source Host to Destination Host

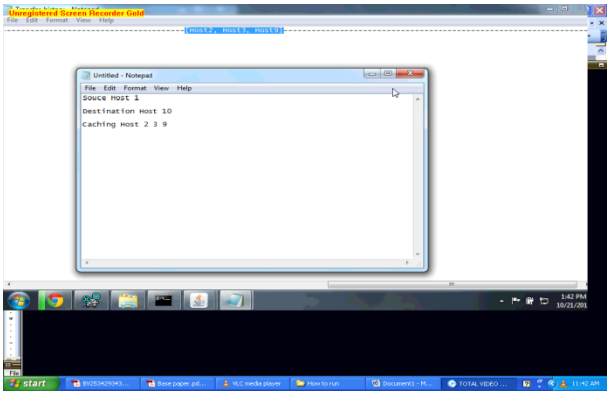


Fig 5. Screen for Data Caching between Hosts 2, 3, 9

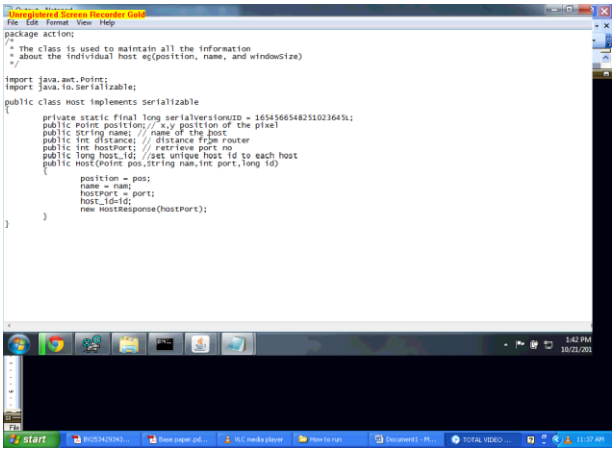


Fig 6. Using Java package function for each caching node possition

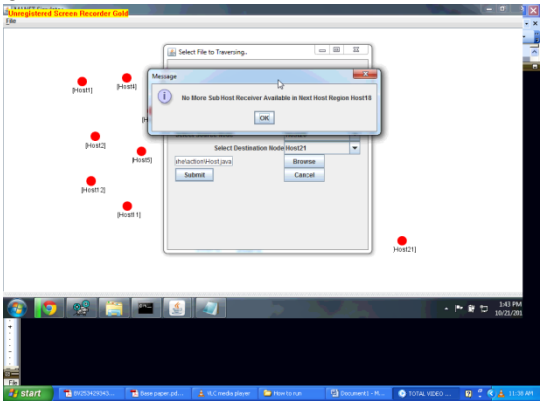


Fig 7. Screen for No More Sub Hosts for caching the data

VIII. Conclusion

We have introduced Hamlet, which is a caching strategy for adhoc networks whose nodes exchange information items in a peer-to-peer fashion. Hamlet is a fully distributed scheme where each node, upon receiving a requested information, determines the cache drop time of the information or which con-tent to replace to make room for the newly arrived information. These decisions are made depending on the perceived "presence" of the content in the node's proximity, whose estimation does not cause any additional overhead to the information sharing system. Data caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. Data caching is a fully distributed scheme where each node, upon receiving requested information, determines the cache drop time of the information or which content to replace for the newly arrived information. We have developed a paradigm of data caching techniques to support effective data access in ad hoc networks. In particular, we have considered memory capacity constraint of the network nodes. We have developed efficient data caching algorithms to determine near optimal cache placements to maximize reduction in overall access cost. Reduction in access cost leads to communication cost savings and hence, better bandwidth usage and energy savings.

However, our simulations over a wide range of network and application parameters show that the performance of the caching algorithms. Presents a distributed implementation based on an approximation algorithm for the problem of cache placement of multiple data items under memory constraint.The result is the creation of content diversity within the nodes neighborhood so that a requesting user likely finds the desired information nearby. We simulate our caching algorithms in different ad hoc network scenarios and compare them with other caching schemes, showing that our solution succeeds in creating the desired content diversity, thus leading to a resource-efficient information access. We showed that, due to Hamlet's caching of information that is not held by nearby nodes, the solving prob-ability of information queries is increased, the overhead trafficis reduced with respect to benchmark caching strategies, and this result is consistent in vehicular, pedestrian, and memory-constrained scenarios. Conceivably, this paper can be extended in the future by addressing content replication and consistency.

The procedure for information presence estimation that was developed in Hamlet can be used to select which content should be replicated and at which node (even if such a node did not request the content in the first place). In addition, Hamlet can be coupled with solutions that can maintain consistency among copies of the same information item cached at different network.

References

[1] J. Wortham (2009, Sep.). Customers Angered as iPhones Overload AT&T. The New York Times. [Online]. Available: http://www.nytimes. com/2009/09/03/technology/companies/03att.html

[2] A. Lindgren and P. Hui, "The quest for a killer app for opportunistic and delay-tolerant networks," in Proc. ACM CHANTS, 2009, pp. 59–66.

[3] P. Padmanabhan, L. Gruenwald, A. Vallur, and M. Atiquzzaman, "A sur- vey of data replication techniques for mobile ad hoc network databases,"VLDB J., vol. 17, no. 5, pp. 1143–1164, Aug. 2008.

[4] A. Derhab and N. Badache, "Data replication protocols for mobile ad hoc networks: A survey and taxonomy," IEEE Commun. Surveys Tuts., vol. 11 no. 2, pp. 33–51, Second Quarter, 2009.

[5] B.-J. Ko and D. Rubenstein, "Distributed self-stabilizing placement of replicated resources in emerging networks," IEEE/ACM Trans. Netw., vol. 13, no. 3, pp. 476–487, Jun. 2005.

[6] G. Cao, L. Yin, and C. R. Das, "Cooperative cache-based access in ad hoc networks," Computer, vol. 37, no. 2, pp. 32–39, Feb. 2004.

[7] C.-Y. Chow, H. V. Leong, and A. T. S. Chan, "GroCoca: Group-based peer-to-peer cooperative caching in mobile environment," IEEE J. Sel. Areas Commun., vol. 25, no. 1, pp. 179–191, Jan. 2007.

[8] T. Hara, "Cooperative caching by mobile clients in push-based informa- tion systems," in Proc. CIKM, 2002, pp. 186–193.

[9] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," IEEE Trans. Mobile Comput., vol. 5, no. 1, pp. 77–89, Jan. 2006.

[10] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Cooperative caching in wireless multimedia sensor networks," ACM Mobile Netw. Appl., vol. 13, no. 3/4, pp. 337–356, Aug. 2008.

[11] Y. Du, S. K. S. Gupta, and G. Varsamopoulos, "Improving on-demand data access efficiency in MANETs with cooperative caching," Ad Hoc Netw.,vol. 7, no. 3, pp. 579–598, May 2009.

[12] Y. Zhang, J. Zhao, and G. Cao, "Roadcast: A popularity-aware content sharing scheme in

VANETs," in Proc. IEEE Int. Conf. Distrib. Comput. Syst., Los Alamitos, CA, 2009, pp. 223–230.

[13] W. Li, E. Chan, and D. Chen, "Energy-efficient cache replacement policies for cooperative caching in mobile ad hoc network," in Proc. IEEE WCNC, Kowloon, Hong Kong, Mar. 2007, pp. 3347–3352.

AUTHORS LIST

Tulasi Rami Reddy Yeddula (M.Tech) Received his Master's Degree in CSE in MKU University, Madurai and Pursuing Masters of Technology in Computer Science & Engineering in Kottam College of Engineering, Kurnool, affiliated to JNTU Anantapur. His research areas of interest are Computer Networks and Data Mining.

K.GOPINATH M.Tech (Ph.D), Completed his B.Tech ,Computer Science in (JNTU, Anantapur) 1999 and M.tech, Computer Science (JNTU, Hyd) 2002 .Presently Pursuing Ph.D in Data Mining JNTUniversity, Hyderabad . His area of intrest are Computer Networks and Data Mining.