International Journal of Computational Intelligence and Informatics, Vol. 2: No. 3, October - December 2012

Efficient Cache Replacement Techniques Based on Information Density over Wireless Adhoc Networks

S Vimala
Department of Computer Science and Engineering
Selvam College of Technology
Namakkal, India
jk.vims@gmail.com

R Anbarasu
Department of Computer Science and Engineering
Selvam College of Technology
Namakkal, India
anbarasume@gmail.com

***Abstract*- 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 to make room for the newly arrived information. These decisions are made depending on the perceived “presence” of the content in the nodes proximity, whose estimation does not cause any additional overhead to the information sharing system. 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 perception 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 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.**

Keywords- Data caching, mobile ad hoc networks

I. INTRODUCTION

Ad hoc networks are multi hop wireless networks of small computing devices with wireless interfaces. The computing devices could be conventional computers (for example, PDA, laptop, or PC) or backbone routing platforms or even embedded processors such as sensor nodes. The problem of optimal placement of caches to reduce overall cost of accessing data is motivated by the following two defining characteristics of ad hoc networks. First, the ad hoc networks are multi hop networks without a central base station. Thus, remote access of information typically occurs via multi hop routing, which can greatly benefit from caching to reduce access latency. Second, the network is generally resource constrained in terms of channel bandwidth or battery power in the nodes. Caching helps in reducing communication, this results in savings in bandwidth, as well as battery energy. The problem of cache placement is particularly challenging when each network node has a limited memory to cache data items. In this paper, our focus is on developing efficient caching techniques in ad hoc networks with memory limitations.

Research into data storage, access, and dissemination techniques in ad hoc networks is not new. In particular, these mechanisms have been investigated in connection with sensor networking peer-to-peer networks mesh networks world wide Web and even more general ad hoc networks Distributed implementations.

Our goal in this paper is to develop an approach that is both analytically tractable with a provable performance bound in a centralized setting and is also amenable to a natural distributed implementation. In our network model, there are multiple data items; each data item has a server, and a set of clients that wish to access the data item at a given frequency. Each node carefully chooses data items to cache in its limited.

II. RELATED WORKS

Several papers have been addressed the 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 propose.

2.1. Cooperative Caching

A cooperative caching technique is presented and show into provide better performance than Hybrid Cache. However, the solution that was proposed is based on the formation of an overlay 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 [1] proposes a complete framework for information retrieval and caching in mobile ad hoc networks, and it is built on an underlying routing protocol.

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 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 [2], where neighboring mobile nodes implement a cooperative cache replacement strategy.

2.3 Hamlet Framework

The Hamlet framework allows wireless users to take *caching decisions* on content that they have retrieved from the network. In particular, for each information item, a node records the distance (in hops) of the node that issues the query, i.e., where a copy of the content is likely to be stored, and the distance of the node that provides the information.

Based on such observations, the node computes an index of the *information presence* in its proximity for each of the *I* items, then, as the node retrieves content that it requested A node that receives the requested information has the option become a provider for that content to the other nodes. Determining a strategy of taking such caching decisions is the main objective of this paper, and as such, the corresponding decision blocks are highlighted and point out that Hamlet exploits the observation of query and information messages that are sent on the wireless channel as part of the operations of the content-sharing application.

The reference system that we assume allows user applications to request an information item I ($1 \leq i \leq I$) that is not in their cache. Upon a request generation, the node broadcasts *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. The exact algorithm that is followed by a node upon the reception of a query message is detailed in the flowchart in Fig. 1.

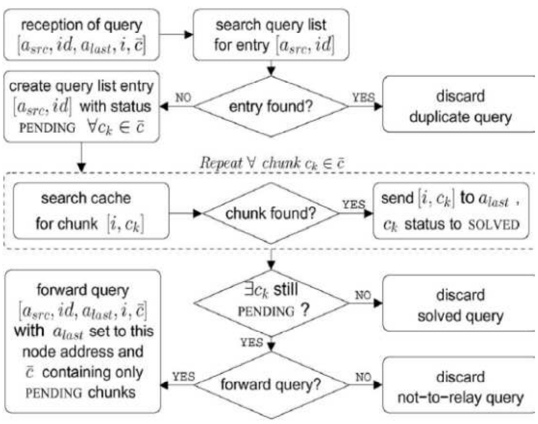

Fig. 1. Flowcharts of the processing of query

Once created, an information message is sent back to the query source. Information messages are transmitted back to the source of the request in a Unicast fashion, along the same path from which the request came. To this end, backtracking information is carried and updated in query messages.

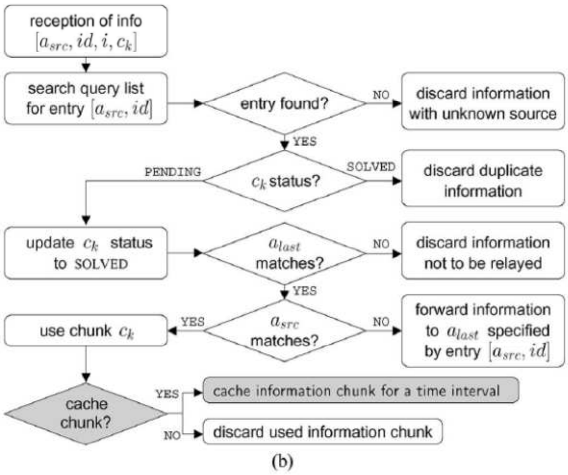

Fig.2.Flowcharts of the processing information messages at user nodes

Fig. 2 depicts the flowchart of the operations at a node that receives a message that contains an information chunk. 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. Determining a strategy of taking such caching decisions is the main objective of this paper, and as such, the corresponding decision blocks are highlighted in Fig.2 As a consequence, Hamlet does not introduce any signaling overhead.

Furthermore, several optimizations can be introduced to improve the aforementioned basic scheme for the discovery of content although our focus is not on query propagation, it is important to take the query process into account, because it is directly determines the network load associated with the content retrieval operation. While deriving the results, we consider the following two approaches to query propagation.

a) Mitigated flooding. This approach limits the propagation range of a request by forcing a time to live (TTL) for the query messages. In addition, it avoids the forwarding of already-solved requests by making the nodes wait for a query lag time before rebroadcasting a query.
b) Eureka [2]. This approach extends mitigated flooding by steering queries toward areas of the network where the required information is estimated to be denser.

Note that this paper focuses on cooperative caching and we do not tackle information consistency; thus, we do not take into account different versions of the content in the system model. We note, however, that the previous version of this paper [4] jointly evaluated Hamlet with a basic scheme for weak cache consistency based on an epidemic diffusion of an updated cache content and we showed that weak consistency can be reached, even with such a simple approach, with latencies on the order of minutes for large networks. Node R in the lower plot is a relay node, overhearing the exchanged messages. The upper and lower plots respectively represent the case 1 hQ value for the provider node P and the case 2 hQ and hP values for the relay node R with respect to the query source Q and the provider P.

III. ZONE COOPERATIVE CACHING SCHEME

This section describes our zone cooperative (ZC) caching scheme for data retrieval in mobile ad hoc networks. The design rationale of ZC caching is that it is considered advantageous for a client to share cache with its neighbor in the zone (i.e., mobile hosts that are accessible in one hop).Mobile hosts belonging to the zone of a given host then form a cooperative cache system for this host since the cost for communicating with them is low both in terms of energy consumption and message exchange. Fig.4 shows the behavior of ZC caching strategy for a client request.

3.1. ZC cache management

Cache management is more complex in cooperative caching because deciding what to cache can also depend on the client's neighbors. In this section, we present cache management including cache replacement policy, cache admission control and cache consistency.

3.2 Cache Consistency

Cache consistency issue must be addressed to ensure that clients only access valid states of the data. Two widely used cache consistency models are the weak consistency and the strong consistency model.

In the weak consistency model, a stale data might be returned to the client. In the strong consistency model, after an update completes, no stale copy of the modified data will be returned to the client.

Recently, we have done some work [5][6] on maintaining strong cache consistency in the one-hop based mobile environment. However, due to bandwidth and power constraints in ad hoc networks, it is too expensive to maintain strong consistency, and the weak consistency model is more attractive [6][7][. The ZC caching uses a simple weak consistency model based on the time-to-live (TTL), in which a client considers a cached copy up-to-date if its TTL has not expired. The client removes the cached data when the TTL expires. A client refreshes a cached data item and its TTL if a fresh copy of the same data passes by.

IV. SIMULATION RESULTS

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 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.

Here couple both schemes with mitigated flooding. While deriving the results, we noted that caching the data paths leads to poor performance due to the high cache replacement frequency in the simulated scenarios in Fig.5.

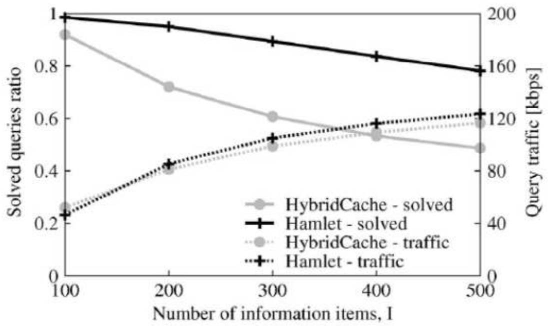

Fig.3. Static memory-constrained nodes: Solved-queries ratio and query traffic as the information set size varies, with Hybrid Cache and Hamlet.

Therefore, we set the Hybrid Cache parameters so that the following two conditions are satisfied: 1) The size of the data never results in data path caching but always in information caching, and 2) mitigated flooding is always employed for query forwarding. In addition, to reduce the numbers of query transmissions in the network, queries for missing chunks are not reissued, and both Hamlet and Hybrid Cache are coupled with the preferred group broadcasting (PGB) technique [3].

V. CONCLUSION

We have introduced Hamlet, which is a caching strategy for ad hoc networks whose nodes exchange information items in a peer-to-peer fashion. Hamlet 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 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.

This paper showed that, due to Hamlet’s caching of information that is not held by nearby nodes, the solving probability of information queries is increased, the overhead traffic is 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 nodes, as well as with the versions stored at gateway nodes.

REFERENCES

[1] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, “An analysis of the optimum node density for ad hoc mobile networks,” in Proc. IEEEICC,Helsinki, Finland, Jun. 2001, pp. 857–861.

[2] G. Cao, L. Yin, and C. R. Das, “Cooperative cache-based data access in adhoc networks,” Computer, Feb. 2004, vol. 37, no. 2,pp. 32–39,

[3] L. Yin and G. Cao, “Supporting cooperative caching in adhocnetworks,”IEEE Trans. Mobile Comput. , Jan. 2006, vol. 5, no. 1, pp. 77–89.

[4] M. Fiore, F. Mininni, C. Casetti, and C.-F. Chiasserini, “To cache or not to cache?” in Proc. IEEE INFOCOM, Riode Janeiro, Brazil, Apr. 2009,pp. 235–243.

[5] M. Fiore, C. Casetti, and C.-F. Chiasserini, “Information density estimation for content retrieval in MANETs,” IEEE Trans. Mobile Comput., Mar. 2009 vol. 8, no. 3, pp289–303.

[6] Marco Fiore, Claudio Casetti, Carla-Fabiana Chiasserini,” Caching Strategies Based on Information Density Estimation in Wireless Ad Hoc Networks”,IEEE transactions on vehicular technology, June 2011 vol. 60, no. 5.

[7] Narottam Chand, R.C. Joshi and Manoj Misra,” Efficient Cooperative Cache in Ad Hoc Networks”, in Proc. IEEE ICC Roorkee, Jan 2002.

[8] B. Tang, H. Gupta, and S. Das, “Benefit-based data caching in adhoc networks,” IEEE Trans, Mobile Comput., Mar. 2008,vol. 7,no. 3, pp. 289–304