Cross-layer Networking for Peer Databases over Wireless Ad-Hoc Communities

Evangelos Papapetrou, Efthymia Rova, Apostolos Zarras and Panos Vassiliadis
Department of Computer Science
{epap,efirova,zarras,pvassil}@cs.uoi.gr
University of Ioannina, Dourouti Campus
45110, Ioannina, Greece

***Abstract*— In this paper we address the problem of efficiently distributing query messages among peers in a wireless ad hoc network. We assume that peers are organized in classes. Each peer possesses a local database and can answer queries posed by other peers. Each peer can also pose queries to all the peers belonging to a certain class and/or within a certain range of distance in the network. Contrary to traditional p2p lookup queries, we are interested in collecting answers from as many peers as possible. To efficiently serve this purpose, we take advantage of routing and application layer specifics (e.g., class information, network distance) to avoid flooding and at the same time preserve compatibility with traditional routing and transport mechanisms.**

I. Introduction

Over the last years, the scientific community has witnessed a significant rise of data-centric networking applications. This type of applications introduced a new communication paradigm. Instead of communicating one-to-one based on their identity, users generally communicate in groups depending on the data they possess and need to exchange at that time. Take for example p2p file sharing applications [1]. Users communicate spontaneously depending on their content. The communication can be considered to consist of two phases. In the first phase a node has to communicate simultaneously with multiple other nodes in order to discover a node possessing the desired data. Then in the second phase a classical communication between the two nodes takes place. As a second example, the CarTel project at MIT [2] aims at providing a data management system for querying data collected over large communities of sensor-enabled cars. The queries that can be posed concern the position, speed and other sensor data of the involved cars. Since road and highways do not offer continuous connectivity, each time a car has access to "islands of connectivity" the interesting data are relayed via the Internet. On the contrary, in traditional applications routing of data does not depend on the application protocol but it is rather determined by the addresses of the communicating nodes.

Since data-centric communication depends on the application layer, the key for providing such communication services is to incorporate application layer specifics into networking functions. However, this is not possible in existing network infrastructure since its development has neglected application layer dependencies so far. For example, the routing services required by a p2p file sharing application (for example routing of queries) cannot be performed by existing networking technologies. So far, the approach taken in order to overcome the lack of flexibility in existent infrastructures was the introduction of overlays mostly in wired [1] but also in wireless networks [3]. According to this approach, data-centric communication services are implemented on the application layer by means of the overlay network. Then, traditional networking services are used to support the construction of the overlay network. However, this approach has received considerable skepticism [4] regarding the use of network resources.

Although overlays are attractive for networks that can not be upgraded, such as the Internet, this is not the case for wireless ad-hoc networks. Moreover, in this case the problem of limited network resources is more critical, rendering the use of overlays questionable. Recently, many researchers have focused on the implementation of efficient data-centric routing protocols in wireless ad-hoc networks. Most of the reported research concerns p2p file and/or resource sharing applications. In this scenario, resources are available in the network. Users produce lookups in order to locate one node that possess the requested resource/data. Several solutions have been proposed for efficiently routing lookup messages in wireless mobile ad-hoc networks [5]. Furthermore, in the same context, scientists have proposed ancillary procedures for the dissemination and replication of resources [6], [5]. Another example of data-centric communication comes from sensor networks. Several routing protocols have been proposed to efficiently route data towards a specific network node [7]. Application layer functions such as data fusion and data aggregation are introduced at the network layer in order to optimize the overall system operation.

Although significant work has been already performed, there is still open field for research. This is because the design of data-aware routing protocols is strongly related to the characteristics of the application to which the network has to provide its services. In this paper we are concerned with the cross-layer design of a routing protocol suitable for supporting the operation of a peer database application over a wireless mobile ad-hoc network. Recently there is an increased interest for such applications especially in wireless mobile environments [1]. Although the communication scenario within such an application shares similarities with other data-centric applications, at the same time significant differences call for the design of customized networking procedures. Those differences will be analyzed in Section II where the problem is described in detail and the proposed solution is delineated. The remainder of the paper is organized as follows. Section III presents the description of the simulation model used for the assessment of the proposed protocol. Then, in section IV simulation results are presented and discussed, while conclusion remarks are drawn in section V.

II. Proposed Cross-Layer Design

A. Problem Description

As mentioned in the previous paragraph, our motivation stems from distributed databases and the communication services required to support their operation over a wireless mobile ad-hoc network. The basic scenario that we consider in our work is as follows. Mobile users, which roam in an area, carry information that is organized in a local database. Users are organized in groups, which are called *classes*. All peers of the same class support the same *interface* (i.e., provide the same type of data or services) to the peer network. Therefore, once a peer's class is known, its is straightforward for other peers to pose queries to it. In order to understand the notion of class, we provide an example. Suppose that a wireless network is set up to a mall. Users on cars, pedestrians or even users on buildings constitute different classes and their databases are organized in a manner relevant to their identity. For example, it is reasonable to assume that all cars carry information about their brand while that information does not exist in the databases of pedestrian users. An example network is depicted in fig. 1. In a network like the one described,

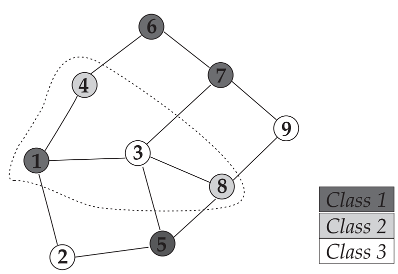


Fig. 1: An example network with three classes of nodes

mobile users are allowed to perform queries toward other users in order to retrieve information. Depending on the type of the desired information, a query may be relevant to a specific class of users, to more than one classes, or it can even be global, that is to address all network nodes independent of their class. Furthermore, mobile users have to be capable of limiting the range of their query. After receiving a query, a user constructs a reply message based on its local database and routes this message back to the originator of the query.

In the described scenario, the network has to provide users with specific services:

- **Routing of queries**. This is the most important service and its main objective is to deliver a query to the appropriate recipients.
- **Delivery of reply messages**. Aims at delivering the reply messages produced by a query to the originator of that query.

As far as routing of queries is concerned it must be stated that although it bears similarities to the routing of lookups in p2p systems, there are also important differences that affect the network operation. The most important is that, instead of locating a user that possess the requested data, in our case the network has to deliver the query to *as many as possible* nodes within a specific class, so that as much information as possible is collected. A different kind of query involves querying all peers within a certain network distance (number of hops). Furthermore, techniques such as data dissemination and replication do not significantly affect the overall setting since they can take place only to a limited extent, in the sense that every peer carries its personal data that are probably highly volatile and time variant (e.g., a car's speed, a pedestrian's position, etc).

B. Technical requirements and challenges

It is well understood that traditional routing protocols for wireless ad-hoc networks cannot provide the communication services described previously in an efficient manner. The main issue consists in developing a mechanism for routing queries. A solution based on traditional routing protocols would be to unicast the query to each of the network nodes, leading to the waste of valuable bandwidth. Another more elaborate solution would be to use multicast routing protocols [8]. However, the implementation of such a scenario would require the construction of a multicast tree based on the classes of nodes that the query refers to. This is translated to different multicast trees for different queries. Bearing in mind the overhead involved in building and maintaining multicast trees, the inefficiency of the approach is clear. Finally, the most appealing approach is to use a flood-based technique in order to reach all possible query recipients. However, in this case, the incurred overhead is also considered significant. This is because all network nodes are involved in a transmission at least once, even if they are not appropriate recipients of the query. Consider the case in fig. 1 where node 1 wishes to perform a query for discovering data related to class 2. All nodes in the network will receive at least one copy of the query. Our objective is to minimize the involved overhead by limiting the transmission of the query. To this end, we make the observation that we must capitalize on the information provided by the application layer such as the organization of nodes into classes. In other words, we must opt for a cross-layer design of the routing protocol. In our previous example, by exploiting class information we can limit the propagation of the query within the region indicated by the dashed line, which is the minimum connected tree including node 1 and nodes 4 and 8 that belong to class 2. In this way we economize on the consumption on bandwidth and at the same time we are able to retrieve information from all target nodes.

As far as the delivery of reply messages is concerned it is clear that traditional one-to-one communication can be used. Therefore, the proposed scheme should be compatible with existent routing protocols. Another reason in favor of this design choice is that we do not aim at building a special purpose network. On the contrary, we wish to support the coexistence of data-centric applications such as a peer database with other types of applications.

To summarize, we state that the basic objectives of our new method are:

- the minimization of the cost involved in routing a query by means of interaction with the application layer
- the compatibility and coexistence with traditional routing protocols

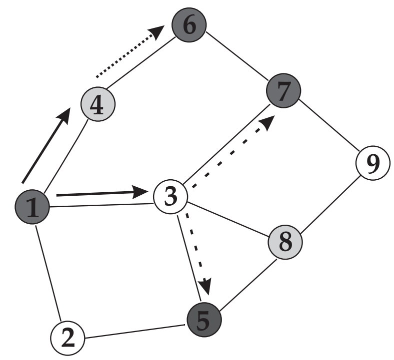
C. The proposed solution

As mentioned in the previous subsection one of our key objectives is the compatibility with existing routing protocols. Therefore, we are not interested in building a new routing protocol rather than propose generally applicable modifications so that the described functionality is incorporated into existing routing protocols. We must make clear at this point that the proposed solution aims at proactive routing protocols due to the nature of the application. Keep in mind that when a node performs a query, data must be collected by as many as possible network nodes. The only way to achieve this without performing a network-wide search is to use a proactive routing protocol. Our proposed method consists of two steps:

1) Introduction of Application Layer Specifics: In order to be able to make decisions based on application information at the networking layer, we must devise an efficient mechanism for making available this information (the class that a node belongs to) to the routing protocol so that we can make routing decisions. To this end, we introduce the id of the node's class to the routing updates sent by this node. Furthermore, each node receiving such updates from other nodes should store class information to the entry of its routing table that is related to the node that sent the update. In this way, each node becomes aware of the class that every other node in the network belongs to. The proposed procedure incurs some overhead due to the propagation of extra information in the network. However, this information is rather small (only a couple of bytes). Furthermore, bear in mind that the information regarding the class of a node is in most cases fixed. Therefore its propagation in the network by means of update messages is limited. Finally, we should point out that the proposed procedure does not depend on the routing protocol that is implemented.

2) Query Forwarding: The cornerstone for implementing an efficient query routing mechanism is to make forwarding decisions for query messages based on the application layer information, already incorporated into the routing tables. In our approach, when a node produces a query it forwards it to its network layer. To avoid flooding, the routing protocol uses its routing table in order to determine the nodes that belong to the class/classes that the query refers to. To understand the procedure consider the example depicted in fig. 2 where node 1 issues a query towards nodes in class 1. By consulting its routing table ($RT{:}Node1$), node 1 can determine the *target nodes*, i.e., the nodes to which the query must reach. In our example, the target nodes are nodes 5,6 and 7 (highlighted in fig. 2). After determining the target nodes, the originator of the query constructs a header of the structure depicted in fig. 3. To construct this header, the node, based on its routing table, groups the target nodes according to the next hop used to reach them. The next hop nodes, that are used to reach a target node, are called *forwarding nodes* and are stored in the header fields $FW_1, FW_2, \ldots$. Each forwarding node is followed by a set of target nodes ($T_1, \ldots, T_k$) which are the target nodes that can be reached through this forwarding node. The forwarding node may be a target node as well. However, as it will be made clear later, its address does not need to be included in this set. For each forwarding node, the list of target nodes is also accompanied by its size (TL_size). In our example, nodes 5 and 7 are grouped under node 3 and node 6 under node 4. Nodes 3 and 4 are the forwarding nodes. The header also contains the number of forwarding nodes in field FL_size and the address of the node originating the query (QN_Addr).

After its construction the header is appended to the query message and the new packet is broadcasted in the network. Each node that receives the packet forwards it to its application layer. Furthermore, it checks if its address appears in the list of forwarding nodes. If this is not the case, the node drops the packet. However, if the node is a forwarding node then it constructs a new header by following the procedure described



RT:Node 1

Dst	NH	Hop Cnt	Class
1	-	0	1
2	2	1	3
3	3	1	3
4	4	1	2
5	3	2	1
6	4	2	1
7	3	2	1
8	3	2	2
9	3	3	3

RT:Node 3

Dst	NH	Hop Cnt	Class
1	1	1	1
2	1	2	3
3	-	0	3
4	1	2	2
5	5	1	1
6	7	2	1
7	7	1	1
8	8	1	2
9	8	2	3

Fig. 2: Example of query forwarding

QN_addr	FL_size	FW_1	TL_1_size	T_1	....	$T_{k(1)}$	....	FW_M	TL_M_size	T_1	....	$T_{k(M)}$

Fig. 3: Structure of query header

before and using as target nodes only the nodes included in its target list. To make this clear, consider again the example of fig. 2. In this example, node 3 is a forwarding node. When receiving the query from node 1, it constructs a new header following the same procedure as node 1 and using as targets only the nodes 5 and 7 (node 6 is responsibility of node 4). The described procedure ends when a node that receives the query discovers that its list of target nodes is empty. At this point, all target nodes have received the query. Whenever a target node receives a query, computes the answer and forwards it back to the originator of the query through traditional routing mechanisms.

The result of the proposed procedure is to minimize the number of transmissions within the network since the retransmission of query packets is strictly controlled. Consider again the example of fig. 2. The total number of transmissions used to deliver the query to all recipients is three (each transmission is depicted with a different line style). In the case that a flood-based method is used, the total number of transmissions for the same example is at least nine. The disadvantage of the method is that it relies on the accuracy of the routing protocol used to deliver queries. However, this is a tradeoff that we are willing to pay in order to save resources and to achieve the compatibility with traditional routing protocols.

III. Simulation Framework

The experiments conducted in this work aim at evaluating the performance of the proposed cross-layer method for routing queries. For this purpose, we compare the proposed approach to the traditional approach which is to deliver queries by means of flooding. To the best of the authors' knowledge there is no other method suitable for the specific problem. The simulations were conducted using the MANET simulation environment existing within Ns2 [9] after performing appropriate modifications. In both cases the routing protocol that is used as the base protocol was the well-known DSDV [10]. It is clear from the previous discussion that the choice of the protocol affects the two methodologies in the same way as far as data delivery is concerned. Furthermore, as far as routing of queries is concerned, the routing protocol can only affect the performance of the proposed cross-layer method in terms of accuracy in delivering queries. However, the main purpose of this work is to prove the efficiency of the proposed method in terms of bandwidth consumption.

A. Simulation Model

As a simulation model we used a network consisting of a number of nodes (N) that roam in a rectangle area. For the simulation of node movement we used the well-known Random Waypoint Algorithm. Nodes are randomly organized into C_{cnt} classes of equal size. A number $Q(Q < N)$ of nodes performs queries using an exponential distribution with mean rate λ. Queries are propagated to a maximum number of hops (H_{max}). Each node chooses uniformly the class to which it addresses its queries. Finally, each node receiving a query replies with a packet of 1000 bytes in size. The values used for the simulation parameters are summarized in table I. Unless stated, these values are valid throughout all experiments. Finally, it should be noted that the results that will be presented are obtained as average values over 15 independent simulation runs. That number of runs provided 99% confidence intervals of $\pm 4\%$ in the worst case.

TABLE I: Simulation parameters

Number of nodes (N)	50
Region Size	1500x300
Node mobility	20 m/sec
Node Range (R)	250 m
Number of Classes (C_{cnt})	5
Number of querying nodes (Q)	10
Query generation rate (λ)	0.1 query/sec
Maximum number of hops (H_{max})	3
Simulation Time	900 sec

B. Evaluation metrics

For the evaluation of the proposed method we use three different metrics: a) *query overhead*, i.e., the mean number of query packets involved in the routing of a single query, b) *mean number of replies*, i.e., the mean number of data packets received as replies to a single query and c) *delivery ratio*, i.e., the percentage of packets containing data (replies) that was delivered to the node performing the query. The first metric evaluates the efficient use of system resources and provides an indication of the method scalability. The second metric is an indication of the method's ability to collect the appropriate information from peer nodes. Finally, the third metric is used to show the impact of resource consumption to the overall system operation.

IV. Performance Results

In order to evaluate the proposed technique we conducted three different experiments. Each one aims at exploring the impact of different system parameters to the system performance.

A. Variable User Mobility

In this first experiment we vary the speed at which nodes are moving. The purpose of this experiment is to evaluate the system performance under various levels of mobility that affect the network connectivity. In fig. 4a the overhead involved in the routing of one query is depicted for both the proposed and the competitor approach. Results prove the effectiveness of the proposed method. In fact the transmissions are reduced to approximately 20% of those taking place when flooding is used. The induced overhead is practically the same over the entire mobility range. Although this is expected for flooding, for the cross-layer approach this is a very useful feature because it shows that query forwarding is not affected by link failures. The reduced overhead in the case of the cross-layer approach has an effect on the overall system performance as illustrated in fig. 4c. Because the network is more congested when the flood-based approach is used, the delay in data delivery is higher. This has an increased effect in the range of

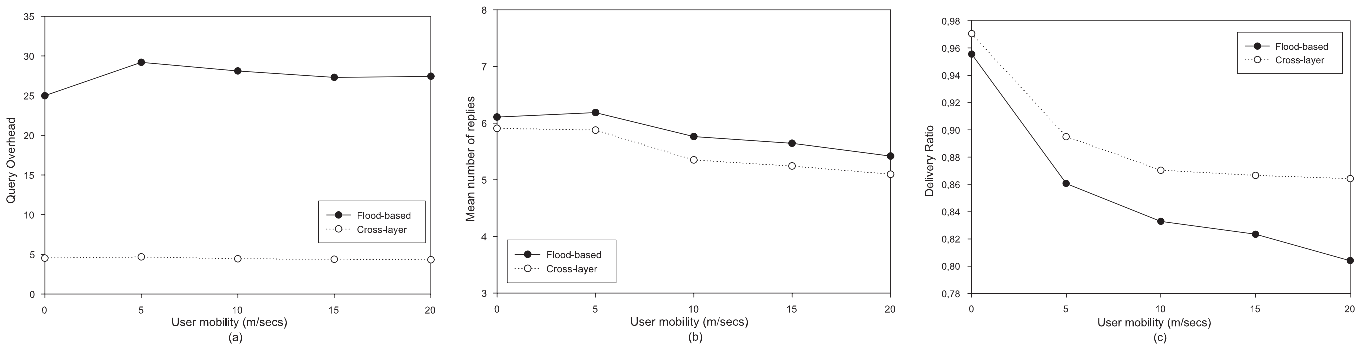


Fig. 4: Variable user mobility

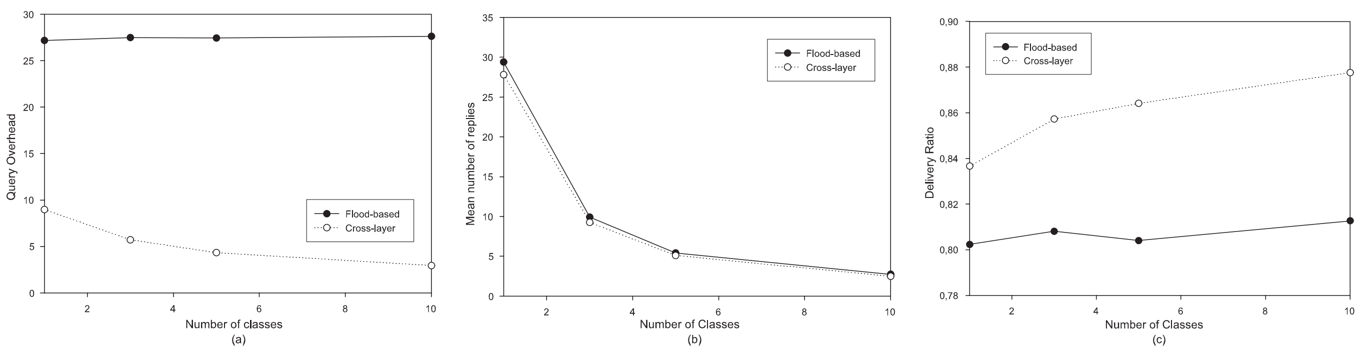


Fig. 5: Variable number of classes

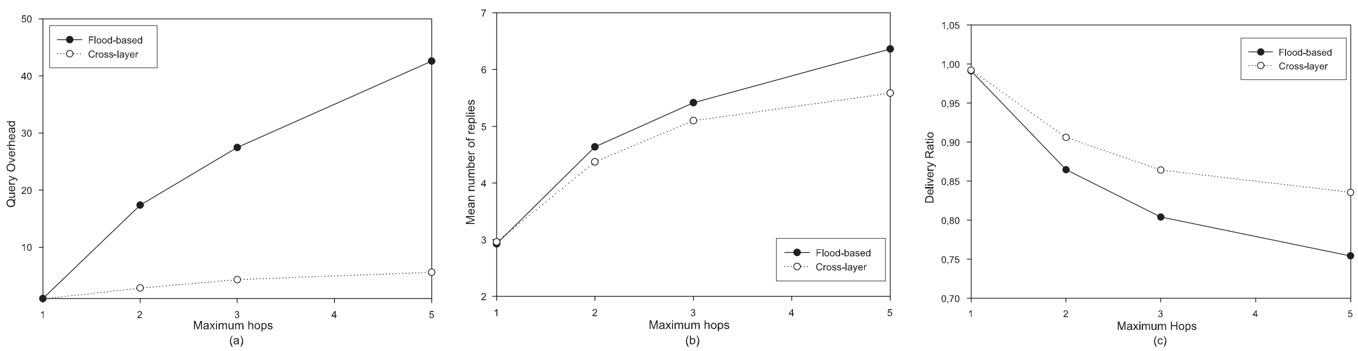


Fig. 6: Queries limited in terms of maximum number of hops

high mobility where link failures become more frequent. On the contrary, the cross-layer technique manages a smoother network operation. Finally, in fig. 4b the mean number of replies per query is presented. As expected, both methodologies are affected when mobility raises due to the increased level of link failures. An interesting result is that the proposed method manages to deliver queries to less nodes as a result of the increased failure rate of the routing protocol. This is the price paid for minimizing the overall overhead. However, it is interesting that the differences between the cross-layer and the flood-based approach are small.

B. The Impact of Classes

In this experiment we vary the number of the classes into which network nodes are organized. The purpose of this experiment is to directly capture the impact of using application layer information. Fig. 5a presents the query overhead with respect to the number of classes. Again the cross-layer approach achieves significantly less overhead. As actually expected, the performance of the flood-based method does not change as the number of classes increases, since it always asks all the peers of the network. Contrary to the constant performance of the flood-based method, the cross-layer approach improves its performance when the number of classes increases. This is due to the fact, that the proposed method takes advantage of class information and tries to reach only the peers of the requested class/classes. As the number of classes increases, the number of such peers drops. It is worth noticing that even when only one class exists, i.e., the query concerns every node in the network, the proposed method manages a reduction of 67%. At the same time, the ability of the proposed method to deliver queries and therefore collect data is practically the same as in the flood-based case. This is illustrated in fig. 5b where the mean number of replies is presented. It is clear that the reduction of replies with the increase of classes is attributed to the smaller number of nodes belonging to each class (the total number of nodes is fixed to 50). Fig. 5c presents the delivery ratio for data in the system. When the number of classes increases the delivery ration increases for both approaches as a result of the reduced overhead which is confirmed by fig. 5a. Again, the proposed method manages significantly improved performance with respect to the flood-based approach. Observe that the benefits gained by the increase in the number of classes increase in the proposed approach, whereas the flood-based practically remains stable.

C. Performing Limited Queries

In this last experiment, queries are limited within a certain network distance (number of hops). The objective of this experiment is to emphasize on the efficient query forwarding of the cross-layer method and provide indications regarding its ability to scale. In fig. 6a the query overhead is depicted with respect to the number of maximum hops that a query packet is allowed to travel. The performance improvement of the proposed method in terms of query overhead becomes greater as the number of maximum hops increases. The interesting observation is that the proposed method presents an almost stable performance as opposed to its competitor that presents a practically linear behavior. The latter effect is typically anticipated, since, in the flood-based method, the larger the part of the network explored, the larger the number of messages exchanged. On the contrary, our proposed method limits the number of exchanged messages, since it restricts the number of contacted peers by exploiting the application-level, class information. This is a clear indication that the proposed method presents good scalability. As far as the mean number of replies is concerned (fig. 6b), the cross-layer method presents a negative discrepancy from the flood-based method which increases slowly with the maximum number of hops. This is anticipated because in high mobility environments, nodes that are located far from the querying node are harder to reach due to frequent link failures. The increase in both measures is explained due to the fact that we explore a larger part of the network. Finally, in fig. 6c we depict the delivery ratio of the two methods. The difference in their performance is a result of the increased overhead induced in the case of the flood-based method which creates increased levels of network congestion. Furthermore, the decreasing tendency in both methods is due to the fact that data replies are originated from nodes that are far in the network and therefore, the possibility of dropping packets increases as the length of the resulting paths increases too.

V. Conclusions

In this paper we have proposed a new routing and transport mechanism for wireless ad-hoc networks that incorporates application layer specifics in order to support both traditional and data-centric communications and at the same time optimizes the management of scarce network resources. The proposed method has been evaluated through extended simulations for its query overhead, mean number of replies and delivery ratio and outperforms its competitor flood-based method. Specifically, our method produces significantly lower query overhead than its competitor, achieving 4 to 5 times better performance in several cases. This is very important for the ability of the proposed method to scale and for the overall operation of the network. The delivery ratio is improved in our method, and supports the latter argument too. In terms of the mean number of replies, the effectiveness of the proposed method is quite close to the one of the exhaustive, flood-based method, without significant losses.

VI. Acknowledgement

This research was co-funded by the European Union in the framework of the program "Pythagoras I" of the "Operational Program for Education and Initial Vocational Training" of the 3rd Community Support Framework of the Hellenic Ministry of Education, funded by 25% from national sources and by 75% from the European Social Fund (ESF).

References

[1] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies." *ACM Comput. Surv.*, vol. 36, no. 4, pp. 335–371, 2004.

[2] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden, "Data management in the cartel mobile sensor computing system." in *SIGMOD Conference*, 2006, pp. 730–732.

[3] H. Pucha, S. Das, and Y. Hu, "Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks," in *In Proc. of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2004).*, 2004.

[4] M. Waldvogel and R. Rinaldi, "Efficient topology-aware overlay network," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 101–106, 2003.

[5] D. Chakraborty, A. Joshi, and Y. Yesha, "Integrating service discovery with routing and session management for ad-hoc networks." *Ad Hoc Networks*, vol. 4, no. 2, pp. 204–224, 2006.

[6] A. Datta, S. Quarteroni, and K. Aberer, "Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks." in *ICSNW*, 2004, pp. 126–143.

[7] K. Akkaya and M. F. Younis, "A survey on routing protocols for wireless sensor networks." *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[8] J. Jetcheva and D. B. Johnson, "Adaptive demand-driven multicast routing in multi-hop wireless ad hoc networks." in *MobiHoc*, 2001, pp. 33–44.

[9] K. Fall and K. Varadhan, "The ns manual," VINT Project, Univ. California, Berkeley, CA, 2001. [Online]. Available: http://www.isi.edu/nsnam/ns/ns-documentation.html

[10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers." in *SIGCOMM*, 1994, pp. 234–244.