

Autonomous Pickup and Delivery for Delay Tolerant Mobile Networks

Ozcan Koc Justin Yackoski Chien-Chung Shen
Computer and Information Sciences
University of Delaware, USA
{koc,yackoski,cshen}@cis.udel.edu

Abstract—Delay tolerant mobile networks are characterized by the possible non-existence of end-to-end paths. For instance, environmental factors and energy constraints in mobile ad hoc and wireless sensor networks make the preservation of end-to-end connectivity challenging. To overcome the problem, the “store-carry-and-forward” (SCF) paradigm has been proposed, where mobile nodes may physically carry and transport messages between nodes to facilitate eventual message delivery. We propose a suite of autonomous and dynamic SCF algorithms, together called *Pickup and Delivery (PAD) Routing*, which adapt the traditional algorithmic Vehicle Routing Problem (VRP) and Covering Tour Problem (CTP) to the context of delay tolerant mobile networks. The objectives of PAD Routing are to reduce the message transport delay and to increase the delivery ratio in the presence of network partitions. Simulation results demonstrate the efficiency of these algorithms.

Keywords: Delay tolerant mobile networks, pickup and delivery, store-carry-and-forward, Vehicle Routing Problem, Covering Tour Problem.

I. INTRODUCTION

Delay tolerant networks are characterized by the possible non-existence of end-to-end paths. For instance, preserving end-to-end connectivity is not always possible in mobile ad hoc networks, especially in the presence of factors such as node mobility, physical obstacles, large terrain, severe weather, and jamming. Such factors cause networks to partition, either temporarily or permanently. As another example, the severe energy constraint in wireless sensor networks also makes the preservation of end-to-end connectivity challenging.

Many ad hoc routing protocols have been proposed, and their effective operations assume constant network connectivity and employ the “store-and-forward” routing paradigm. However, this paradigm is not applicable to delay tolerant networks due to the fact that it is not always possible to achieve end-to-end connectivity between source/destination pairs. To address the issue, recent research [1], [2], [3], [4], [5], [6] has adopted the “store-carry-and-forward” (SCF) paradigm, where some mobile nodes may physically carry and transport messages between nodes to facilitate eventual message delivery for delay tolerant networks. In addition to its use in routing, such paradigm may also be used by a mobile node to tour a sensor network and retrieve remote sensor information. For instance, to conduct warehouse inventory,

This work is supported in part by National Science Foundation CAREER award CNS-0347460.

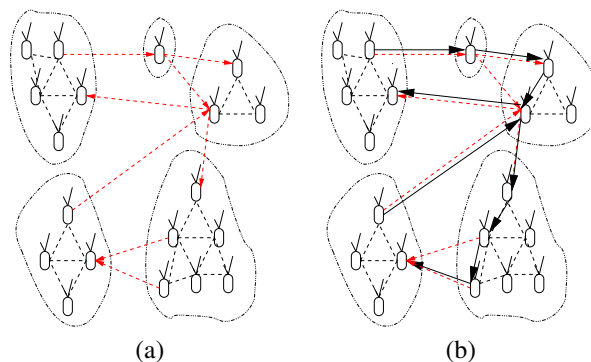


Fig. 1. A delay tolerant mobile network scenario with 5 partitions and 8 requests: (a) message transport requests (dotted arrows), and (b) two tours satisfying the requests (concatenated solid arrows)

mobile RFID readers may tour the warehouse in proximity to the RFID-tagged goods to retrieve the data and carry the inventory information back to a sink.

Similar problems arise in the real world scenarios such as (1) freight routing, which have been formulated as the Vehicle Routing Problem (VRP), where courier vehicles, from companies such as FedEx, UPS, and DHL, plan tours to pickup and deliver goods, and (2) routing of rural health care delivery team, which has been formulated as the Covering Tour Problem (CTP), where medical services can only be delivered to a subset of villages, but individuals at villages not directly on the route must be able to reach a visiting medical team at its nearest stop [7].

Our aim is to develop autonomous and dynamic mechanisms for data retrieval and/or delivery in delay tolerant mobile networks to fulfill message transportation missions in the presence of network partitions. We propose a suite of such mechanisms, together called *Pickup and Delivery Routing*, or PAD Routing for short, which adapts the traditional algorithmic VRP and CTP problems to the context of delay tolerant mobile networks. Our main objectives are to reduce the transport delay and to increase the pickup and/or delivery ratio in the entire network. Figure 1(a) depicts a delay tolerant mobile network with 5 partitions and 8 communication requests, and Figure 1(b) depicts the fulfillment of these requests with 2 tours of 2 couriers. Notice that PAD Routing coexists with traditional ad hoc routing protocols, where the latter are used for intra-partition traffic and the former deliver data across

partitions.

In this paper, we review the traditional VRP and CTP problems, and discuss the fundamental differences from the problems that PAD Routing will address (Section II). Specifically, we describe the PAD Routing problem and solutions that are most relevant to VRP (Sections III and IV). In Section V, we present simulation results of the PAD Routing solutions. Related work is described in Section VI, and Section VII concludes the paper.

II. DIFFERENCES FROM VRP AND CTP PROBLEMS

Both VRP and CTP problems have attracted researchers and practitioners alike due to their many applications. We review the definitions of certain VRP and CTP problems, and discuss the fundamental differences from the PAD Routing problem.

A. Vehicle Routing Problems (VRP)

VRP is a combinatorial optimization problem, which generalizes the Traveling Salesperson Problem (TSP). TSP is the problem of finding a minimal length Hamiltonian cycle of a given weighted graph. Since TSP is NP-hard and VRP is a generalization of TSP, VRP is also NP-hard.

In original VRP, given a fleet of vehicles without capacity constraints, a common depot, a set of customers (represented as a set of geographically scattered points) and their delivery requests (each represented as an arrow pointing from the depot to the customer), the goal is to find a set of routes (or tours) with overall minimum cost which serve all the customer requests. All the tours must start and end at the depot and each customer request must be served exactly once by one of the vehicles. From this definition, one can infer that TSP is VRP with one vehicle of zero capacity, no depot, and customers with no request.

VRP has many variants. In the Capacitated VRP Problem, in addition to the above mentioned constraints, every vehicle has a fixed finite capacity which must be observed. In the Multi-Depot Vehicle Routing Problem, there may be multiple depots from which customers are served, where a fleet of vehicles is based at each depot, and each vehicle originates from one depot, services the customers assigned to that depot, and returns to the same depot. Vehicle Routing Problem with Time Windows is another variant with the additional constraint that a time window is associated with each customer wherein the customer has to be serviced. The goal is to minimize the vehicle fleet and/or the sum of travel time and waiting time needed to service all customers in their required time windows. The Pickup and Delivery Problem with Time Windows (PDPTW) [8] is yet another variant. In PDPTW, the goal is to find a set of optimal routes for a fleet of vehicles in order to serve transportation requests, where each transportation request is defined by a pickup location, a delivery location, pickup time and/or delivery time, and a load. When the size of the fleet is fixed to m vehicles, the problem is called m -PDPTW.

B. Covering Tour Problems (CTP)

CTP is also a generalized version of TSP. Given an undirected graph, CTP computes a minimum cost cycle passing

through a subset of vertices such that every vertex not in the cycle lies within a given distance from at least one node in the cycle [7]. For instance, the routing of rural health care delivery teams in developing countries can be modeled as CTP, where the objective is not to “serve” each request, but rather to effectively “cover” each destination by being physically close to the requesting site.

CTP is formally defined as follows. Let $G = (V, E)$ be an undirected graph, where $V = X \cup Y \cup Z$ is the set of vertices and E is the set of edges. X is the set of vertices that *must* be visited, Y is the set of vertices that *may* be visited in addition to X , and Z is the set of vertices that *must* be covered. Each edge $ij \in E$ is assigned a non-negative cost c_{ij} and a *covering distance* d_{ij} associated with vertex i . A vertex $j \in Z$ is covered by a vertex $i \in X$ if and only if $d_{ij} \leq D_i$. The objective is to find a minimum cost cycle (tour) connecting vertices in X such that each vertex in Z is covered by the cycle.

m -CTP is a generalization of CTP [9], where the goal is to find a set of m routes of minimum total length satisfying the following constraints: (1) there are at most m vehicle routes, and each of them starts and ends at the same depot, (2) each vertex of Y belongs to at most one route, while each vertex of X belongs to exactly one route, and (3) each vertex of Z must be covered by a route in the sense that it lies within a preset distance D_i of a vertex i in X belonging to a route.

C. Differences from the PAD Routing Problem

First, we describe the difference between the “on-line” and the “off-line” VRP problems. A VRP problem is “off-line” if all the requests are known in advance. In contrast, for an “on-line” problem, requests are generated dynamically over time, which need to be incorporated into tour planning on-the-fly. Almost all VRP research work concentrates on improving the performance of off-line algorithms. Although these studies have great theoretical contributions, they are not applicable to mobile networking scenarios where it is impossible to know all message pickup and delivery requests beforehand.

m -PDPTW is the VRP variant that is most relevant to PAD Routing. Although PDPTW is a well studied subject (see [8]), its solutions are not directly applicable to PAD Routing. First of all, the vast majority of the PDPTW research is concentrated on improving PDPTW performance in off-line environments, where these PDPTW algorithms assume all of the requests are known *a priori*. Furthermore, existing solutions do not make use of historical data such as recently visited nodes, and may schedule frequent visits to nearby nodes, resulting in longer delays.

Second, the use of wireless communication contributes a major difference. In PAD Routing, nodes communicate wirelessly within each other’s transmission ranges. That means a courier node does not need to travel to the ‘exact’ position of a node to service its pickup or delivery request, but must only be within transmission range. In contrast, in VRP the vehicle must physically be at the locations of the sender or receiver. In essence, VRP has a transmission range of zero. Clearly, PAD

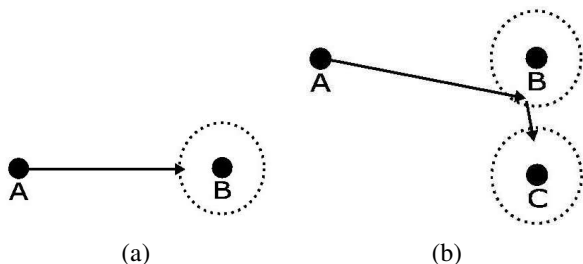


Fig. 2. Requests are not known *a priori* in an on-line scenario: (a) B requests service, best route is the straight route from A to B and (b) C requests service, moving on a diagonal produces a shorter route

Routing will exploit the benefit of a non-zero transmission range, and save time and energy by never going all the way to a request's exact location. However, this flexibility complicates the route calculation performed by PAD Routing.

For example, in Figure 2(a), the best tour from A is to move directly to and just in range of node B . However, after a request from node C has been added to the tour, an overall shorter route can be computed if the vehicle moves slightly toward node C even before it has serviced node B , as shown in Figure 2(b). Additionally, due to the overlapping coverage between vehicle routes and other issues, the exact movements may be optimized further once the full routes are constructed.

Furthermore, the PAD Routing problem and the CTP problem also differ in several aspects which make PAD Routing more complicated. First, as stated earlier, nodes only need to be within wireless communication range. In contrast, in traditional CTP, a courier has to move to the exact locations of nodes which must be visited (as denoted by set X). PAD Routing should exploit the benefit of this non-zero transmission range to save time and energy, but doing this complicates route calculations. For instance, in CTP, before any algorithm is executed to find a solution, all of the Z^1 requests can generally be 'statically' assigned to an X request which, when serviced, will result in the Z being covered. Since in CTP the vehicles always go exactly to the positions of X requests, the set of Z 's that will be covered when a particular X is visited is constant. In PAD Routing, vehicles have a range of positions they can be at when servicing X requests, and the choice of positions greatly affects which nearby Z 's are covered. This increases the calculations needed at each request.

Second, in CTP, Z 's are *only* covered when an X they are within range of is visited. For instance, consider supplies being distributed in a rural area, where the villages are X 's and the people are Z 's. The supply trucks can only cover a person by dropping off supplies at a village that the person is in range of, so that the people can easily travel to pick them up. For various reasons, trucks do not dispense supplies while they are traveling down the road between villages. However, in PAD Routing, wireless communication *can* do this, since transmissions can be made at any time. Thus, any Z 's within

¹ Z and X denote nodes that must be covered and visited, respectively.

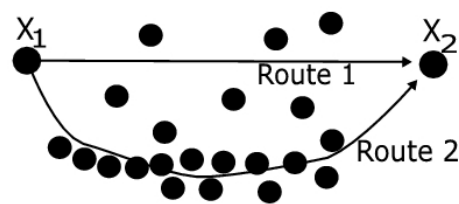


Fig. 3. Direct route between X nodes (X_1 and X_2) is not always optimal

range of *any portion of the physical route of a vehicle* can be covered, even if they are very far from any X location. This adds further complexity to the calculations:

Fuzzy Destinations. Since the exact next destination is not immediately known, the exact route also cannot be immediately determined. The distance, then, between Z 's and the route of a vehicle must potentially be recalculated several times.

Duplicate Coverage. In CTP, it is easy to determine whether a particular Z is covered by multiple nearby X 's, and the number of X 's serviced can be decreased to simplify and improve the solution. Clearly, overlap in a solution indicates potentially wasted vehicle movement. In PAD Routing, this cannot be checked as easily since, again, the exact position of a vehicle relative to an X 's position may vary. Additionally, since in PAD Routing a vehicle may cover a Z while in transit, the possibility of overlap also exists when a vehicle crosses over the route of either itself or another vehicle. Thus, each segment of a route must be checked for overlap with other routes or visited X 's.

Finally, in some situations in PAD Routing, some Z 's may be very far from the nearest X location. In CTP, this is also possible, however such Z 's are *never* covered, due to CTP's constraint that covering can only be done when the vehicle is at an X location. In PAD Routing, such Z 's can be covered if they are within range of a vehicle's route. The traditional, direct route, represented by Route 1 in Figure 3, may not always be best. By adjusting the route and instead using the non-direct Route 2, significantly more Z 's could be covered. Therefore, many possible routes of travel and associated Z 's covered must be considered when determining the next X request to service. Additional methods, such as creating artificial X locations near a dense group of Z 's or allowing vehicles to take non-direct routes to their next location, may be needed to provide adequate coverage of such Z 's.

These fundamental differences preclude the use of existing VRP and CTP algorithms, necessitating the development of new and efficient routing mechanisms that will exploit wireless communication and mobility to deliver messages among nodes in a delay tolerant mobile network.

III. THE PAD ROUTING PROBLEM

In this section, we describe the model, terminology, and assumptions used for the definition of the PAD Routing

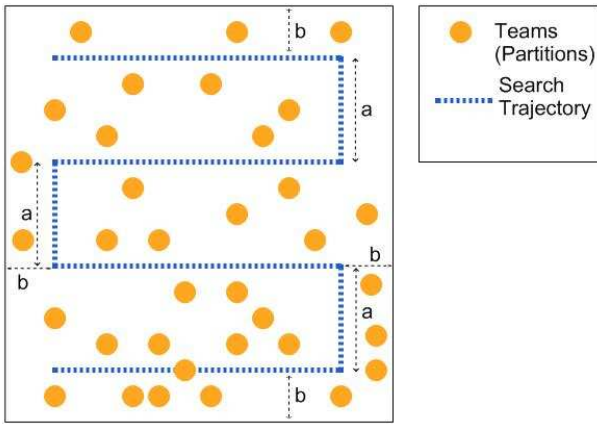


Fig. 4. Discovering Teams

problem.

A. Model, Terminology, and Assumptions

In our model, regular nodes are grouped (partitioned) into teams. A team consists of a (possibly empty) set of regular nodes and one team leader (TL). In addition to these nodes, there exist a set of courier vehicles (or nodes) that are less constrained in terms of battery energy, processing capacity, bandwidth, *etc.* We assume that every node has a short-range radio, and courier vehicles and TLs have long-range radios. Short-range communication is employed within a team, and between a TL and a nearby courier. Long-range communication is employed between a TL and a courier, and among couriers for the purposes of location announcements, service requests, *etc.* Intra-team communication is performed by a traditional ad hoc routing protocol using short-range radio. When a regular node s needs to send a message to a destination regular node d belonging to a different team, s first forwards the message to its TL s' . Then, s' requests pickup and delivery service from a courier vehicle c via long-range radio. When s' and c are within short-range communication distance, s' forwards the message to c via the short-range radio. Next, c delivers the message to the destination TL d' when c and d' are within short-range communication range. Finally, d' forwards the message to the destination node d . We assume that courier vehicles have limited message cache capacity, and each message is associated with a deadline. Furthermore, there is a system-wide fixed cache timeout period for messages. Messages are discarded if their deadline or cache timeout has expired.

We assume that courier vehicles know the TL of each regular node. This can be done by running an initial search algorithm as illustrated in Figure 4. A courier vehicle “sweeps” the entire terrain, row by row, following the search trajectory. In Figure 4, a denotes the distance between the rows and b represents the search path’s distance to the terrain boundaries. As long as $a \leq R/2$ and $b \leq R\sqrt{2}/2$, the entire terrain will be covered, where R is the long distance communication range. The courier vehicle may “ask” the TLs encountered about

their team membership². Note that due to node mobility, some teams may not be discovered using the scheme when $a = R/2$ and $b = R\sqrt{2}/2$. Picking more conservative a and b values can remedy the problem. If there are multiple couriers, the terrain can be divided among couriers and each courier can discover teams and team members in its region. Later, couriers can meet and exchange team information. Details on this alternatives are beyond the scope of this paper.

Another important question is how does a source node know which team a destination node belongs to. This information may be hard-coded in TLs and regular nodes. Alternatively, we can also employ the following mechanism. A node will first try to send a message using a conventional ad hoc routing protocol. If the message is delivered successfully, both the source s and the destination d reside in the same partition. Otherwise, source s forwards the message to its TL s' , and s' is responsible for delivering the message. TL s' can keep track of a dynamic list l of nodes in its partition. TL s' can lookup l to determine if the destination node exists within its own partition. If not, TL s' will request a courier vehicle to forward the message. Details on the above mechanism is beyond the scope of this paper.

We assume that there is no physical depot, and message transport requests are made only between regular nodes. We also assume that nodes have synchronized clocks. Since we are dealing with larger delay time intervals, nodes do not have to have very tight synchronization, and this assumption can be relaxed by incorporating the clock drift in the computations.

B. Problem Definition

The goal of PAD Routing is reduce the transport delay and to increase the pickup and delivery ratio in the entire network. Given a set of n nodes (excluding the courier nodes), we would like to minimize

$$D = \sum_{1 \leq i, j \leq n} d_{i,j} \quad (1)$$

where $d_{i,j}$ is the average delay for transporting messages from node i to node j . With respect to the delivery ratio, the measurement of delivery ratio is differently defined. Typically, delivery ratio is defined as the number of successful deliveries divided by the total number of delivery requests, or the total number of messages successfully delivered divided by the total number of messages need to be delivered. However, given a PAD Routing scenario where nodes make (the last) requests until time t' , at what (future) time t should we stop measuring the delivery ratio? This is due to the fact that it may take some time for PAD Routing to plan the tour, and to pickup and deliver the message. If $t \equiv t'$, there will be pending requests which will not be picked up and/or delivered. If $t \gg t'$, the delivery ratio could be 100%, assuming that there is no occurrence of message deadline or cache timeouts. To facilitate the performance evaluation, we introduce the metric

²Although this method allows the courier vehicle to cover the entire terrain in a short amount of time, it may be undesirable due to excessive energy consumption. In that case, after learning the location of TLs, the courier may visit each TL and learn the group members using short range communication.

completion time, which is the time required for an algorithm to complete the delivery of ‘eligible’ messages. We say that a PAD Routing algorithm achieves its completion time when either all the messages are delivered, or when some messages are delivered and for the remaining ones either the deadline or cache timeout has occurred. In our simulation study, we present both the completion time and the delivery ratio³.

It can be proved, using the technique described in [5], that the PAD Routing problem is NP-complete even with one courier. Therefore, our goal is to design heuristic solutions that will increase the delivery ratio and reduce the average delay, while avoiding excessive energy consumption⁴.

IV. PAD ROUTING ALGORITHMS

We describe the PAD Routing algorithms using a dynamic scenario with unicast traffic and a fixed number of courier vehicles. In this scenario, PAD Routing is transparent to regular nodes. In essence, a PAD routing algorithm has to address the following issues.

- **Courier assignment.** Each request should be associated with a courier, *i.e.*, a particular courier must be responsible for each request.
- **Location management.** Couriers need to keep track of the locations of TLs and other couriers. Location management helps couriers to make better routing decisions and can reduce the energy consumption and increase the accuracy of communication among long-range capable nodes.
- **Route (tour) planning.** Courier nodes have to decide the order in which the TLs would be visited. Note that if the list of TLs to be visited is fixed and known, then the problem becomes a TSP-like problem rather than a VRP/PDPTW-like one.
- **Route (tour) re-computation frequency.** Since the system is dynamic, courier nodes need to decide the frequency in which the re-computation of routes is performed.
- **Territory division.** We need to define the area that a courier needs to “serve.” A courier can serve the entire area or it could only pickup and/or deliver messages to/from a particular region. For example, in Figure 6, the terrain is divided into 6 regions and each courier can serve the entire region or may serve only a particular region.

³From a technical standpoint, there is no way for courier nodes, or any other node, to know that the completion time has passed, since it is impossible to know the future requests. However, as the designer of the simulation, since we know t' and the number of requests until t' , we can compute the completion time, and determine whether a simulation has arrived at its completion time. This metric is only used to compare algorithms and is not a part of the algorithms. Furthermore, notice that even if an algorithm runs until the completion time, the delivery ratio may not be 100% due to deadlines and/or cache timeouts.

⁴While energy efficiency is not the main focus of this study, our algorithms try to conserve energy as much as possible. For example, REQUEST packages (to be described in Section IV-A) are designed to be short, and the frequency of sending those packets is low to save energy.

A. The Framework

Unless otherwise specified, the PAD routing algorithms use the following framework. If there is no request to be served, the courier vehicles go to their *center* positions and wait for requests. If there is only one courier vehicle available, the *center* position is the center of the entire terrain. Otherwise, the terrain is divided into regions (to be described later) and each courier waits in the centroid of its corresponding region. If the terrain size of a courier’s region exceeds its long-range coverage, the courier may need to periodically sweep its region for new requests.

A request consists of a source/destination pair and a deadline. TLs periodically broadcast a set of requests for their respective teams. TLs will buffer requests until they are forwarded to a courier vehicle. Messages will be purged from the cache of a TL, if a deadline has expired before the message is forwarded to a courier. Similarly, courier vehicles will purge expired messages from their cache. To achieve more efficient location management and to save communication resources, location update messages are piggy-backed onto request/confirmation messages.

PAD Routing algorithms use the following control packets: HEARTBEAT, REQUEST, CONFIRMATION, and SRA. Long-range-capable nodes (TLs and couriers) will keep track of the locations of other long-range-capable nodes. To do so, courier nodes and TLs periodically send HEARTBEAT packets. A HEARTBEAT packet is a 5-tuple: $\langle nodeType, nodeId, seqNo, x, y \rangle$. *nodeType* specifies whether a node sending the packet is a TL or a courier, *nodeId* is the node ID of the sender, *seqNo* is a sequence number, and (x, y) is the location of the node. *seqNo* is used to detect and discard old HEARTBEAT packets. Long-range-capable nodes keep a courier list l_c and a TL list l_{TL} . Once a new HEARTBEAT packet h is received, its *nodeType* is checked. If the sender node is a courier, then l_c is checked to see if it contains information about the sender. If not, the new entry is recorded into l_c . Otherwise, the *seqNo* field of h is checked to see if h contains newer information than the one in l_c . If so, l_c is updated, else the h is discarded. Similar operations take place for l_{TL} if the sender is a TL node.

TL nodes will cache the inter-partition traffic requests from their teams in an outgoing cache and periodically send them to a courier. TLs also have an incoming cache for the distribution of messages received via courier nodes from other teams. Caches contain 8-tuple items: $\langle itemID, msg, srcId, destId, size, scheduledCourier, requestTime, deadline \rangle$. *itemID* is an automatically generated sequence number to uniquely identify each item, *msg* is the message, *srcId* and *destId* are source and destination node ID numbers, *size* is the size of the packet, *scheduledCourier* is the courier responsible for pickup, *requestTime* is the time of the request, and *deadline* is latest time by which the message has to be delivered. The *scheduledCourier* field is initially set to INVALID_ADDRESS and will be set to the responsible courier once a courier sent a CONFIRMATION packet (to be

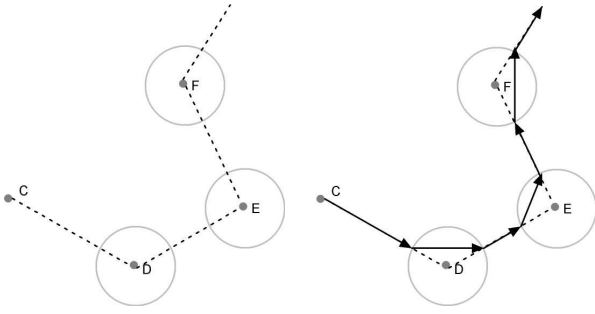


Fig. 5. Wireless communication feature: (a) nodes to be visited, and (b) the actual route taken.

discussed below). When a node wants to send an outgoing message to another team, the message is sent to the TL. TL will receive the message and insert it into its outgoing cache. The TL will periodically send the requests using a REQUEST packet. A REQUEST packet consists of all the information in outgoing cache, except the *msg*, of course. A REQUEST packet has a fixed length, in our implementations 10 requests, and if there are more cached requests, additional REQUEST packets have to be generated. We employ a *nearest courier* heuristic for courier assignment, *i.e.*, TL nodes will request service from nearest couriers in their l_c lists. When a courier node receives a REQUEST packet, it adds the new requests to its *task list*, and sends a CONFIRMATION packet. A CONFIRMATION packet specifies the *itemID* fields of the received requests. A CONFIRMATION packet has a fixed length (10 *itemIDs*) similar to REQUEST packets. Upon receiving a CONFIRMATION packet, a TL node checks its outgoing cache and marks the corresponding *scheduledCourier* fields.

Courier nodes use their task lists to plan their tours. This list is basically a list of source/destination TL pairs⁵ and the corresponding *requestTime/deadline*. Given such a list, a courier node generates an initial route using a nearest neighbor approach. Later, the route is improved using the 2-OPT [10] and the 2H-OPT delay heuristics. The route is improved until no improvement is possible and a visit order of TLs is finalized. Given a route $v_1, \dots, v_i, A, B, \dots, C, D, v_j, \dots, v_n$, the 2-OPT heuristic produces the new route $v_1, \dots, v_i, A, C, \dots, B, D, v_j, \dots, v_n$ if the new estimated delay is lower. Similarly, the 2H-OPT heuristic takes a route $v_1, \dots, v_j, X, v_{j'}, \dots, v_n$ and replaces it with $v_1, \dots, v_k, X, v_{k'}, \dots, v_n$ if the new route gives a lower expected delay. After the route computation, the courier node visits the TL nodes following the computed tour. After the tour is completed, the route computation is performed again.

Our algorithms take into account the fact that nodes communicate wirelessly. Hence, instead of following the route depicted in Figure 5(a), courier nodes follow the route in Figure 5(b). With wireless communication, nodes do not have

⁵In the implementation, courier nodes assign their *nodeId* to source TL field once the item is picked up. By doing so, courier nodes distinguish between the items that require pickup and the ones that were already picked up.

to be exactly co-located to communicate. Couriers compute the entry point (toward the transmission range of the next TL to be visited) and the exit point (away from the transmission range), and move from the entry point to the exit point. Furthermore, courier nodes compute the time they need to spend inside the transmission range of a node and slow their speed down, if necessary. This helps couriers to exchange all messages with each TL in the minimum time possible. To do so, we perform the following computations:

Let R be the data transmission rate (in bps), $\alpha \in (0, 1]$ the channel availability, h the transfer overhead (in bits) per packet (due to RTS/CTS/ACK/DIFS *etc.*), s_i the size of a sent packet i (in bits), r_j the size of a received packet j (in bits), *speed* the maximum speed of the courier, and d the distance traveled inside the transmission range of the node, then

$$t = \frac{h \times (n_s + n_r) + \sum_{i=1}^{n_s} s_i + \sum_{i=1}^{n_r} r_i}{R \times \alpha} \quad (2)$$

gives⁶ the time required to transfer $n_s + n_r$ packets, where n_s is the number of sent packets and n_r is the number of received packets. We compute the speed of the courier inside the transmission range as follows.

$$v = \begin{cases} \text{speed}, & \text{if } t \times \text{speed} \leq d; \\ d/t, & \text{otherwise.} \end{cases} \quad (3)$$

Given a route v_1, \dots, v_n , a task list *tasks*, courier's speed *speed*, the current position v_0 , we compute the estimated delay as follows.

ESTIMATEDDELAY($n, v_1, \dots, v_n, \text{tasks}, \text{speed}, v_0$)

```

1  delay = 0
2  t = 0
3  b0 = v0 // previous exit point
4  for i = 1 to n
5    do Compute the entry(ai)/exit(bi) points for vi
6       Compute the distance di between bi-1 and ai
7       Compute the distance di' between ai and bi
8       Compute the courier's speed vi using
9       the formula in Eq 3
10      ti = di/speed
11      ti' = di'/vi
12      t = t + ti + ti'
13      for each task in tasks
14        do if task's destination TL is vi
15           then delay = delay + t
16
17  return delay
```

In addition to the HEARTBEAT packets (long range location announcements), courier nodes also periodically send SRA (short range announcement) packets. A SRA packet contains the *nodeId* of the courier (*courierId*), courier location (x, y), and a sequence number (*seqNo*). Upon hearing a SRA message, a TL initiates an exchange sequence. Then the courier

⁶Clearly, this is a simplified formula which gives a reasonable estimation.

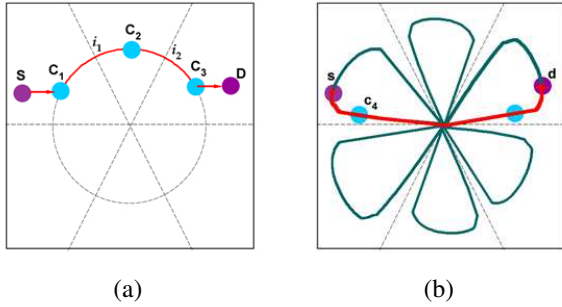


Fig. 6. Cooperative case: (a) couriers meet at the intersection of their territories i_1, i_2 etc., and (b) couriers meet at the center.

node delivers the packets sent to the TL, and picks up the items⁷ that the TL wants to send to other teams.

B. The Algorithms

We have developed the following two groups of algorithms that use different techniques for route planning and territory division.

- **Simple:** In this group, which include FIFO (First-In-First-Out), NN (Nearest Neighborhood), and DMF (Dynamic Message Ferrying), team leaders select the nearest courier and request service from that courier. A courier serves the entire terrain. Couriers are autonomous in the sense that each courier computes its own route. Furthermore, each message is carried exactly by one courier, hence there is no cooperation/coordination among the couriers. Since there is no dedicated courier for each node/request, any courier can pick up the messages. In FIFO, requests are served in the order of arrival. When deciding the next TL node to visit, a courier checks its task list, finds the TL that submitted the earliest request, and picks up the item and delivers it. To improve the performance, a courier will pick up any messages that are ready to be picked up from the source and destination TLs. NN is similar to FIFO, but instead of using the request times, the courier serves the closest node first. Hence, while planning the next TL to visit, the courier checks its task list, computes the distances to all TLs, chooses the closest TL that needs a pickup or delivery service. The courier repeats this operation after each pickup or delivery service. In DMF, nodes do not issue explicit requests. Instead, courier nodes *calculate* fixed routes and follow these routes. When a TL is within the short-range communication distance of a courier, the TL receives messages sent to its team and forwards new messages to the courier node. The default route is computed using a nearest neighbor approach. The route is improved by using the 2-OPT and the 2H-OPT heuristics.

⁷Normally, only messages that are confirmed by the courier is picked up. However, sometimes a courier is dedicated for a particular region. In such a case, the courier picks up all messages from TLs that are located inside its region. Similarly, some algorithms will pick up all the messages regardless of confirmation/region. Such distinctions will be made clear inside the algorithm descriptions.

- **Region Based:** We design two region based algorithms: WEDGE1 (uncooperative) and WEDGE2 (cooperative). In both algorithms, the terrain is divided into regions, as in Figure 6, and each courier is responsible for one area.⁸ In WEDGE1, a courier only picks up items from the region assigned to it, but can deliver anywhere. Hence in WEDGE1, at each round a courier visits all the nodes inside its region, picks up⁹ items, and delivers the messages outside the region. This means that the courier makes route planning twice at each round. First, a route is generated to visit all the TLs inside the region. Then, once the courier finishes following this route, task list is checked and a new route is constructed to deliver all the messages picked up on the previous route. Then the courier will deliver all the messages. However, no new messages will be picked up by the courier outside its region. Hence, WEDGE1 is an autonomous algorithm. WEDGE2, on the other hand, is a cooperative algorithm and requires coordination. In WEDGE2, couriers pick up and deliver messages only from their own region. If a packet needs to be sent to another region, the courier will send such messages to another courier and the new courier will be responsible for delivering the message. Therefore, the courier nodes require synchronization. Several synchronization schemes are possible and two of them are depicted in Figure 6. In Figure 6(a), each pair of couriers sharing a border have a rendezvous point and this mechanism requires synchronization for each such pair. A simpler approach, implemented in this paper, requires the courier nodes to meet around the center of the terrain, as depicted in Figure 6(b). The courier nodes meet at the center, exchange messages and serve their region, and after a time period (in our simulation, 450s), they meet again and exchange messages. This results in better load balancing compared to other algorithms.

V. SIMULATION RESULTS

We have conducted simulation experiments using QualNet to study the characteristics and evaluate the performance of the proposed PAD Routing algorithms. Unless specified otherwise, 12 random networks were generated with 40 teams, each of which contains 5 regular nodes. The teams are randomly distributed over a square terrain of size 5000m×5000m. For short-range communication, each node is equipped with a radio transceiver capable of transmitting signals up to approximately 250 meters over a 2Mbps wireless channel, using the two-ray path loss model without fading. We use IEEE 802.11DCF as the MAC layer protocol, and IP for the network layer protocol. For long-range communication, we chose not to simulate any MAC protocol, but rather assume a model where there is no loss or delay. We assumed that long-range capable nodes can send HEARTBEAT packets to any other

⁸As mentioned earlier the center positions are calculated as the centroid of these regions. In fact, simple algorithms (FIFO, NN, DMF) also use the same territory division strategy to compute the center positions.

⁹and possibly delivers some of them inside the same region

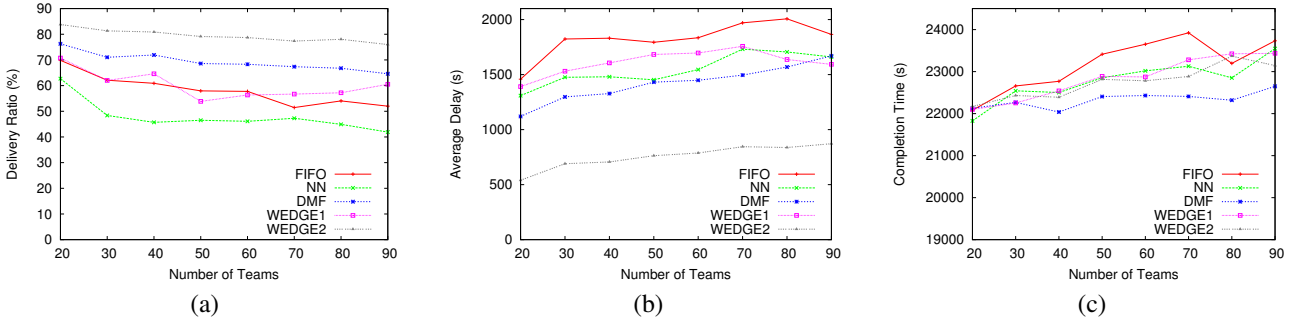


Fig. 7. The impact of network size (number of teams) on: (a) delivery ratio. (b) average delay. (c) completion time.

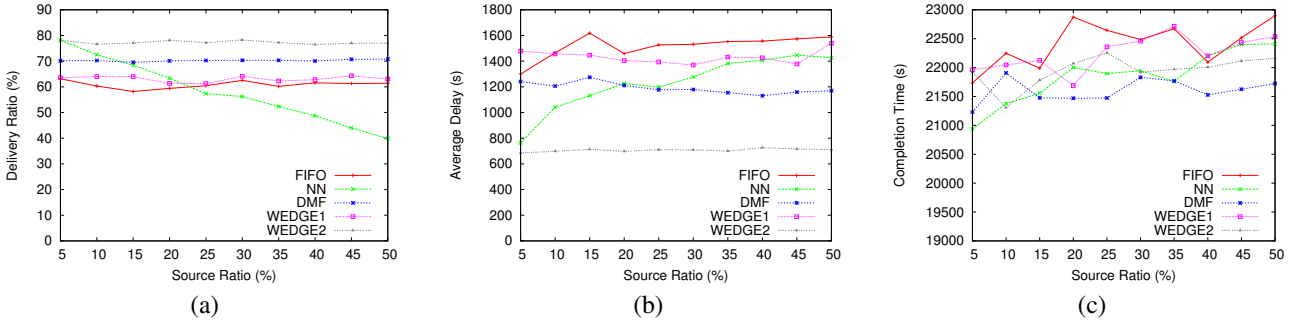


Fig. 8. The impact of traffic load. Specifically, source percentage vs.: (a) delivery ratio. (b) average delay. (c) completion time.

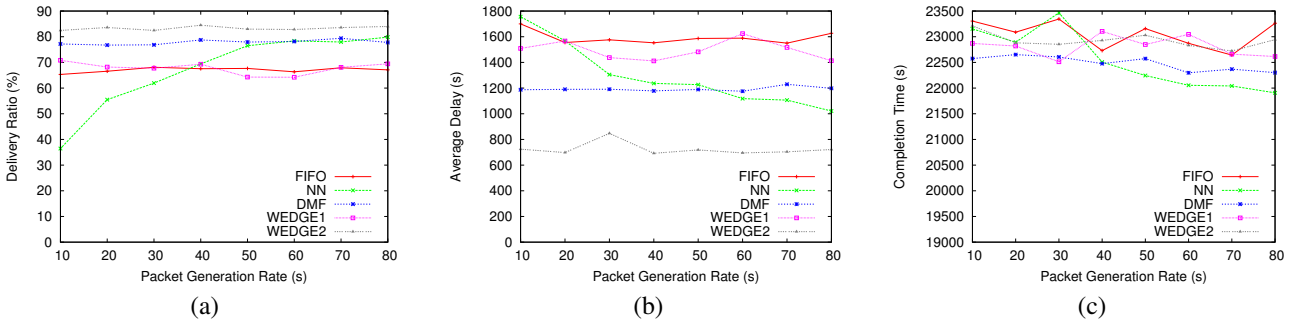


Fig. 9. The impact of traffic load. Specifically, traffic rate vs.: (a) delivery ratio. (b) average delay. (c) completion time.

node, without any range limitations (*i.e.*, $5000 \times \sqrt{2} = 7071\text{m}$). Since long-range communication is seldom used for control messages (requests/confirmations and location updates) and not used at all for data transmissions, MAC layer contention is not a significant problem and our assumption does not affect simulation results. Every t seconds, p percent of nodes are selected as the source nodes. Source nodes generate a message to be delivered to a destination node belonging to a different team. Each packet is associated with a deadline value which is picked from a minimum-maximum deadline interval randomly. All the experiments are performed with 4 courier nodes. These couriers are initially placed at their *center locations*, as defined above, within the terrain. Unless otherwise is stated, we used the following default values: $p = 30\%$, $t = 25\text{s}$, request interval = 500s, location update interval = 400s, cache timeout = 7200s, for deadline generation minimum deadline = 3600s (1 hour) and maximum deadline = 7200s

(2 hours) and packet length = 1K. We used QualNet's group mobility model [11] with minimum and maximum group speeds of 1m/s and 2m/s, respectively. Each group occupies a $300\text{m} \times 300\text{m}$ area.¹⁰ We performed five experiments over a 20000s time period¹¹ and the results are presented in Figure 7 to Figure 11. Notice that graphs related to completion time is not meant to compare the performance of the algorithms. Rather, we consider completion time graphs as supplementing the delivery ratio and delay graphs. For example, sometimes one algorithm gives slightly better delay results, but at the expense of increasing the completion time, *e.g.*, comparison of the performances of FIFO and WEDGE1 algorithms in

¹⁰We chose a small region for groups to ensure connectivity among group members.

¹¹This is the time interval over which traffic is generated. In order to obtain the 'completion time' values, the simulations were executed longer, *i.e.*, we incremented simulation times by the maximum deadline value.

Figure 7(a) and Figure 7(c) suggests that although FIFO rarely outperforms WEDGE1, FIFO achieves this gain at the expense of higher completion time.

In Figure 7, we study the impact of the number of teams. The number of teams are varied from 20 to 90, and the impact of this change on delivery ratio, delay and completion time is presented. Clearly, WEDGE2 algorithm outperforms other algorithms in both delivery ratio and delay. In fact, it outperforms others in every scenario. This shows the importance of cooperation among courier nodes. Cooperation helps both to improve the overall delivery ratio and also to reduce delay. Figure 7(b) shows that other algorithms experience 2 to 3 times longer delays. Contrary to our expectation, DMF algorithm performs better than WEDGE1. In DMF, courier nodes visit teams in different order, and also any courier node can pick up messages from any team leader. This reduces the difference between packet's generation and pickup times and improves the performance. On the other hand, in WEDGE1 a dedicated courier has to pick up items and it sometimes takes considerable amount of time before the courier finishes its tour outside its region and returns back for pickups. FIFO algorithm achieves a delivery ratio close to, and sometimes better than, WEDGE1 but at the expense of a higher completion time and a longer delay. NN has the worst performance. The main reason for such poor performance is that NN algorithm tends to spend more time especially if the area is crowded, and newer pickup requests cause courier nodes to stay in the same area and couriers miss the deadlines.

Figure 8 and Figure 9 illustrate the impact of traffic load on the performance. In Figure 8, the packet generation rate is fixed, but the ratio of nodes selected as source nodes is increased from 5% upto 50%. In Figure 9, the source ratio is fixed and the packet generation frequency is increased from 10s upto 80s. The negative impact of fast message generation on the NN algorithm is clear on both figures. However, the traffic load does not affect other algorithms significantly, because as the load increases the couriers can pick up and deliver more items. Reduced load does not also improve the performance much. Even if the number of packets to/from a TL may be reduced, the courier node still needs to visit the TL and hence the performance remains pretty steady.

Figure 10 shows the impact of the deadlines. As we mentioned earlier, each packet is associated with a deadline. This value is randomly selected from a minimum-maximum deadline interval. In the simulation study, we increased the minimum deadline from 1000s up to 5000s. Clearly, strict deadlines increase the packet drops and reduce the delivery ratio. This pattern is common in all algorithms. As depicted in Figure 8 and Figure 9, the average delay is pretty steady, since no matter what the deadline is, all the TLs that required service should be visited. Similar to other experiments, WEDGE2 gives the best results, while NN gives the worst. Again, DMF is the second best and other two algorithms (WEDGE1 and FIFO) perform very closely.

Another parameter that affects the performance is the maximum speed of the courier vehicles. As we mentioned earlier,

speed of the vehicles inside the transmission range of a TL (if a message exchange takes place) is determined, adaptively, by Eq. 2 and Eq. 3. However, maximum speed is a parameter in Eq. 3, which also significantly affects the inter-partition travel time. The maximum speed is increased from 12 m/s up to 18 m/s and the results are presented in Figure 11. These graphs show that increasing speed improves the delivery ratio and reduces the delay significantly. The impact of courier speed is more obvious on average delay since increased speed reduces the tour times directly. Although increased speed improves the performance, other factors such as short deadlines limit the impact of courier speed on the delivery ratio. The relative order of the algorithms is similar to the previous experiments.

VI. RELATED WORK

Vahdat *et al.* [3] propose to harness mobility by exchanging data when encountering new nodes. This is a passive approach, where nodes solely rely on their inherent movement. However, this random behavior leads to low delivery rates and large delays. To enhance the performance, nodes may propagate multiple copies of the same message, which results in many redundant copies in the network and wastes scarce resources. Li *et al.* [1] propose an optimal proactive approach, which describes how trajectory changes can be used to transmit messages in disconnected ad hoc networks. To do so, they propose to compute the trajectory for sending a message from host A to host B by recruiting intermediate hosts to help. However, as pointed out in [5], it is difficult to extend this algorithm to efficiently support multiple simultaneous transmissions.

Inspired by its real life analogy, the "Message Ferrying" (MF) scheme [5] was proposed which implements the SCF routing paradigm. In MF, a set of mobile nodes called message ferries take responsibility for carrying and transporting messages between disconnected nodes. Message ferries move around the deployed area according to known routes and communicate with other nodes they meet. By using ferries as mobile relays, nodes can communicate asynchronously with other nodes that they are otherwise disconnected from. The main idea of the MF scheme is to introduce non-randomness in the movement of nodes and exploit such non-randomness to help data delivery. In [6], the same authors described the use of multiple ferries in networks with stationary nodes, and focus on the design of ferry routes. They show that performance scales well with the number of ferries in terms of throughput, delay and resource requirements in both ferries and nodes. They also studied the trade off between the incurred cost of multiple ferries and the improved performance. Although similar to MF, PAD Routing is different from both [5] and [6] in several ways. The MF scheme in [5] works with only one ferry, while PAD Routing employs all available courier nodes. Furthermore, in the MF scheme the ferry follows a default path. However, [5] does not show how to define default paths for multiple couriers. PAD Routing instead uses an adaptive approach. Rather than following a default path, routes are computed according to TL requests and TL

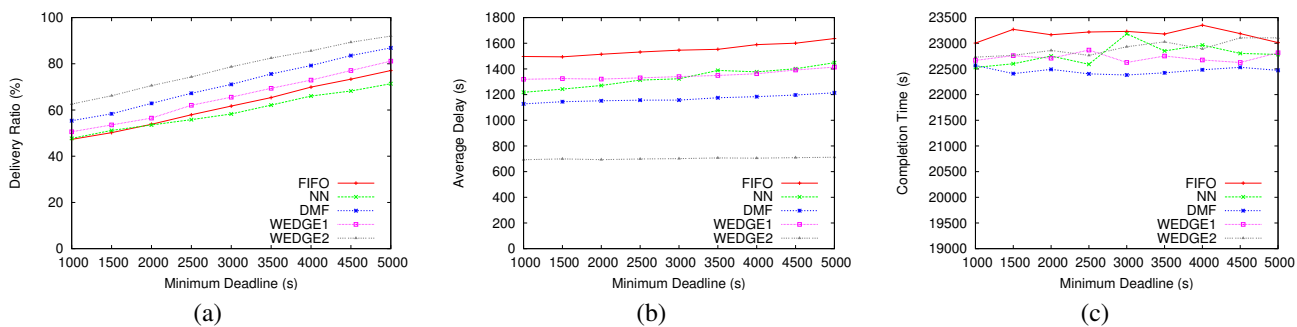


Fig. 10. The impact of minimum deadline on : (a) delivery ratio. (b) average delay. (c) completion time.

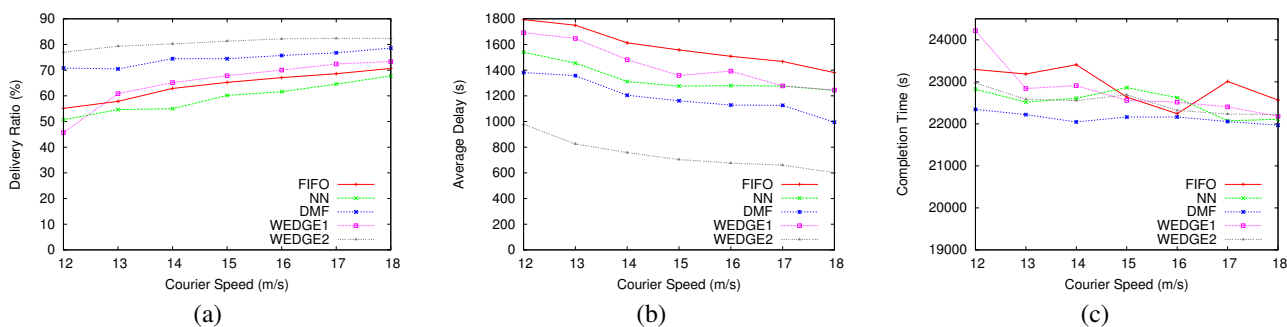


Fig. 11. The impact of courier speed on : (a) delivery ratio. (b) average delay. (c) completion time.

locations maintained at each courier, hence the system is more dynamic and adaptive. Although [6] works with multiple ferries, the authors assume that the nodes are stationary and traffic pattern is known beforehand. PAD Routing on the other hand investigates dynamic scenarios, which do not make any assumptions about the future traffic, and regular nodes and TLs may be mobile.

VII. CONCLUSION

In contrast to the previous work of VRP/PDPTW and Message Ferrying, the paper describes autonomous and dynamic mechanisms, together termed PAD Routing, to transport data for delay tolerant mobile networks, where efficient message pickup and delivery mechanisms are designed to fulfill dynamic message transportation requests in a continuous fashion with respect to time in the presence of network partitions. PAD Routing algorithms may employ multiple couriers, and reduce average delay while achieving high delivery ratio. Simulation results compare the performance of different PAD Routing algorithms and demonstrate their efficiency. Work is in progress to extend PAD Routing with the multicast capability, and to investigate the impact of high mobility.

REFERENCES

- [1] Q. Li and D. Rus, "Communication in disconnected ad-hoc networks using message relay," *Journal of Parallel and Distributed Computing*, vol. 63, no. 1, pp. 75–86, January 2003. [Online]. Available: <http://www.cs.dartmouth.edu/liqun/paper/jpdc.ps>
- [2] R. Shah, S. Roy, S. Jain, and W. Brunette., "Data mules: Modeling a three-tier architecture," 2003. [Online]. Available: citeseer.ist.psu.edu/694681.html
- [3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," 2000. [Online]. Available: citeseer.ist.psu.edu/vahdat00epidemic.html
- [4] C.-C. Shen, S. Rajagopalan, G. Borcar, and C. Jaikao, "A flexible routing architecture for ad hoc space networks," *Elsevier Computer Networks Journal, Special Issue on Networking for the Earth Sciences*, vol. 46, no. 3, October 2004.
- [5] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. ACM Press, 2004, pp. 187–198.
- [6] —, "Controlling the mobility of multiple data transport ferries in a delay-tolerant network," in *Proceedings of IEEE INFOCOM 2005*, 2005.
- [7] V. Maniezzo, R. Baldacci, M. Boschetti, and M. Zamboni, "Scatter search methods for the covering tour problem," in *Adaptive Memory and Evolution: Tabu Search and Scatter Search*, C. Rego and B. Alidaee, Eds. Kluwer Academic Publishers, 2003.
- [8] M. Savelsbergh and M. Sol, "The general pickup and delivery problem," *Transportation Sci.*, vol. 29, no. 1, pp. 17–29, 1995. [Online]. Available: <http://www.isye.gatech.edu/mwps/publications/ts29.pdf>
- [9] M. Hachicha, M. J. Hodgson, G. Laporte, and F. Semet, "Heuristics for the multi-vehicle covering tour problem," *Comput. Oper. Res.*, vol. 27, no. 1, pp. 29–42, 2000.
- [10] S. Lin, "Computer solutions of the traveling salesman problem," *Bell System Technical Journal*, vol. 44, pp. 2245–2269, 1965.
- [11] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*. New York, NY, USA: ACM Press, 1999, pp. 53–60.