

Dominant Pruning based Query Routing in Peer Databases over MANETs

Theodosios Aggelidis and Evangelos Papapetrou
Department of Computer Science, University of Ioannina

Abstract—In this paper we address the problem of efficiently routing queries among peers in a mobile ad-hoc network. Peers possess a local database and can pose queries to other peers, forming a distributed *peer database* application. In contrast to traditional networking, routing a query involves its delivery to as many as possible peers that are relevant to the query while at the same time minimizing the number of non-relevant peers involved in the process, therefore economizing on the limited network resources. To tackle this problem, we propose two novel protocols, namely TDP-ECT and CTPD-ECT. The new protocols utilize dominant pruning to distribute queries while at the same time incorporate application layer specifics in order to minimize the cost of routing. The proposed protocols are evaluated through extensive simulations and are proved suitable for wireless mobile environments.

I. INTRODUCTION

Data-centric applications have received great attention over the last years. In such applications, instead of communicating based on their identity, users communicate in groups depending on the data they possess and need to exchange at a specific time. In this paper we are concerned with a specific type of data-centric applications, known as *peer databases*, when deployed over a mobile ad-hoc network. Applications of this type [1] are foreseen to thrive in the context of mobile computing. Peer database applications pose significant challenges to traditional networking, especially in a wireless mobile environment of limited resources, because the communicating entities are not known prior to the actual exchange of data and need to be determined at the time of communication. More specifically, the communication scenario of peer databases requires *query packets to be forwarded to all network nodes able to respond to the query* while such *nodes should be determined at the time of the query*.

Although significant work has been performed on the field of resource discovery over MANETs ([2], [3]), this cannot be applied to the case of peer databases. The reason is that in peer database applications the objective of routing is to deliver the query to as many relevant nodes as possible. On the contrary, in resource discovery, locating a single node providing the resource is usually sufficient. Furthermore, in peer databases each peer carries its personal data that are probably highly volatile and time variant. In the domain of peer databases, there have been several interesting approaches that aim towards facilitating queries that collect as many data as possible from the peer network [4],[5]. However, mobility is not applicable since routing of queries takes place on overlay networks that rely on conventional wired infrastructures. The authors in [6] propose a broadcast based algorithm for peer

databases. However, the presented mechanism is based on the periodic dissemination of predefined reports which is not suitable for our scenario where the exchanged data are highly volatile. Finally, in [7], the authors consider a query routing mechanism through the support of an underlying traditional routing protocol.

To tackle the problem of routing queries in a peer database environment, our approach is to construct a connected dominating set (CDS) on an on-demand basis. Only nodes belonging to the connected dominating set may forward a query. To this end, we utilize total dominant pruning [8]. However, in order to minimize the cost of routing a query, the connected dominating set is constructed so that it connects only the nodes that can respond to the query. This requires an application-aware design of the network layer that allows application layer specifics to be incorporated into the routing protocol. Then, we use the application layer information to: i) modify the termination criterion used by the traditional dominant pruning algorithm, and ii) develop an application-aware method for constructing the set of nodes that are required to forward the query. Based on these modifications, two algorithms are proposed, namely TDP-ECT and CTPD-ECT. While the first utilizes the modified termination criterion, the second combines the use of both the aforementioned modifications.

The rest of the paper is organized as follows. In Section II, we provide a description of the problem and then we discuss some of the challenges regarding the design of an appropriate routing protocol. Then, in Section III, the two proposed algorithms are delineated, highlighting their advantages. In Section IV, we describe the simulation model, present the simulation results and discuss the performance of the proposed algorithms. Finally, in Section V useful conclusions are drawn.

II. BASIC CHALLENGES AND CONCEPTS

A. Problem Description

As mentioned in the previous paragraph, our motivation stems from a peer database application deployed over MANETs. In this work, we consider the following scenario; Mobile users carry a lightweight local database and can pose queries to other peers. Furthermore, peers are organized in groups, which are called *classes*. Peers of the same class provide the same type of data or services to the peer network. In a real life scenario, a class of nodes may consist of taxi drivers with wireless devices that provide information about traffic jams and available taxis or may consist of wireless-enabled busses providing schedule information. Mobile users

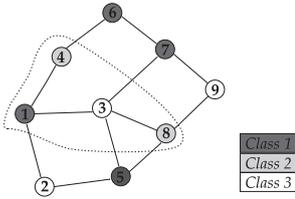


Fig. 1: An example network consisting of three classes.

may pose queries to one, more or every class in the network depending on the type of information they want to retrieve.

Definition 1. Target of a query: A node or a class of nodes that should receive the query.

Clearly, the network has to route the query to the target nodes while at the same time minimize the number of nodes involved in the process.

B. Challenges and Design Choices

To understand the challenges in routing a query, let us consider the example network depicted in fig. 1 where there exist three classes of nodes. Suppose that node 1 issues a query towards nodes of class 2. In order to minimize the cost of routing, it is desired to minimize the number of nodes that need to forward the query. To this end, we consider only the connected subnetwork that includes all the target nodes (dashed line in fig. 1). Then, a connected dominating set (CDS) of the subnetwork is used to route the query. Minimizing the size of the CDS results in reducing the routing cost. The aforementioned approach depends on application layer specifics since the target nodes are defined by the posed query. Furthermore, the plethora of possible queries may produce various combinations of target nodes which makes it impossible to maintain a routing structure for each possible query. Therefore, an on-demand approach for routing queries is required. In order to meet these requirements, the two major design choices in this work are: i) to use the dominant pruning concept for query routing, and ii) to adopt an application-aware design. More specifically:

Dominant Pruning: Although many schemes have been proposed for constructing a CDS ([9], [8]), dominant pruning algorithms and especially Total Dominant Pruning (TDP) [8] are known to provide an *on-demand* solution of low cost [9]. However, dominant pruning algorithms are not application-aware, therefore cost minimization is limited.

Application-aware design: Our approach is to minimize the number of nodes that forward a query by exploiting application layer information. To this end, the key concept is to take advantage of the organization of nodes into classes in order to: i) *define the part of the network that the dominant pruning algorithm should be executed by introducing a termination criterion that takes into account the query*, and ii) *construct the CDS according to the ability of each node to forward messages to nodes that are targets of the query*.

C. Background and notation

In this section we briefly present the operation of TDP which we consider as the reference algorithm. The following notation is employed:

- $N(v)$ denotes the set of nodes that lie 1-hop away from node v and is called the 1-hop neighborhood of v .
- $N(N(v))$ denotes the 2-hop neighborhood of node v , i.e., the set of nodes that lie within 2 hops from node v .

In dominant pruning, each node v has knowledge of $N(N(v))$ by exchanging hello messages that contain $N(v)$. When a node v wants to forward a message,¹ it constructs a local CDS of $N(N(v))$, called the *forwarding set* ($Fw(v)$). The construction of $Fw(v)$ involves the following sets:

- $B(v)$: a subset of $N(v)$ consisting of nodes that are candidates for forwarding a message.
- $U(v)$: a subset of $N(N(v))$ consisting of all the nodes that should receive a message.

Clearly, when v is the first node to send the message, $B(v) = N(v)$ and $U(v) = N(N(v))$. If v receives the message from a node u then $B(v) = N(v) - N(u)$ and $U(v) = N(N(v)) - N(N(u))$ because the nodes in $N(N(u))$ have already or going to receive the message by u 's transmission. Upon calculation of $B(v)$ and $U(v)$, $Fw(v)$ is calculated by solving the set cover problem [10] with the Greedy Set Cover (GSC) algorithm [8]. GSC uses a simple heuristic for electing the forwarding nodes. In each repetition, GSC elects from $B(v)$ the node whose neighborhood covers² more nodes in $U(v)$ until no node remains uncovered. Node v informs the forwarding nodes by piggybacking $Fw(v)$ on the forwarded message. Each node $w \in Fw(v)$ that receives the message, repeats the process unless a *termination criterion* is met. In this work, we focus on the relayed/unrelayed criterion [8] since it provides an effective and low cost solution. This criterion specifies that a node forwards a message, at most, once.

III. THE PROPOSED ALGORITHMS

In this section we propose two novel algorithms, namely *TDP-Enhanced Class-based Termination (TDP-ECT)* and *Class TDP-Enhanced Class-based Termination (CTDP-ECT)*. Both algorithms inherit the operation of TDP. However, the first algorithm utilizes a class-based termination criterion for stopping the algorithm execution while the second one additionally modifies the selection process of GSC in order to construct a class-aware CDS. Before delineating the two algorithms, we first describe the mechanisms that are used by both algorithms to facilitate the proposed modifications.

A. Basic mechanisms

It is clear that the knowledge about class information holds a key role in the application-aware design. Therefore, class information is included in the Hello messages advertised by each node. In this way, each node becomes aware of the class of each of its 2-hop neighbors. Furthermore, both algorithms use a novel technique, called *Split Forwarding Set (SFS)*, for organizing the forwarding set based on the relevance of each forwarding node to the routed query. According to SFS, the forwarding set in a node v is split in two parts, i.e., $Fw(v) = \{F_c(v), F_{nc}(v)\}$. The first part ($F_c(v)$) includes

¹In this work the terms message and query may be used interchangeably.

²A node $w \in B(v)$ "covers" a node $z \in U(v)$ if $z \in N(w)$.

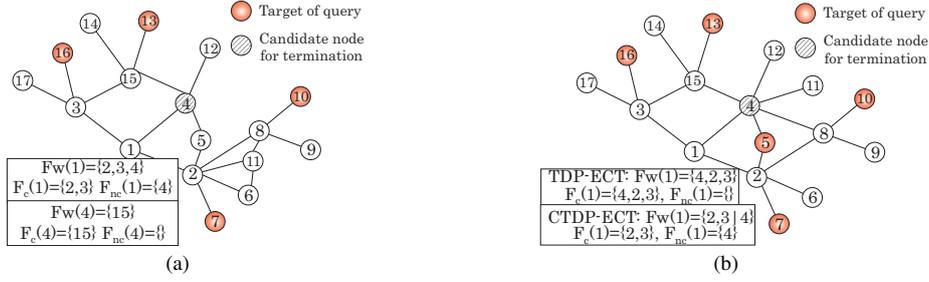


Fig. 2: The operation of (a) the TDP-ECT algorithm, (b) the CTDP-ECT algorithm

nodes that are selected by GSC to cover at least one target node. All other nodes are placed in the second part ($F_{nc}(v)$). SFS is oriented towards the elimination of non-target nodes. The intuition is that for a specific query all nodes found in $F_c(v)$ must forward the query in any case, while nodes found in $F_{nc}(v)$ are candidates for stopping the algorithm execution since they cover non-target nodes.

B. TDP-Enhanced Class-based Termination

As mentioned, TDP-ECT extends the operation of TDP by introducing a class-based termination criterion in an effort to minimize the incurred cost. The new criterion is implemented by forwarders that cover non-target nodes. Suppose that a node v has received a message from u and that $v \in F_{nc}(u)$. According to TDP-ECT, after producing its own forwarding set $Fw(v)$, node v checks whether $Fw(v) = \emptyset$. If this is the case, this means that none of v 's neighbors will forward the message. Furthermore, v covers non-target nodes since $v \in F_{nc}(u)$. Consequently, the transmission by v may be canceled without affecting the algorithm's efficiency. The requirement of $Fw(v) = \emptyset$ is imposed by the fact that nodes have limited knowledge of the network (2-hop). To understand the requirement, bear in mind that a node v , producing a non empty forwarding set, might be a bridging node that leads to an area full of target nodes, although v covers non-target nodes. In the case that v belongs to the only path leading to the aforementioned area, ceasing the transmission of the query will have a significant impact on the algorithm's performance.

To relax the strict requirement of $Fw(v) = \emptyset$ and allow for further minimization of transmissions, TDP-ECT operates as follows; In the case that $Fw(v) \neq \emptyset$, node v checks whether every node in $Fw(v)$ is an 1-hop neighbor of at least n nodes belonging to the first part of the received forwarding set ($F_c(u)$). If this condition is satisfied, node v is considered a candidate for stopping the transmission. The rationale behind this approach is that in this case all nodes in $Fw(v)$ will receive at least n copies of the query after all nodes in $F_c(u)$ forward the query. In the second step $N(v)$ is checked. If there is no target node in $N(v)$, the transmission is canceled since no node needs to receive the message. Alternatively, if target nodes exist in $N(v)$, they are required to be 1-hop neighbors of either node u or any node in $F_c(u)$, in order to ensure the reception of the query. It is imperative that the parameter n is wisely chosen since it determines the number of duplicate packets received by a node. In this work we set $n = 1$ which represents the simplest case that

allows for significant reduction in transmissions. An example of TDP-ECT operation is illustrated in fig. 2(a) where node 4 stops transmission although its forwarding set is not empty. As a result, less transmissions occur therefore overhead is minimized. However, in extreme cases of mobility there may be a slight reduction of the algorithm's efficiency due to stale neighborhood information.

C. Class based selection of forwarding nodes

CTDP-ECT inherits the functionality of TDP-ECT and extends it by introducing a class-aware version of GSC, called *Class-based GSC (C-GSC)*, for calculating the forwarding set. Traditional GSC, when executed at a node v for a message coming from u , elects the forwarding nodes based on the number of nodes that they cover in $U(v)$. On the other hand, C-GSC elects forwarding nodes based on the number of nodes that they cover in $U(v)$ and are also target nodes of the query. The pseudo-code of C-GSC is presented in fig. 3.

Fig. 2(b) provides an example of the operation of GSC (used in TDP-ECT) and C-GSC (used in CTDP-ECT). In the case of TDP-ECT, the forwarding set is populated by nodes 4, 2 and 3, based on the number of nodes that they cover. On the other hand, when CTDP-ECT is used, the first part of the forwarding set consists of nodes 2 and 3 while node 4 is in the second part. Node 2 is chosen first since two of its neighbors are targets of the query. Furthermore, node 4 is placed in the second part of the forwarding set since node 5, which is a target node and neighbor of node 4, has already been covered by node 2. In the aforementioned example, TDP-ECT will result in three transmissions, while CTDP-ECT cancels the transmission by node 4. Clearly, there are cases that the use of C-GSC may result in an increased size of the forwarding set. However, the intuition behind the use of C-GSC is the minimization of the first part of the forwarding set and the maximization of its second part which represents opportunities to cancel transmissions. In other words, C-GSC maximizes the probability of using the ECT termination criterion.

```

1: procedure C-GSC( $(B(v), U(v))$ )
2:    $Fw(v) \leftarrow \emptyset$ 
3:   while  $U(v) \neq \emptyset$  do
4:     do select  $u \in B(v)$  with max target_cnt in  $N(u) \cap U(v)$ 
5:     do resolve ties in favor of max  $|N(u) \cap U(v)|$ 
6:      $U(v) \leftarrow U(v) - N(u)$ 
7:      $Fw(v) \leftarrow Fw(v) \cup u$ 
8:   end while
9:   return  $Fw(v)$ 
10: end procedure

```

Fig. 3: The pseudocode of the C-GSC algorithm

IV. PERFORMANCE EVALUATION

A. Simulation Framework

To assess the performance of the proposed algorithms we compare them to TDP. The presented results were obtained by using ns2 [11]. The simulation model consists of N nodes roaming in a rectangle area of $1000 \times 1000 \text{ m}^2$. The well-known Random Waypoint (RW) algorithm with the perfect simulation model [12] has been used in order to avoid transient artifacts in the nodes' movement. Nodes periodically exchange hello messages and are randomly organized into C_{cnt} classes. Nodes perform queries with exponentially distributed interarrival times with mean rate λ . Each node chooses randomly the class to which it addresses its queries. The simulation parameters are summarized in Table I. Unless otherwise noted, these values are valid throughout all experiments. Finally, it must be noted that all presented results were obtained as average values over 10 independent trials.

B. Impact of the number of classes

In the first experiment we evaluate the impact of the number of classes in the algorithms' performance. This experiment can be seen as a horizontal scalability test (increase in number of classes-reduction of members per class). The results of this experiment are illustrated in fig. 4. As expected, the performance of TDP is not affected by the number of classes since it does not use the class information. Regarding the number of forwards per query, i.e., the average number of nodes that forward a query, TDP-ECT outperforms TDP as the number of classes increases and manages a reduction of up to 34%. This is reasonable since more classes result in less class members. Therefore, the opportunity of canceling a transmission increases. This is also confirmed by the performance of CTDP-ECT which also manages a reduction of 30% compared to TDP. However, CTDP-ECT outperforms TDP-ECT in terms of delivery ratio, i.e., the ratio of nodes that actually received the query to the total number of query targets, and manages to keep up with TDP. The reduction of less than 1% can be considered reasonable considering the reduced packet redundancy of CTDP-ECT. In general, as the number of classes increases, target nodes are sparse and obviously more difficult to reach since more forwards are needed on average and less paths are available. At the same time, the increased mobility (20 m/sec) creates stale neighborhood information. TDP-ECT is more vulnerable to such conditions since some nodes are scheduled to receive only one duplicate packet (since $n = 1$). As a result, TDP-ECT presents a slightly degraded performance. On the other hand, CTDP-ECT is more resilient due to its forwarder selection process that allows it to route queries towards target nodes more effectively. Finally, regarding mean delay, both TDP-ECT and CTDP-ECT outperform TDP as a result of the reduced congestion due to the limited number of duplicate packets. CTDP-ECT outperforms TDP-ECT since the operation of C-GSC maximizes the possibility that a target node will receive the query in less hops. The delay for both schemes increases with the number of classes since target nodes are becoming more sparse and path diversity

TABLE I: Default Simulation Parameters

Region Size	1000x1000 m ²	Trans. range	250 m
Num of nodes (N)	100	Max speed	20 m/secs
Sim. time	900 secs		
Num of classes	5	Num of q. nodes	$0.1N$
Query rate	0.1 q/sec	HELLO interval	2 secs

reduces. Consequently, packet losses may result in a query being delivered through a longer path. On the contrary, TDP is not affected because its message redundancy guarantees that the shortest (in hops) path is used.

C. Variable number of nodes

In the second experiment we gradually increase the size of the network. The results of this experiment are illustrated in fig. 5. Concerning the number of forwards per query, again TDP-ECT and CTDP-ECT outperform TDP for all network sizes. The improvement for TDP-ECT ranges between 23% and 28% while for CTDP-ECT is between 18% and 28.5%. As far as the delivery ratio is concerned, again CTDP-ECT performs similar to TDP while TDP-ECT presents a small hysteresis of $\sim 2\%$ for the reasons explained in the previous experiment. The reduction of delivery ratio that is observed for all algorithms in small networks can be ascribed to the reduced network connectivity. Finally, in terms of mean delay both TDP-ECT and CTDP-ECT achieve a better performance compared to TDP since they produce less congestion. The delay decreases in bigger networks for all schemes as a result of the increased connectivity that in turn increases path diversity. As a result, the probability that a query will reach its destination through a shorter path is increased.

D. Variable node mobility

The last experiment evaluates the algorithms under mobility in order to assess the impact of stale neighborhood information. Clearly, higher mobility results in smaller accuracy regarding a node's view of its neighborhood. The results of this experiment are illustrated in fig. 6. Regarding the forwards per query, the superiority of both TDP-ECT and CTDP-ECT is once again confirmed. More specifically, TDP-ECT achieves a reduction of 18% – 26%, while the reduction for CTDP-ECT ranges in 20% – 22%. Furthermore, the number of forwards per query decreases as mobility increases, for all schemes. This is because stale neighborhood information may lead to the selection of non valid forwarding nodes. As a result, queries are lost and not forwarded. This is also confirmed by the trend of the delivery ratio where the impact of mobility on all schemes is evident. TDP is more resilient to mobility due to its increased message redundancy. On the contrary, TDP-ECT experiences a delivery ratio reduced by $\sim 2.5\%$. On the other hand, CTDP-ECT manages to keep up with TDP because of its efficient forwarding process. In the highest mobility, it achieves a delivery ratio that is smaller by only 0.8% compared to TDP while at the same time significantly reduces the number of forwards per query. Finally, regarding mean delay, again CTDP-ECT outperforms all other schemes. Clearly, mobility produces increased delay since queries are delivered through longer paths. This is because paths of less hops contain hops that are longer in distance and therefore are becoming stale more frequently.

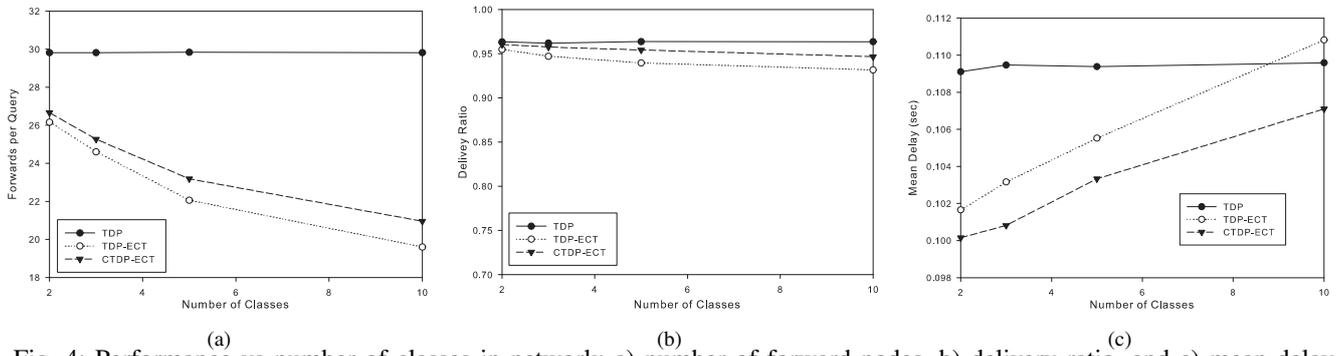


Fig. 4: Performance vs number of classes in network: a) number of forward nodes, b) delivery ratio, and c) mean delay

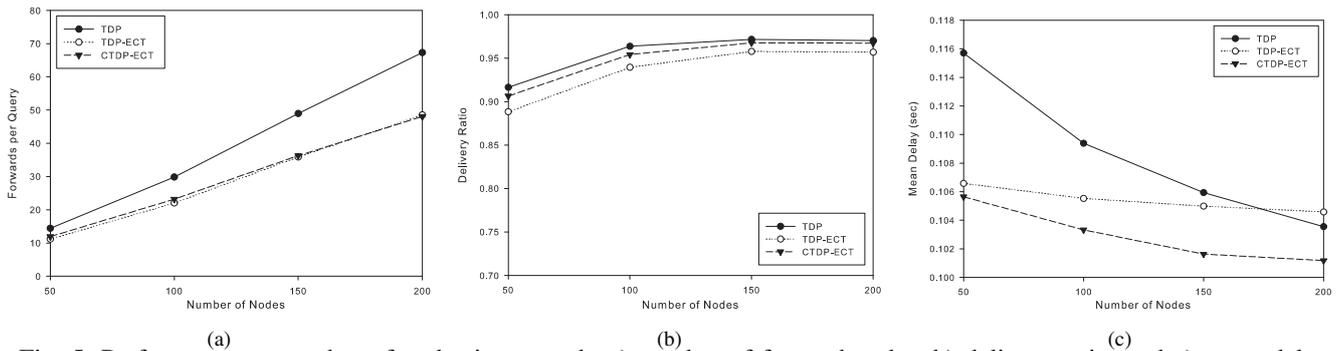


Fig. 5: Performance vs number of nodes in network: a) number of forward nodes, b) delivery ratio, and c) mean delay

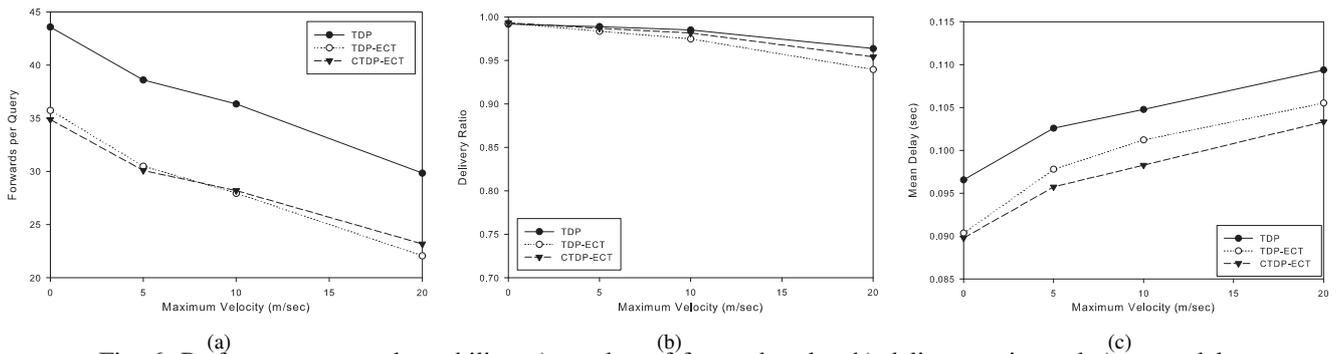


Fig. 6: Performance vs node mobility: a) number of forward nodes, b) delivery ratio, and c) mean delay

V. CONCLUSIONS

In this paper, the problem of routing a query in a peer database application, deployed over a MANET, has been addressed. More specifically, an application-aware design of dominant pruning schemes has been presented, resulting in the proposal of two novel algorithms, namely TDP-ECT and CTDP-ECT, that take advantage of application layer information and minimize the cost of routing a query. The two algorithms introduce novel criteria for terminating the execution of traditional dominant pruning algorithms as well as an innovative method for determining the forwarding set. The proposed algorithms have been evaluated through simulations that proved them suitable for providing services to a peer database application.

REFERENCES

- [1] Y. Luo and O. Wolfson, "Manet databases," in *Encyclopedia of Database Systems*, 2009, pp. 1685–1691.
- [2] 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.
- [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 IEEE WMCSA 2004*, 2004.
- [4] A. Bhambe, M. Agrawal, and S. Seshan, "Mercury: Supporting scalable multi-attribute range queries," in *ACM SIGCOMM*, 2004, pp. 353–366.
- [5] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. J. Miller, and J. Mylopoulos, "The hyperion project: from data integration to data coordination," *SIGMOD Rec.*, vol. 32, no. 3, pp. 53–58, 2003.
- [6] O. Wolfson, B. Xu, H. Yin, and H. Cao, "Search-and-discover in mobile p2p network databases," in *ICDCS*, 2006, p. 65.
- [7] E. Papapetrou, E. Rova, A. Zarras, and P. Vassiliadis, "Cross-layer networking for peer databases over wireless ad-hoc communities," in *IEEE ICC '07*. IEEE, 2007, pp. 3443–3448.
- [8] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 2, pp. 111–123, 2002.
- [9] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *MobiHoc '02*, 2002, pp. 194–205.
- [10] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.
- [11] K. Fall and K. Varadhan, "The ns manual," VINT Project, Univ. California, Berkeley, CA, 2001.
- [12] J.-Y. L. Boudec and M. Vojnovic, "Perfect simulation and stationarity of a class of mobility models," in *In Proc. of INFOCOM '05*, 2005.