DTFR: A Geographic Routing Protocol for Wireless Delay Tolerant Networks

Anna Sidera
Department of Electrical and Computer Engineering
University of Cyprus, Cyprus

Stavros Toumpis
Department of Informatics
Athens University of Economics and Business, Greece

***Abstract*—We introduce the *Delay Tolerant Firework Routing (DTFR)* protocol, a protocol designed for routing in wireless Delay Tolerant Networks (DTNs) that consist of very large numbers of highly mobile nodes. Under DTFR, each packet initially travels, using high priority transmissions, to a target region in the network where the destination is expected to be. Once there, the packet is replicated to a number of copies that spread across the target region, in search of the destination. As soon as a copy finds a known route to the destination, it follows it and gets delivered.**

To evaluate DTFR's performance, we have developed a simulation tool that can handle networks with numbers of nodes on the order of 10^4. The simulation is optimized for use in DTNs and is very detailed, taking into account, among other things, the Media Access sublayer and the contents of buffers.

Our protocol is compared against *(i) Spray and Wait*, *(ii) GeoCross*, *(iii) GeoDTN+Nav*, *(iv)* a simple flooding protocol (chosen as one extreme of the design space), and *(v) Bethlehem Routing (BR)*, an idealistic protocol that upper bounds the performance of a wide class of protocols. For a wide range of parameters, our protocol is superior (in terms of packet delay and aggregate throughput) to Spray and Wait, GeoCross, GeoDTN+Nav, and the flooding protocol, and performs close to the Bethlehem upper bound.

I. Introduction

This work investigates the problem of routing in wireless Delay Tolerant Networks (DTNs). These are wireless ad hoc networks that are characterized by very large delays in the delivery of data, to the point where the topology of the network typically changes substantially while a packet is *en route* to its destination. This delay is in some cases unavoidable, because the network is never totally connected (as in the case of intermittently connected networks [1], like ZebraNet [2]). In some other cases, this delay could be eliminated, but it is conscientiously traded off to achieve higher levels of throughput [3].

We introduce a novel routing protocol termed *Delay Tolerant Firework Routing (DTFR)*. DTFR can be used in many DTN settings, but it is designed with an eye to DTNs that are characterized by a very large number of nodes (on the order of many thousands) with high levels of mobility, and where each node has means of establishing its own location. A typical example of such networks are vehicular DTNs [4], [5].

In [6], wireless ad hoc networks are classified in four categories: *(i)* End-to-end paths exist almost always, *(ii)* End-to-end paths exist for some fraction of the time, *(iii)* The network is always partitioned, *(iv)* Multi-hop paths are rare.

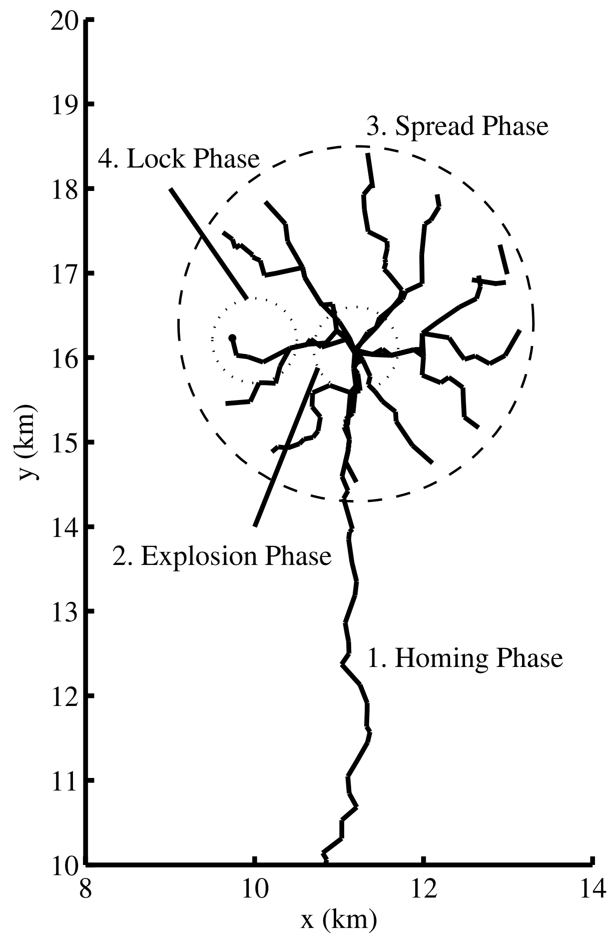

Fig. 1. Example trajectory of a packet being routed with DTFR. The dot within the circle of the Lock Phase shows the position of the destination at the time the packet is delivered. The source is outside the figure.

DTFR is designed for use in categories *(ii)* and *(iii)* of the above classification.

In summary, DTFR works as follows (see Section IV for a detailed description): upon the creation of a packet, the source uses past information about the location and the mobility pattern of the destination in order to send the packet to a location termed the Firework Center (FC), where the destination is estimated to be at the time of the packet's arrival. The packet travels there using a delay tolerant version of

978-1-4577-0899-2/11/$26.00 ©2011 IEEE

geographic routing. In particular, the packet gets transmitted from node to node, in a greedy fashion towards the FC, as long as nodes are found closer to the FC than current holders. When there is no such node, the packet simply sojourns at its current holder, awaiting the emergence of a suitable neighbor, so that the greedy mode can resume. We call this phase the *Homing Phase*. Once the packet arrives at the FC, it enters the *Explosion Phase*, during which a number of replicas are created. Then, during the *Spread Phase*, the replicas propagate toward various directions, so as to maximize the chances that they will find the destination. If, at any time, a replica encounters a known route to the destination, the packet enters the *Lock Phase*, traveling to the destination using that route. Whenever a replica reaches the boundary of the region where the destination is expected to be, it is discarded.

A snapshot of the operation of the protocol, taken from our simulation tool, appears in Fig. 1. Notice that the trajectories of the packet and its replicas resemble the pattern made by an exploding 'palm tree' firework [7], hence the name of our protocol.

The rest of this work is organized as follows: in Section II we review related work. In Section III, we review the basic assumptions about the capabilities and constraints of our target wireless network that influence the design of our protocol, which is presented in Section IV. In Section V we present simulation results and in Section VI we conclude.

II. Related Work

Research on DTNs has received impetus from two different directions: the existence of a delay-throughput tradeoff in mobile wireless networks (where large delays are a design choice), and the existence of intermittently connected mobile wireless networks (where large delays are unavoidable).

Regarding the delay-throughput tradeoff, it was first observed in the seminal work of Grossglauser and Tse [8] that it is possible for a network of n wireless *mobile* nodes to achieve an aggregate throughput on the order of n, provided the nodes are willing to tolerate very large delays (later shown to be on the order of n [3]). This aggregate throughput is much larger than $\sqrt{n}$, which had been shown to be the theoretical limit in the case of networks of *immobile* nodes [9]. This impressive result was soon followed up by a number of results that explored the tradeoff more (see, for example, Toumpis and Goldsmith [3] and references therein).

Concurrent to this line of research were a number of works showing that many important applications can be designed for use over networks that are, per force, intermittently connected. For example, the problem of collecting data from zebras that move in a large area is investigated in [2]. Animals are tagged with wireless devices that can exchange packets only if the animals are within a small physical distance of each other, and the resulting network is always partitioned. These applications contributed to the introduction of the Delay Tolerant Network (DTN) concept, in [10], and the study of routing algorithms for such networks.

Next, we review those routing protocols that are most relevant to our work.

The MDDV (Mobility-centric approach for Data Dissemination in Vehicular networks) protocol [11] is based on two phases. During the *Forwarding Phase*, the message travels to the destination region, and then, in the *Propagating Phase*, it is distributed to all nodes there. In the Forwarding Phase, a group of nodes are forwarding the message along a trajectory consisting of road segments chosen by the protocol. The group consists of the nodes that estimate that they are near the *message head* which is the node closest to the destination region along the trajectory. The members of the group change as the message propagates or the vehicles move. Nodes estimate the position of the message head based on information that is inserted in the copies of the packet, by nodes that estimate they might be the message head. In DTFR, the packet also travels to the location of the destination during the *Homing Phase*, but using a delay tolerant version of geographic routing and high priority transmissions. In addition, during the *Homing Phase*, there is only a single copy of the packet at any time, and if the node that has that copy moves away from the Firework Center, it still has to forward the copy. Finally, our Explosion, Spread, and Lock phases are more efficient from the Propagating phase of MDDV which distributes the packet to all nodes in the destination region.

In [12], the *Spray and Wait* protocol is proposed. Spray and Wait consists of two phases. In the *Spray Phase*, the source distributes L copies to L distinct relays. In the *Wait Phase*, the relays move around the network, until eventually one of them meets the destination and hands over its replica of the packet. The authors also introduce the *Spray and Focus* protocol which has a *Spray Phase*, as in Spray and Wait, and a *Focus Phase*. In the *Focus Phase* a node A carrying a copy of a message for node D decides if it will forward its copy to another node B within its communication range, based on information about previous contacts of A and B with D. Our protocol also employs replicas, however, here, the replicas are created not at the location of the source, but at a location estimated to be close to the destination, in order to conserve bandwidth. In addition, as we will show, under our protocol, nodes make use of the geographic information available to them to perform a more efficient search of the network during the spread phase.

GPSR (Greedy Perimeter Stateless Routing) [13] uses a combination of greedy forwarding on the full network graph and perimeter forwarding on a planarized network graph, i.e., a subgraph of the original graph with no crossing links. Initially, the packet is forwarded on the full network graph using the greedy mode; if, at some point, there is no neighbor closer to the destination than the node holding the packet, the packet enters the perimeter mode, traversing the faces of the planarized network graph using the right hand rule [13]. If the packet, while in perimeter mode, reaches a node closer to the destination than the point at which the packet entered the perimeter mode, the packet switches back to the greedy mode.

In [14] the authors propose *GPCR* (Greedy Perimeter Coordinator Routing), a protocol designed for use in vehicular networks. GPCR is based on the observation that the road network creates a naturally planar graph that can be exploited for communication purposes. Both greedy routing and perimeter routing are executed on that graph. However, GPCR suffers from the problem that when there is no node at a junction, packets will be forwarded across that junction. Forwarding packets across empty junctions is likely to lead to a non-planar graph that causes routing loops. To alleviate this problem, the GeoCross protocol is introduced in [15]. GeoCross is similar in its operation to GPCR, but its perimeter mode is enhanced and capable of detecting and removing crossing links and creating a planar graph.

GeoDTN+Nav [5] consists of the greedy and perimeter modes of GeoCross and a third mode, termed the DTN mode, which can deliver packets even in the absence of end-to-end routes. In GeoDTN+Nav, packets are first forwarded using the greedy mode and, when this fails, using the perimeter mode. If the perimeter mode also fails, the protocol finally switches to the DTN mode and relies on mobility to deliver packets. To decide when to switch to the DTN mode, a node uses a cost function related to network partition detection and to the navigation information of its neighbors. When a packet is in the DTN mode, it returns to the greedy mode when it encounters a node that is closer to the expected location of the destination than the point where the perimeter mode started.

DTFR and GeoDTN+Nav are related, as they both employ a geographic routing mode and a DTN mode. However, they have a number of key differences. Firstly, GeoDTN+Nav makes use of a perimeter mode, which DTFR avoids, in order to conserve bandwidth, and in order to avoid the routing loops associated with running a perimeter mode in a network of highly mobile nodes. Secondly, GeoDTN+Nav was designed without taking into account links between nodes that are not on the same road and so makes no use of potentially useful links between nodes lying on different roads. Thirdly, in GeoDTN+Nav, the packet only travels to the destination position inserted in the packet by the source, whereas, in DTFR, if the destination is not found in the *Homing Phase*, the *Explosion Phase* and the *Spread Phase* take place. Also, the rules for entering the greedy mode from other modes are different. Finally, DTFR uses a set of priority rules for gaining access to the medium. As we show in the simulation section, all these differences lead to significant deviations in the performance of the two protocols.

We note that the term ‘Firework’ has also been used in Peer-to-Peer Networking [16], [17] where a content-based ‘Firework Query Model’ is proposed. Also, in [18] a multicast protocol called ‘Fireworks Routing’ is presented, for use in a general, non-DTN, multicast wireless ad hoc setting. This protocol organizes multicast group members into *cohorts*. One group member in each cohort is selected to be a *cohort leader*. Cohort leaders establish a sparse multicast tree among themselves and the source and they use broadcasting to deliver the packets to other group members in their cohort. Although Fireworks Routing and DTFR have a number of similarities, they also have key differences. DTFR applies a delay tolerant version of geographic routing, avoids broadcasting in order to conserve bandwidth, and prioritizes transmissions. Also, in DTFR no multicast structure is maintained and the packet is not given to nodes near a cohort leader but to nodes in a geographic region where the destination is estimated to be. As the works in [16], [17], [18] appear in very different contexts, a meaningful comparison with them is not possible.

III. Basic Network Assumptions

In this section we outline our fundamental assumptions on the network. These assumptions influence our design choices and frame the scope of our work.

Node Mobility: We assume a large number of nodes moving in a region independently of each other, and independently of their communication needs. Moreover, each node is capable of knowing its own location, either directly (for example, through GPS) or indirectly (for example, using beacons).

Communication Needs: Nodes are executing one or more applications that depend on recurring communication between node pairs. (One member of the pair could be an Access Point, or similar entity, communicating with multiple nodes.) Nodes are equipped with wireless transceivers and buffers of finite size, thus forming a wireless ad hoc network that can forward packets to their destination either directly or via other nodes. The application(s) running at each node are delay tolerant, however there is a maximum acceptable delay for the delivery of the packets. In addition, nodes would like to maximize the amount of traffic that the wireless network can carry. Therefore, the available bandwidth should be used judiciously, and this means that many small transmissions are preferable to a few long ones [9]. The nodes can inform their neighbors on their location.

Neighborhood Awareness: We assume that each node is aware of the network in its local neighborhood. In the most limiting case, this means that the node is aware of all nodes close enough for direct communication with them to be possible. In the more general case, each node knows all the nodes within a number of hops away. Nodes can use this information in order to route packets in their local neighborhood.

Although these assumptions can be satisfied in a variety of settings, a good example are the large vehicular Delay Tolerant Networks discussed in [4], [5].

IV. The DTFR Protocol

The DTFR protocol consists of four mechanisms: 1) a *Dissemination Rule*, responsible for disseminating a number of replicas in the vicinity of the destination, 2) a *Forwarding Rule*, responsible for node-to-node packet forwarding, 3) a *Priorities Policy*, for assigning priorities to nodes contending for access to the wireless medium, and 4) a *Buffer Policy*. We now discuss each of these.

We stress that some implementation details of these mechanisms will depend on the details of the application, and so are left undefined in this section. In Section V the implementation details are given for the setting specified there.

A. Dissemination Rule

The dissemination rule of DTFR consists of four phases:

1) *Homing Phase*: The packet travels to a point called the *Firework Center (FC)*, at the center of a region where the source estimates the destination to be, taking into account the packet travel time. Depending on the application and mobility model, a variety of choices is possible.
2) *Explosion Phase*: Then, the packet is replicated to L copies and given to L relays.
3) *Spread Phase*: Then, the copies travel, in a multihop way, to L different points called the *Firework Endpoints (FE)*, that are symmetrically placed around the FC, at a distance D from it. Once there, the replicas are discarded. The distance D is chosen to be such that the destination will be between the FC and the Firework Endpoints (FEs) with high probability, taking into account the nature of the node mobility.
4) *Lock Phase*: At any time during the first three phases, if a packet comes near enough to the destination to discover a multihop route, it enters the *Lock Phase* wherein it is forwarded to the destination using that route.

B. Forwarding Rule

We now describe the rule used for selecting the next node that will relay a packet as it travels through the network.

In the *Homing* and *Spread* phases, packets travel toward the given target location (i.e., the FC or an FE) in a greedy manner, as in many geographic routing protocols [19], at each node selecting a neighbor that is more near to the target location. If there is no node in the neighborhood nearer to the destination than the current holder of the packet, then the node simply keeps the packet, until a node with positive progress is found.

Observe that, as with all other geographic routing protocols, our forwarding protocol is greedy. However, in contrast to them, it is also lazy: upon failure to find a next hop, rather than resort to a more thorough search, it does nothing and just waits for the topology to get better. We believe that, in delay tolerant networks, this combination of greediness with laziness makes sense. Our simulations verify that this combination leads to a robust design, with very high probability of packet delivery.

During the *Explosion Phase*, the current holder of the packet broadcasts it until L replicas are residing in L different nodes. Each replica is assigned to a different FE.

Finally, if the *Lock Phase* is initiated, the packet (or replica) is sent over that route in the usual, non-DTN, multihop fashion.

C. Priorities Policy

In order to access the medium, packets are given different priorities, depending on the phase they are in. *Lock Phase* transmissions have priority over transmissions of all other phases. This is because when a packet goes near its destination we do not want to loose the opportunity to deliver it, given the changing topology. *Homing Phase* transmissions have priority over *Spread Phase* transmissions and *Explosion Phase* transmissions, as we do not want to delay the only copy of a packet from reaching the Firework Center and so delay the search in the whole region near its destination. *Explosion Phase* transmissions have priority over *Spread Phase* transmissions, as we want to create all replicas quickly.

D. Buffer Policy

The buffer of each node has a finite size B. Once a buffer is full, the node cannot receive any packet unless it is destined for that node, and must discard the packets its user creates. Packets are discarded when they reach the Firework Endpoints. Also, the packets have a time to live (TTL), equal to the maximum acceptable delay for the delivery of the packets. When the TTL elapses, packets get discarded.

V. Results

A. Simulation Tool

In order to evaluate our protocol, we have developed a simulation tool specifically designed for DTNs, and written in C. Using NS (or a similar tool) was not an option, as *(i)* we are interested in simulating very large networks, with many thousands of nodes, which NS cannot handle effectively, and *(ii)* DTN routing is very different from traditional routing, for which NS has been optimized. Efforts have been made to make the simulations as accurate as possible. Among others, *(i)* full buffer information for all nodes is kept, *(ii)* a realistic physical layer (described below) is used, and *(iii)* contention in the channel is taken into account. At the same time, efforts have been made so that the simulator is as fast as possible and as a result, the tool is capable of detailed simulations of networks of more than 10^4 nodes on a desktop computer, and for a variety of routing and media access protocols. Challenging simulations with 10^4 nodes take at most a few hours, depending on the protocols used and the various inputs. The source code, detailed instructions for its use, and example outputs are available in [20].

B. Network Setting

In the following, we specify the setting used for the simulations of DTFR and the protocols we compare it with. We stress that DTFR, as described in Section IV, can be applied in any setting that satisfies the basic assumptions of Section III. Here, we list the extra assumptions needed in order to arrive at a quantitative evaluation of our protocol.

Mobility Model: Nodes move on a square grid composed of vertical and horizontal roads. Each node travels, along the road network, to a randomly chosen location, using a constant speed, uniformly distributed between 0 and $v_{\max}$. Then, it chooses another random location, and another speed, moves to that location, and so on.

Traffic Pattern: All nodes are divided in pairs, each node communicating with its counterpart. Pairs do not change for the whole duration of the simulation.

Channel Model: We assume an urban environment that prohibits Line Of Sight (LOS) communication between two nodes, unless both of them lie on the same road. If a node is within a threshold distance R_T from the intersection between two roads we assume that this node belongs to both roads. In the case of LOS transmissions, the signal power P_r received at distance d from a transmitter is calculated using the model

$$P_r = P_0 \left(\frac{d_0}{d}\right)^{\alpha_{\mathrm{LOS}}}$$

and, for Non Line of Sight (NLOS) transmissions, using

$$P_r = P_0 \left(\frac{d_0}{d}\right)^{\alpha_{\mathrm{NLOS}}}$$

where P_0 is the received power at a reference distance d_0 from the transmitter and α_{LOS} and α_{NLOS} are exponents that describe the environment, typically 2-6 [21], with $\alpha_{\mathrm{LOS}} < \alpha_{\mathrm{NLOS}}$.

Transmitter Model: While transmitting, a node cannot listen to the transmissions of other nodes. If node k is not a transmitter, a packet from node i is received successfully at node k if the following relationship holds:

$$\frac{P_{ik}}{\sum_{j \in S, j \neq i} P_{jk} + N} > \gamma_T, \tag{1}$$

where N is the background noise, γ_T is the minimum Signal to Interference plus Noise Ratio (SINR) required at the receiver, P_{jk} is the received power at node k from node j, and S is the set of all transmitters.

Slotted Time: We slot time, and at the start of each timeslot each node creates a packet with a predefined probability λ. The packet is immediately stored in the buffer if it is not full. The transmission of each packet takes one timeslot. Timeslots are assumed to be so short, that the topology cannot change appreciably for a timeslot duration, and hence channel gains are constant throughout each of them.

Medium Access Control: At the start of each slot, nodes employ a MAC scheme to decide who will transmit at that slot, what packet, and to whom. At any given time during the execution of this scheme, the state of a node can be either *available* or *reserved*. At the start of the slot, all nodes are available, but progressively attempt to make reservations, according to their priorities. For a node A to be able to send a packet to another node B, both A and B must not be reserved. If this is the case, nodes A, B and all the nodes within distance $K \cdot d_{AB}$ from A or B become *reserved*. K is a constant greater than 1, which we term the *Reservation Radius Constant*. As we are not interested in the evaluation of the MAC layer, we assume that the reservations are all arranged instantaneously, at the start of each slot, and no MAC control messages are simulated.

Power Control: If, after the MAC scheme is executed, node A has decided to transmit a packet to another node B, A uses a power level P_t such that the transmission will be successful if the interference from competing transmissions turns out to be at most $(I_f - 1)$ times the thermal noise, where I_f is a constant we call the *Power Control Safety Margin*. Also there is a maximum allowed transmission power $P_{0\max}$, and communication is not permitted between nodes for which the required power P_t turns out to be greater than $P_{0\max}$. Therefore, node pairs perform a rudimentary power control algorithm, in order to avoid creating too much interference for the other sender-receiver pairs while, at the same time, transmitting with enough power to have a successful transmission.

Local Routing Table: As already discussed, nodes maintain a routing table that can be used for routing in their immediate neighborhood. To conserve bandwidth and improve robustness, nodes do not use local routes that minimize the number of hops. Rather, a link cost is introduced, and nodes try to use paths with minimum total link cost. In particular, each LOS transmission from a node A to another node B is associated with a cost d_{AB}^2, where d_{AB} is the distance between nodes A and B. Each NLOS transmission is associated with a cost $d_{AB}^{\frac{2\alpha_{\mathrm{NLOS}}}{\alpha_{\mathrm{LOS}}}}$. These costs are chosen so that nodes avoid transmissions over large distances and over high-loss links; therefore, we can have more simultaneous transmissions, and experience a higher overall throughput as predicted in [9]. The routing table includes destinations for which there is a path with total link cost at most equal to a threshold value C_T, which we term the *Local Routing Threshold*.

We do not simulate control messages for the creation of the local routing table. Therefore, interference experienced by data packets comes only from data packets. Furthermore, we assume that local routing is always executed perfectly. We believe that, as we are interested in the more challenging case of communication across large distances, these assumptions, that essentially remove local routing issues from the picture, are justified.

C. Implementation Details of Simulated Protocols

Bethlehem Routing: Under *Bethlehem Routing (BR)*, each packet is continuously aware of the location of its destination, and moves towards it by continuously staying in the Homing Phase with the location of the destination chosen as the FC. Once near enough to the destination to discover a route, the packet will enter the *Lock Phase*. Subject to this modification, BR is identical to DTFR. Clearly, BR is a very simple and not very realistic protocol, as nodes make use of location information available to them only in very ideal situations. However, it allows us to estimate the cost of not having this information in practical protocols, and hence its performance represents an interesting upper bound to the performance of all protocols that use estimates on the location of destinations. The performance is still limited by the fact that there may not be a neighbor more near to the destination and the packet may need to wait at certain relays, and also because of congestion.

Flooding: Under the flooding protocol, each node sends copies of all packets it has in its buffer to all nodes it meets. All transmissions have the same priority.

Spray and Wait: The Spray and Wait protocol is described in [12]. In our implementation, and in order to have a more fair comparison to DTFR, nodes make use of the local routing table. *Lock Phase* transmissions have priority over *Spray Phase* transmissions.

PARAMETER	NUMERICAL VALUE
Slot Duration	0.01 sec
Packet arrival rate	$\lambda = 0.02$ packets/sec/node
Number of nodes	$n = 5000$
Side of the grid in which the nodes move	7 km
Distance between junctions	200 m
Junction radius	$R_T = 10$ m
Maximum node speed	$v_{\max} = 10$ m/sec
Packet TTL	6 min
LOS exponent	$\alpha_{\mathrm{LOS}} = 3$
Non-LOS exponent	$\alpha_{\mathrm{NLOS}} = 5$
Propagation model reference distance	$d_0 = 1$ m
SINR Threshold	$\gamma_T = 10$
Power Control Safety Margin	$I_f = 10$
Thermal Noise over Transmitter Power	$\frac{N}{P_{0\max}} = 1.25 \cdot 10^{-9}$
Local Routing Threshold	$C_T = 4 \cdot 10^4$
Simulation duration	1 hour
Buffer size	$B = 10^4$ packets
Reservation Radius Constant	$K = 2.5$

TABLE I
DEFAULT SIMULATION PARAMETERS

GeoCross and GeoDTN+Nav: In our implementation of GeoCross and GeoDTN+Nav, nodes make use of the local routing table. *Lock Phase* transmissions have priority over all other transmissions. Greedy mode transmissions, perimeter mode transmissions, and DTN mode transmissions are equal in priority, but transmissions from junction nodes have priority over transmissions from street nodes. As in [5], we assume that the source knows the position of the destination at the time the packet is created and sends the packets at that location.

DTFR: To ensure a fair comparison with GeoDTN+Nav, the FC is set to be the position of the destination at the time of the packet's creation, and it does not change with time[1].

Bethlehem GeoDTN+Nav: We also simulate a protocol that we call Bethlehem GeoDTN+Nav (BetGeo), which is identical to GeoDTN+Nav except from one point: whenever a routing decision is made that involves the location of the destination, instead of using the position that the destination occupied at the time of the packet's creation, its current position is used. As with the Bethlehem protocol, this is an idealization, however the performance of this protocol allows us to evaluate the cost on the performance of GeoDTN+Nav of using location information that is not current.

More details on the implementation of the various protocols appear in the technical report in [20].

D. Simulation Results

All simulation results appear in Fig. 2. Unless otherwise stated in each particular case, the parameters used are the default ones appearing in Table I. For each point in the plots we simulate each protocol for different values of its various parameters, and select the values that produce the best results.

[1]In [20] a detailed comparison of DTFR with Spray and Wait is offered, for networks of up to 10^4 nodes, under a Random Waypoint mobility model. There, a node A wishing to send a packet to B chooses for the FC the location where B is estimated to be based on its location and destination of movement at the time of the creation of the latest packet from B to reach A.

In Plot (a), we show the packet delivery ratio versus the packet arrival rate. Even with very small arrival rates, no protocol manages to deliver all packets within the TTL. This is due to the fact that the network is often partitioned for periods of time comparable or larger than the time to live. In addition, quite often the network is not partitioned but bottlenecks are formed due to the topology, leading to queueing delays.

Observe that the delivery rate of GeoDTN+Nav is significantly smaller than the delivery rate of DTFR. There are a number of reasons for this. First of all, DTFR uses the explosion, spread, and lock phases to counter the fact that the destination is moving. No similar mechanisms exist in GeoDTN+Nav. Note, however, that even with Bethlehem GeoDTN+Nav, where GeoDTN+Nav is enhanced so that the packets have continuous perfect knowledge of the position of their destination, the delivery ratio improves modestly over GeoDTN+Nav. Secondly, under GeoDTN+Nav it is possible that packets leave the perimeter mode and enter the DTN mode at a node that is further away from the destination than the node they were when they entered the perimeter mode. In between, they were transmitted multiple times, wasting precious bandwidth in the process. DTFR, on the other hand, never backtracks, opting instead to let the packet wait. Thirdly, under GeoDTN+Nav packets stay in the DTN mode even when there are neighbors of the current holder closer to the destination, because their distance to the destination is greater than the distance between the destination and the point where the packet entered the perimeter mode. Under DTFR, on the other hand, nodes always send packets to neighbors closer to the destination than themselves.

In Plot (b) we plot the average delay (over all delivered packets) versus the arrival rate. It is interesting to note that BR has larger delays on the average than DTFR and, in some cases, GeoDTN+Nav and GeoCross. This is due to the fact that BR delivers the most packets, including some with large delays. GeoCross and GeoDTN+Nav exhibit the smallest delays in large arrival rates, but note that they deliver significantly fewer packets.

In Plot (c), we show the packet delivery ratio versus the network size. We change the network size by changing the number of nodes and the dimensions of the area, keeping the number of nodes per unit road length constant. Note that the performance of all protocols diminishes with the network size. This is due to the fact that the TTL counter remains fixed, and with larger network sizes come more frequent partitions.

In Plot (d), we depict the average delay versus the network size. The four protocols that have packets travel to the estimated location of the destination have comparable performance, which surpasses the performance of flooding and Spay and Wait. Plots (c) and (d) reveal that the performance of DTFR deteriorates the slowest as the size of the network increases, with respect to all other protocols, except BR. We attribute this to its robust design.

In Plot (e), we show the packet delivery ratio versus the transmission range. We change the transmission range by changing the value of $\frac{N}{P_{0\max}}$. All protocols gain by an increase

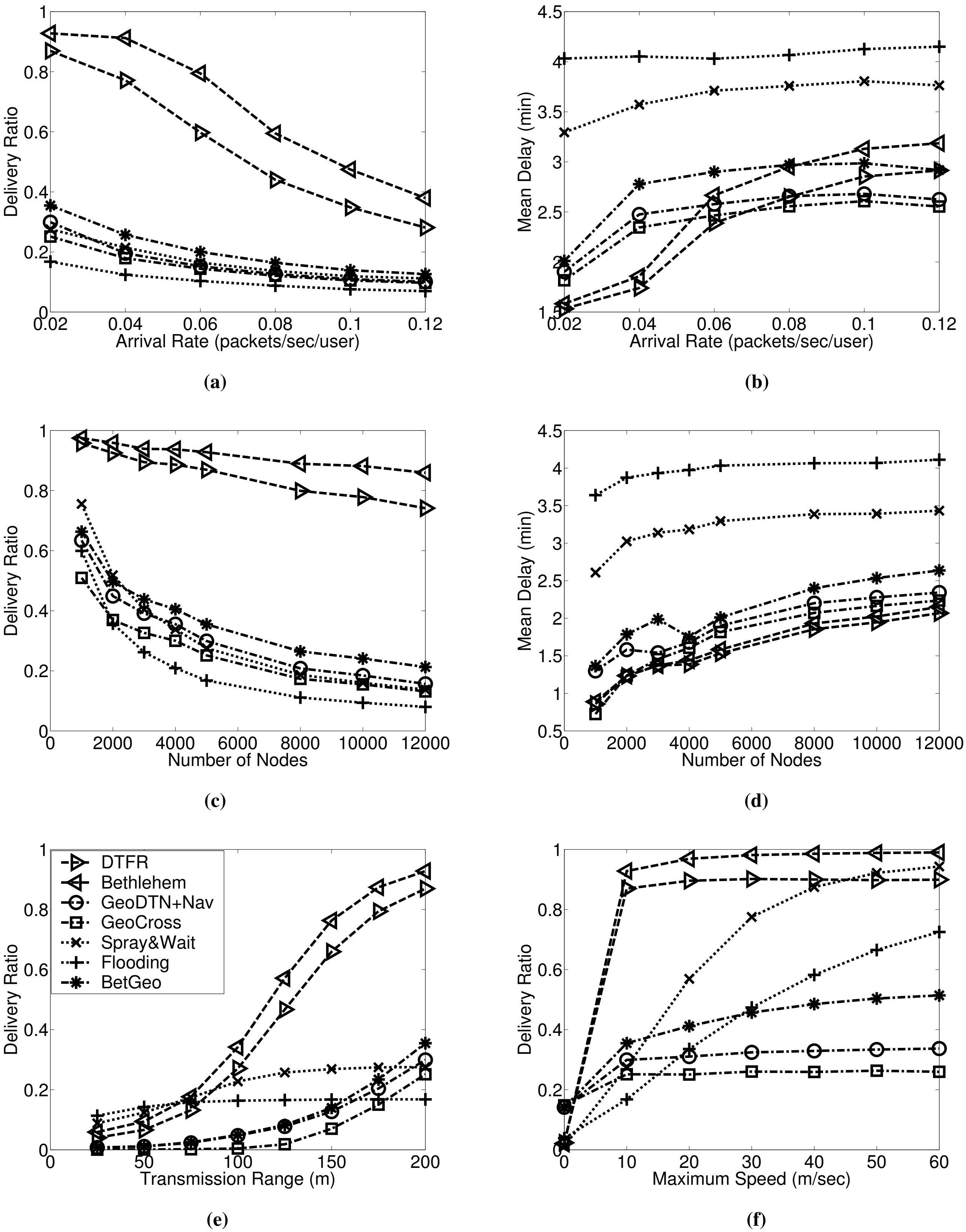

Fig. 2. Comparison of DTFR with other protocols. The legend of Plot (e) applies to all plots. Therefore, for example, the $+$ marker denotes the Flooding protocol in all six plots.

in the transmission range, however the two protocols that do not depend on the fast forwarding of the packets to the area where the destination is expected, Spray&Wait and Flooding, benefit the least. On the other hand, Spray and Wait is slightly superior to the rest (except flooding) in the case of small transmission ranges. The performance of GeoCross and GeoDTN+Nav increases fast as the transmission range increases, because the perimeter mode becomes more efficient for larger transmission ranges.

In Plot (f), we show the packet delivery ratio versus the maximum speed of the nodes. For high speeds, the performance of Spray and Wait surpasses the performance of DTFR. Clearly, when the node mobility is too high, the best strategy for the source is to get out as many replicas as possible. Also observe that, in the other extreme, when nodes are immobile, GeoCross gives better results than DTFR. This is expected: when a packet reaches a local optimum, and nodes are immobile, waiting is futile, and the only alternative is going into perimeter mode. However, for all the cases in the middle, DTFR is surpassed only by BR. Note that to obtain the points in Plot (f) for 0 velocity, we averaged multiple runs of the simulation, each of them for a different network topology chosen randomly from steady state topologies.

We also simulated GeoCross using the parameters of Table I but with a very low arrival rate of 10^{-4} packets/sec/node, immobile nodes (i.e., $v_{\max} = 0$), and a very large number of permitted hops in the perimeter mode, $h_{\max} = 10^4$. It was found that only 42% of the packets reached their destination, although an end-to-end path existed for 47% of the node pairs. Therefore, although GeoCross is a major breakthrough over GPCR, it does not altogether eradicate routing loops.

VI. Conclusions

DTFR is designed for large networks of mobile nodes that are able to establish their positions and where nodes communicate on a regular basis with some destination, so that an estimation for the location of that destination may be maintained. As the simulations show, DTFR can handle very challenging cases where there is a very large number of highly mobile nodes, as is the case with vehicular networks [4], [5].

The performance of DTFR is evaluated using simulation. Large networks are simulated, using a very detailed simulation tool, that takes into account the physical layer and the state of all buffers. DTFR is compared with *Spray and Wait*, *GeoDTN+Nav*, *GeoCross*, *Bethlehem Routing* (an idealized protocol that serves as an upper bound), and Flooding.

Although we have presented DTFR in the context of unicast traffic between nodes, we note that it can also support other types of traffic. Notably, it can be used for cooperative content sharing [4], [22]. We now briefly elaborate on this. (A detailed exposition is subject for future work.) In [4], a node A disseminates information about data it has via k-hop broadcasting. A node B that receives this information can send a query to A about data it needs. Then A can send the data to B. The query and the data are sent using AODV. This could be a possible application of DTFR. In the case of first time communication, no knowledge of the destination location is needed, as the message is send using limited flooding. After this, nodes have an estimate of the location of their destinations, and so can exchange the rest of the messages using DTFR.

Acknowledgment

Research for this work was partly funded by the European Union IST-FP6-FET project NETREFOUND. We would like to thank Pei-Chun Cheng for many useful discussions about the implementation and operation of GeoDTN+Nav.

References

[1] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE Trans. Networking*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[2] P. Juang, H. Oki, Y. Wang, and M. Martonosi, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiments with zebranet," in *Proc. ASPLOS-X*, San Jose, CA, Oct. 2002.

[3] S. Toumpis and A. J. Goldsmith, "Large wireless networks under fading, mobility, and delay constraints," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar.-Apr. 2004.

[4] K. Lee, S.-H. Lee, R. Cheung, U. Lee, and M. Gerla, "First experience with CarTorrent in a real vehicular ad hoc network testbed," in *Mobile Networking for Vehicular Environments*, Anchorage, AK, USA, May 2007, pp. 109–114.

[5] P.-C. Cheng, K.-C. Lee, M. Gerla, and J. Harri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[6] S. Heimlicher, M. Karaliopoulos, H. Levy, and T. Spyropoulos, "On leveraging partial paths in partially-connected networks," in *IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 55–63.

[7] http://en.wikipedia.org/wiki/Fireworks\#Palm.

[8] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc. IEEE INFOCOM*, vol. 3, Anchorage, AL, Apr. 2001, pp. 1360–1369.

[9] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[10] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Applications, Technologies, Architectures, and Protocols for Computer Communication*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[11] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: a mobility-centric data dissemination algorithm for vehicular networks," in *ACM MOBICOM*, Philadelphia, Pennsylvania, USA, Sep. 2004, pp. 47–56.

[12] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multi-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 77–90, Feb. 2008.

[13] B. Karp and H.-T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom*, Boston, MA, Aug. 2000.

[14] C. Lochert, M. Mauve, H. Fubler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, Jan. 2005.

[15] K.-C. Lee, P.-C. Cheng, and M. Gerla, "Geocross: A geographic routing protocol in the presence of loops in urban scenarios," *Ad Hoc Networks*, vol. 8, pp. 474–488, Jul. 2010.

[16] C. H. Ng, K. C. Sia, and C. H. Chan, "Peer clustering and firework query model in the peer-to-peer network," in *12th Int. WWW Conference*, 2003.

[17] K. C. Sia, C. H. Ng, C. H. Chan, and I. King, "P2P content-based query routing using firework query model," in *2nd Int. Workshop on Peer-to-Peer Systems*, 2003.

[18] L. Law, S. Krishnamurthy, and M. Faloutsos, "Fireworks: An adaptive group communications protocol for mobile ad hoc networks," in *IFIP Networking 2005*, Waterloo, Canada, May 2005.

[19] R. Catanuto, S. Toumpis, and G. Morabito, "Opti{c,m}al: Optical / optimal routing in massively dense wireless networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007.

[20] http://pages.cs.aueb.gr/~toumpis/dtns.html.

[21] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ: Prentice Hall, 2002.

[22] V. Lenders, G. Karlsson, and M. May, "Wireless ad hoc podcasting," in *IEEE SECON*, San Diego, CA, Jun. 2007, pp. 273–283.