DTN Routing with Probabilistic Trajectory Prediction

Ionut Cardei[1], Cong Liu, Jie Wu, and Quan Yuan

Department of Computer Science and Engineering,
Florida Atlantic University, Boca Raton, FL 33431, USA
icardei@cse.fau.edu

Abstract. Many real-world DTN application involve vehicles that do not have a purely random mobility pattern. In most cases nodes follow a predefined trajectory in space that may deviate from the norm due to environment factors or random events. In this paper we propose a DTN routing scheme for applications where the node trajectory and the contact schedule can be predicted probabilistically. We describe a technique for contact estimation for mobile nodes that uses a Time Homogeneous Semi Markov model. With this method a node computes contact profiles describing the probabilities of contacts per time unit, and uses them to select the next hop such that the delivery ratio is improved. We develop the Trajectory Prediction DTN Routing algorithm and we analyze its performance with simulations.

Keywords: DTN; delay tolerant networking; routing protocols; message scheduling; Markov process

1 Introduction

Regular MANET routing protocols work on the assumption that there exists at least one path between endpoints, and will,therefore, fail route discovery. Proactive routing protocols (e.g. DSDV, OLSR) will also fail to converge due to rapid topology changes or lack of stable connectivity.

To mitigate these issues a new class of Delay Tolerant Networks (DTN) has been defined in [4]. Connectivity in DTNs relies on nodes physically delivering messages between disconnected partitions, similar to how the postal service delivers packages. This store-carry-forward approach for end-to-end message delivery exploits the increased user mobility instead of being hindered by it. If a message reaches a node that has no link to the next hop towards the destination, it will be buffered until a contact occurs with the next hop. In this way a message can be delivered from a source to its destination even when there never exists an instantaneous path between the two endpoints. In some DTNs the waiting time between successive contacts may be very large (hours for interplanetary networks) and variable, so applications must be designed to tolerate long delivery latencies.

The intermittent nature of end-to-end connectivity and the variable delay require a new approach for routing. The local decision of selecting the next hop for a message depends on available information on current connectivity and on future opportunities to establish links with other nodes (contacts). Hence, routing also may involve scheduling transmissions for future contacts in addition to selecting the next hop. To optimize the network performance, such as delivery ratio or latency, DTN routing must select the right contact to transmit the message. If a contact is not available when a message is received from the upper layer, the DTN transport layer will buffer it until a proper contact begins and transmission time becomes available, or until the message expires and is dropped.

In this paper we propose a Trajectory Prediction DTN Routing scheme (TPDR) for applications where the node trajectory and the contact schedule can be predicted probabilistically. Most sensing applications with mobile nodes define trajectories that are far from random. Node movement is typically controlled such that sensing quality of service is optimized to increase application lifetime, coverage, or to reduce latency. DTN message routing can be employed to extend the operational range of vehicles beyond communication range and to mitigate intermittent connectivity. For instance, Autonomous Underwater Vehicles (AUVs) in a littoral surveillance application follow a scan pattern that provides full sonar coverage of the sea bed (Figure 1). In these networks it is possible to predict nodes trajectories and contact schedules. Due to events affecting the mission plan and environmental factors (such as ocean currents), contact schedules can not be known with 100% accuracy. We describe a technique for trajectory prediction and contact estimation for mobile nodes that uses a Time Homogeneous Semi Markov model. With this method we compute contact profiles for pairs of nodes that describe the probability of a contact per time unit. We develop the Trajectory Prediction DTN Routing algorithm and we analyze its performance with simulations.

This paper continues in Section 2 with a presentation of the application and network models, and the routing architecture. Section 3 describes related work in routing for Delay and Disruption Tolerant networks. Section 4 presents results from performance evaluations using simulations. The paper concludes in Section 5 with some comments.

2 DTN Routing with Trajectory Prediction

This section begins with a description of the application and the network models considered in this paper for the DTN routing algorithm with contact prediction.

The applications addressed in this paper consider small and medium size networks where nodes are mobile, with trajectories that can be predicted for a certain time horizon and where the communication topology is mostly disconnected. Such a scenario is not thoroughly addressed by the DTN research community. More specifically, the proposed routing solution applies to applications where nodes are location aware and also have available enough information on the mobility pattern, such as the velocity vector and mission status, that they can approximate when contacts begin and end. Contact prediction of this nature is possible in applications where node operation is coordinated either centrally (mass transit, trains) or autonomously (mobile sensing). In mobile sensing applications, such as subsea monitoring [13], sensor–actor networks [1, 11], position status and movement data is disseminated to the network. In mobile sensing applications location updates can be broadcasted with low data rate to reach the entire network, while mission payload data can be forwarded on a high speed wireless link with shorter range during contacts. When a broadcast channel is not available, status (position and velocity) updates is propagated throughout the network in the typical DTN store-and-carry fashion, although with considerably higher delays. For all practical reasons it is nor feasible, neither useful to predict contacts for the whole application duration. It is considerably cheaper to limit trajectory and contact predictions to a finite time horizon.

To support the contact prediction and the node state dissemination mechanisms it is necessary for all nodes to have their system clocks synchronized. In networks where the contact duration/link data rate fraction is large, precise node time synchronization is not required. Otherwise, if the contacts are very short, maybe due to high vehicle speed, or if the link data rate is very high, precise time synchronization is important to pinpoint contact begin times to avoid wasting unused transmission time. In most cases broadcast beacon protocols can serve for synchronization.

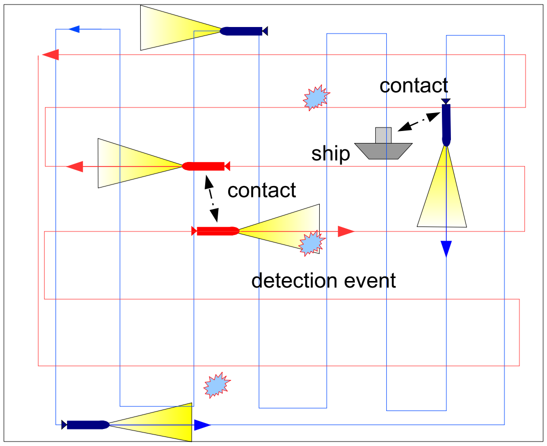


Fig. 1. DTN network scenario with autonomous underwater vehicles scanning an area and reporting to a controller ship.

An application that fits the above description involves underwater littoral monitoring with Autonomous Underwater Vehicles (AUV). As described in [12], the high costs of equipment and deployment make feasible applications with only a relatively small number of AUVs. Increased coverage is achieved by designing the vehicle trajectory to scan the target area. Characteristics of the underwater acoustic communications channel, such as long propagation delays, low data rates and short transmission range, limit the operational capability of AUVs. A store-and-carry delay tolerant approach is an economic solution to improve coverage at the cost of higher delays. The availability of an out-of band slow long-range acoustic channel used to broadcast vehicle status updates brings up the possibility of predicting nodes' trajectories and contact opportunities. A short range high data rate acoustic link forwards sensor data (e.g. sonar or video) during contacts.

Figure 1 shows a notional AUV scenario with vehicles executing repeatedly a closed monitoring path. To simplify the presentation, the application defines two trajectories, a horizontal combing pattern and a vertical one. Several vehicles can follow the same trajectory with a delay between them.

We assume that each node moves according to a predetermined pattern at a constant speed monitoring the environment. Periodically, onboard sensors generate images (from a camera or a sonar) that have to be forwarded towards the controller station on the ship. From time to time pattern recognition algorithms running on sensor data on each AUV will trigger a *detection event*. This event causes the AUV to autonomously interrupt its current trajectory and gather extra data before it continues on its original path. The vehicles report sonar measurement messages to the base station located on a non-moving ship using the acoustic high data rate link, following the DTN store-and-carry bundle forwarding approach if a connected contemporaneous multihop path is not available.

2.1 Network Model and Trajectory Prediction

The monitored area is overlayed on a $w \times w$ square grid. Grid coordinates x, y are numbered between 1 and w. Each cell is assigned a unique cell ID $s(x, y)$ given by the following mapping:

$$s(x, y) = (y - 1)w + x$$

This cell numbering is equivalent to assigning successive numbers to cells from left to right, top to bottom, beginning with the origin cell, (1,1). The set of all cells is denoted with S. The size of set S is $m = w^2$.

Time in this model is discretized. The trajectory followed by a vehicle u up to time unit t is a sequence of cells $S^i_{1,t} = \{s\}_{1..t} = \{(x, y)_{1..t}\}$. We describe this trajectory for an individual vehicle, as it evolves in time, as a Markov process. The set of Markov states is represented by the set of cells on the grid S.

As a vehicle enters a state (grid cell) i, it stays there for a time called *state holding time*, and then leaves to the next state j. The selection of j can be described by a transition probability matrix P^e, $P^e_{ij} = P(\text{transition from } i \text{ to } j)$. Let T_n be the time of the n^{th} transition, $n \geq 0$, and $T_0 = 0$. The state holding time for the n^{th} transition is $T_n - T_{n-1}$. The probability distribution of the state holding time depends on the states involved in the transitions. We believe it is not realistic to assume that the state holding times have the memoryless Markov property, with a geometric distribution.

We model the system with a Time Homogeneous Semi-Markov Process (TH-SMP). The state holding times can have arbitrary distributions – with the constraints that they are i.i.d. and do not change in time (time-homogeneous). The TH-SMP is defined by the tuple $\{(S_n, T_n)|n \geq 0\}$, where S_n is the n^{th} state reached. The TH-SMP *kernel* Q describes the process evolution in time:

$$Q_{ij}(t) = P(S_{n+1} = j,\, T_{n+1} - T_n \leq t \mid S_n = i) = P^e_{ij} H_{ij}(t) \tag{1}$$

We assumed that the selection of the next state (given by P^e) is independent of the state holding time distribution function when the model transitions from state i to state j (denoted with H_{ij}).

$$H_{ij}(t) = P(T_{n+1} - T_n \leq t \mid S_{n+1} = j,\ S_n = i).$$

The state holding time effectively depends on the vehicle speed that we assume is constant in our application. We also assume a vehicle is delayed in a state by a constant time d_{ev} when a detection event occurs. The probability of a detection event per time unit per state is a constant, p_{ev} .

A vehicle geometric path on the grid is defined by the matrix P^e that forms the embedded Markov chain of the TH-SMP, as we observe that
$P^e_{ij} = \lim_{t\to\infty} Q_{ij}(t) = P(S_{n+1} = j \mid S_n = i)$.

The state holding time irrespective of the next state is defined as $D_i(t) = P(T_{n+1} - T_n \leq t \mid S_n = i)$. This is the c.d.f. of the time it takes the vehicle to traverse a grid cell i regardless where it goes next. $D_i(t)$ can be computed as:

$$D_i(t) = \sum_{j=1}^{m} Q_{ij}(t) \tag{2}$$

To predict the future vehicle trajectory, we define the stochastic process $X = (X_t, t \in \mathbf{N})$, where $X_t \in S$ is the vehicle state at time t. The distribution of X_t is given by $\phi_{ij}(t) = P(X_t = j \mid X_0 = i)$. If we know that a vehicle is now in state i, after t time units in the future it will be in state j with probability $\phi_{ij}(t)$. As a special case $\phi_{ij}(0) = \delta_{ij}$, where δ is Kronecker's symbol.

To determine $\phi_{ij}(t)$ one can use the distributions for $D_i(t)$ and $Q_{ij}(t)$, which can be derived from P^e and $H_{ij}(t)$, both easy to determine from the application domain.

To determine $\phi_{ij}(t)$ we start with a special case when the process stays in state i between time 0 and t, with no transitions.

$$P(X_t = i|X_0 = i \text{ and } \mathrm{T_1 \geq t) = P(T_1 - T_0 \geq t \mid X_0 = i) = (1 - D_i(t))}.$$

If the process makes at least a transition between times 0 and t, conditioned on the time of the first transition (at time k) from i, and on the state l to which the process moves after state i, we obtain:

$$P(X_t = j \mid X_0 = i \text{ and at least one transition}) = \sum_{\mathrm{l=1}}^{\mathrm{m}} \sum_{\mathrm{k=1}}^{\mathrm{t-1}} \mathrm{\dot{Q}_{il}(k)}\phi_{\mathrm{lj}}(\mathrm{t-k}),$$

where $\dot{Q}_{il}(k) = \frac{dQ_{il}(k)}{dk} = Q_{il}(k) - Q_{il}(k-1)$ is the time derivative of Q. Putting it together, we obtain:

$$\phi_{ij}(t) = (1 - D_i(t))\delta_{ij} + \sum_{l=1}^{m} \sum_{k=1}^{t-1} \dot{Q}_{il}(k)\phi_{lj}(t-k) \tag{3}$$

ϕ can be calculated iteratively, as $\phi_{ij}(t)$ depends on probabilities $\phi_{lj}(t-k)$ computed in the previous steps.

This approach was inspired by the work in [7] that looked at the problem of predicting access point handoffs in WLANs. Our Markov model (state space, transitions) is defined differently and the prediction is extended to handle inter-node contacts.

2.2 Contact Prediction and the Forwarding Decision

The *contact profile* $C_{ab}(t) = P(a \leftrightarrow b \text{ contact at time } t)$ for two nodes a and b can be determined from the TH-SMP behavior given by ϕ and from the *neighborhood map* N_s. For each state $s \in S$, N_s is the set of states $z \in S$, s.t. a vehicle in state s can communicate with a vehicle in z. The neighborhood map can be computed based on underwater topography maps, channel characteristics, and on the technical properties of the communication device. For the simple disk model, with distance $d()$, $N_s = \{z \in S \mid d(s,z) \leq r\}$.

The contact profile at time t depends on having the two nodes a and b in each other's neighborhood, and it is expressed as:

$$C_{ab}(t) = \sum_{s_a \in S} P(X_t^a = s_a) \sum_{s_b \in N_{s_a}} P(X_t^b = s_b) = \sum_{s_a \in S} \phi^a_{i_a s_a}(t) \sum_{s_b \in N_{s_a}} \phi^b_{i_b s_b}(t) \tag{4}$$

i_a and i_b are the states of nodes a and b, respectively, found out most recently. Note that the contact prediction works with relative time. Different time offsets can be applied as parameters to $\phi(t)$. We note that the contact profile is not a proper pdf and is not normalized, as $\sum_t C_{ab}(t)$ may exceed 1.

Each message has a time-to-live (TTL) field. When the TTL expires, the message is dropped. A nodes buffers messages until a contact begins. Then, the TPDR protocol decides whether to forward a message intended for destination d in a greedy way.

Suppose at time t the current node a is in contact with a set of nodes $\{b_i\}$. Assume the current node a receives periodic updates with the current state s_{b_i} for all nodes b_i over the secondary channel. Node a removes from the set $\{b_i\}$ all nodes that have already received a message node a buffers and has to forward. This can be accomplished either by storing the message path in its header, by query, or by summary vector exchange, as in Epidemic Routing. After that node a computes the following:

– the contact profile for the destination node d: $C_{ad}(t)$, indicating the probabilities of direct contact and delivery to the destination node
– the contact profiles of nodes b_i with the destination d: $C_{b_i d}(t)$. These give the probabilities that the next hop (one of nodes b_i) will be in contact with the destination d in the future.

The prediction window for which the contact profiles are computed is limited by the remaining message time-to-live. Then the routing algorithm at node a takes the following steps:
1. pick the node $c \in \{a\} \cup \{b_i\}$ that maximizes the probability of contact with the destination: $c = \text{argmax}_{u \in \{a\} \cup \{b_i\}} \text{max}_{t=1..TTL}\, C_{ud}(t)$
2. if $c = a$, continue buffering the message. There is no forwarding.
3. else forward the message to next hop c since c has a higher probability of a contact with the destination than the current node a.

Discussion Routing cycles are avoided by preventing forwarding to nodes that have seen a message before. This routing algorithm has several simplifying assumptions. First, it assumes that during a contact all queued messages can be forwarded to the next hop. Then, it assumes infinite buffers and an ideal communications channel during a contact. With the affordability of memory capacity, the infinite buffer assumption is not out of line for most applications. The first assumption applies to cases where the message load is low, the data rate is very high, or the contact duration is very long. For the AUV application, the latter case is more realistic due to the reduced vehicle speed.

In terms of effective implementation, it is worthwhile to note that matrices $\phi_{ij}(t)$ that give the trajectory prediction, must be computed just once, at the beginning of the application runtime, provided the other involved distribution (state holding time H) and the state transition probabilities P^e do not change.

While the number of states (w^4) in the TH-SMP can be daunting, matrices ϕ, H, Q, and P^e are very sparse. For the area combing patterns in Figure 1, the fill factor for these matrices is about $\frac{1}{w^3}$. Sparse matrices support efficient storage and matrix arithmetic operations.

The Trajectory Prediction DTN Routing algorithm could be expanded in many directions. We investigate an extension for a shortest path routing algorithm that uses contact profiles. Similar work has been described in [5]. Our approach benefits from the ability of specifying arbitrary distribution functions for the state holding time $H_{ij}(t)$, which is more realistic. This also supports better the use of recorded historic data on trajectory traces.

3 Related Work

In this section we present a brief overview of DTN routing techniques relevant to our problem. For a comprehensive overview of DTN routing, the reader should consult [17].

One of the first thorough analysis of communications in networks with intermittent connectivity is done by Fall in [4]. Fall proposes a new delay/disruption tolerant architecture, later updated by Cerf et. al. in [3]. The DTN architecture defines a *bundle layer* operating above the transport layer, that offers end-to-end delivery service to applications. The bundle layer forms an overlay network used for transfer of message *bundles* with the option of hop-by-hop custody transfer (delivery responsibility) and optional delivery with end-to-end confirmation.

Routing has better performance when more information are available on the current state of the network topology and on its future evolution. At one end of the spectrum is *deterministic routing*, where the current topology is known and future changes can be predicted. With deterministic routing, message forwarding can be scheduled to optimize network performance and to reduce resource utilization using single-copy forwarding. In contrast, *stochastic routing* techniques assume node mobility is random or unknown and therefore must rely on multi-copy forwarding to increase the end-to-end delivery probability.

Deterministic DTN routing techniques are based on formulating models for time-dependent graphs and finding a space-time shortest path in DTNs by converting the routing problem to classic graph theory. These techniques are appropriate for scenarios with predictable topology (e.g. space networks) or where node mobility is tightly controlled, such as unmanned air vehicles (UAVs) and Autonomous Underwater Vehicles . A major problem facing deterministic routing protocols remains the distribution of network state and mobility profiles under sporadic connectivity, long delays, and sparse resources.

Jain et al. present in [6] a routing framework that takes advantage of increasing levels of information on topology, queue state and traffic demand. Four *knowledge oracles* are defined. The *contacts summary oracle* provides time-invariant aggregate or summary statistics on inter-node contacts, such as average waiting time until a next contact. The *contact oracle* provides full information for all contacts, such as start time and duration, enough to build a time-varying contact multigraph. The *queuing oracle* answers for the instantaneous queue state and current waiting times at all nodes. The *traffic demand oracle* gives information on any present and future messages injected in the network. The authors adapt the Dijkstra shortest path algorithm to run in a time-varying multigraph where the edge cost functions is determined with the available oracles. The edge cost function is the total estimated edge delay, consisting of the sum of the signal propagation delay (computed from node location information), the contact waiting time (given by the contact oracle), and the transmission queuing delay (available from the queuing oracle). A *complete knowledge* centralized linear program optimization is presented that uses the traffic demand oracle to more accurately characterize transmission times. This serves as a benchmark for performance evaluations. Due to the sparse connectivity in DTNs, estimating current queue state across the network and implementing edge capacity reservation are difficult. Simulation results point out that in scenarios with limited resources (buffer space and edge capacity) the benefits from using the additional knowledge from the queuing oracle are not significant.

When network state is too uncertain, *stochastic routing* techniques forward messages randomly hop-by-hop with the expectation of *eventual delivery*. In between, there are routing mechanisms that may predict contacts using prior state, or that adjust the trajectory of mobile nodes to serve as message ferries. Stochastic routing techniques rely on replicating messages and controlled flooding for improving delivery rate, trading off resource utilization against improved routing performance in absence of accurate current and future network state.

Passive routing techniques do not interfere with node mission, do not change the node trajectory and react to a changing topology. Passive routing techniques rely in general on flooding multiple copies of the same message with the objective of *eventual delivery* [14–16]. These protocols trade off delivery performance against resource utilization. By sending multiple copies on different contact paths (such as in epidemic routing [16]), the delivery probability increases and the delay drops at the cost of additional buffer occupancy during message ferrying and higher link capacity usage during contacts. This approach is appropriate when nothing or very little is known about mobility patterns.

Some passive stochastic routing protocols use *delivery estimation* to determine a per contact probabilistic metric for successful delivery based on recorded history of prior contacts ([8, 10]. These protocols are useful when contacts cannot be accurately predicted and when nodes follow non-random trajectories.

Active routing techniques ([2, 9, 18]) rely on controlling the trajectory of some ferry nodes to pick up messages and ferry them in preparation for a contact with the destination node. Active routing techniques provide lower delays with the additional cost of increased protocol and system complexity. They also rely on the availability of mobile message ferries that could be reassigned from their original mission.

4 Performance Evaluation

In this section we present the simulation performance evaluation for the TPDR algorithm described in this paper. The results are compared with other routing protocols for DTNs, Direct Delivery and Epidemic Routing [16]. With Direct Delivery, the source node buffers a message until it is delivered during a contact with the destination node. The simulated scenario involves 4 to 10 AUVs, plus one ship, deployed in a 1 km square area divided in a 20 × 20 grid. The dynamics of the AUV vertical and horizontal scan patterns are modeled by the transition probability matrix P^e and the state holding time distribution matrix $H_{ij}(t)$. The state holding time is 1 with probability $1 - p_{ev}$ and 3 with probability $p_{ev} \in \{0.01, 0.05, 0.1\}$. This models the random detection events that trigger a delay for taking additional sonar measurements. These detection events also generate new messages intended for the ship node. The message TTL is varied between 40 and 100 time units. 100 is also the upper bound for trajectory prediction.

We simulated this topology with a packet-level simulator written in Matlab. The simulation assumed an ideal channel with no delay, as we wanted to focus mostly on the effectiveness of the prediction element on the overall routing performance metrics — message end-to-end delay and delivery ratio. Half of AUVs use the horizontal scan and the others use the vertical pattern.

The quality of the contact prediction is of great interest. In Figure 2 we overlap the computed predicted contact profile between AUV_2 and the ship node ($C_{21}(t)$) with the actual contact trace between these two nodes that was captured during the simulation. The contact prediction accurately indicates future contacts with higher probability values. Tables 1 a) and b) show the variation of the delay and the delivery ratio depending on the AUV count: 6, 8, and 10. One can notice that the delivery ratio comes within 10% of Epidemic Routing, which is a multi-copy routing approach. The delay is not better, as the routing decision picks the next hop with the highest probability of contact with the destination. We also note the better delay for the Direct Delivery protocol. This happens because most messages for this protocol get dropped due to expired TTL from lack of connectivity, and they do not contribute negatively to lower the average delay. In the next experiment shown in Figure 3, the AUVs and the ship use

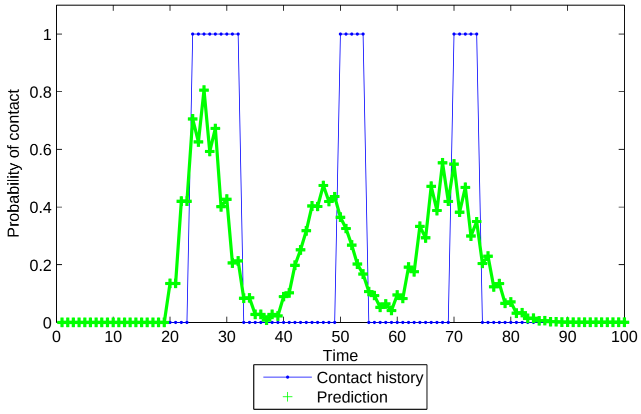


Fig. 2. Predicted contact profile and actual contact trace between AUV_2 and the ship.

	6	8	10
Direct	18.77	18.74	16.63
Epidemic	21.02	19.31	17.48
TPDR	21.02	20.47	18.89

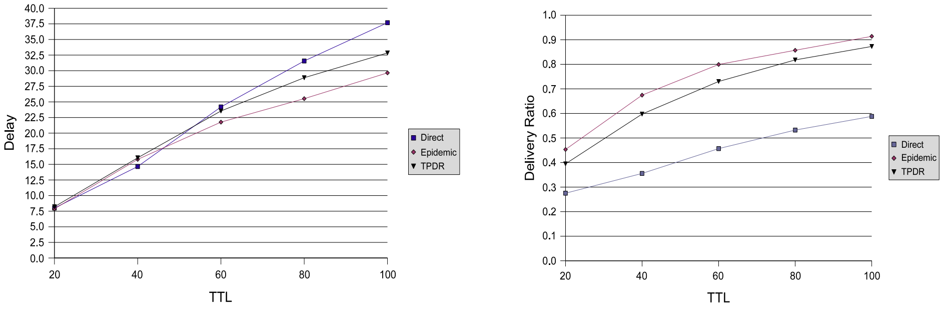
a) Message delay

	6	8	10
Direct	0.37	0.4	0.43
Epidemic	0.77	0.75	0.74
TPDR	0.67	0.68	0.68

b) Delivery ratio

Table 1. End-to-end message delay and delivery ratio for 6, 8, and 10 AUVs

message TTL that changes from 40 to 100. Figure 3 a) illustrates the variation of the message delay depending on TTL. As expected, a larger TTL increases the average delay, but also improves the delivery ratio. TPDR performs well for delay, but excels for delivery ratio. For higher TTLs, TPDR closes on the Epidemic Routing's delivery ratio, which is maximal, reaching above 80%.



a) Delay

b) Delivery ratio

Fig. 3. Protocol performance depending on the message TTL field.

5 Conclusions

In this paper we presented an approach for probabilistic DTN routing that uses prediction of the trajectory of mobile nodes. The prediction mechanism models the geographic rectangular grid as a Time-Homogeneous Markov Process, where the states map to grid cells, and the transitions between cells reproduce vehicle movement. This approach does not depend on the memoryless property for correct state estimation and supports arbitrary state holding time probability distributions that are i.i.d. and time-homogeneous (i.e. do not change over time). Based on the state transition prediction, we developed contact estimation using contact profiles. The profile gives the probability for a contact between two nodes in a particular time unit, regardless of the grid location. A DTN forwarding algorithm was developed, that selects the next hop based on the highest contact probability. Simulations have demonstrated the effectiveness of this algorithm.

References

1. Ian F. Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks, Elsevier*, 2:351–367, 2004.
2. B. Burns, O. Brock, and B.N. Levine. Mv routing and capacity building in disruption tolerant networks. In *IEEE INFOCOM 2005*, volume 1, pages 398–408 vol. 1, 2005.

3. V. Cerf et. al. Delay tolerant network architecture. draft-irtf-dtnrg-arch-05.txt, March 2006.
4. Kevin Fall. A delay-tolerant network architecture for challenged internets. Technical report, Intel Research, 2003. IRB-TR-03-003.
5. Jean-Marc François and Guy Leduc. Predictable disruption tolerant networks and delivery guarantees. *CoRR*, abs/cs/0612034, 2006.
6. Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, volume 34, pages 145–158. ACM Press, October 2004.
7. Jong-Kwon Lee and Jennifer C. Hou. Modeling steady-state and transient behaviors of user mobility: formulation, analysis, and application. In *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 85–96, New York, NY, USA, 2006. ACM.
8. Jeremie Leguay, Timur Friedman, and Vania Conan. Evaluating mobility pattern space routing for dtns. In *INFOCOM 2006*, 2006.
9. Qun Li and Daniela Rus. Communication in disconnected ad hoc networks using message relay. *J. Parallel Distrib. Comput.*, 63(1):75–86, 2003.
10. Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. In *SIGMOBILE: Mobile Computing Communications Review*, volume 7, pages 19–20. ACM Press, July 2003.
11. Tommaso Melodia, Dario Pompili, and Ian F. Akyildiz. A Communication Architecture for Mobile Wireless Sensor and Actor Networks. In *Proceedings of IEEE SECON 2006*, Reston, VA, September 2006.
12. Jim Partan, Jim Kurose, and Brian Neil Levine. A survey of practical issues in underwater networks. In *In Proc. ACM International Workshop on UnderWater Networks (WUWNet)*, pages 17–24, Sept. 2006.
13. Joseph Rice. Seaweb acoustic communication and navigation networks. In *The International Conference of Underwater Acoustic Measurements: Technologies and Results*, Heraklion, Crete, Greece, July 2005.
14. R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications*, pages 30–41, May 2003.
15. Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *The ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM Press.
16. A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Duke University, Apr. 2000. CS-200006.
17. Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges. In *IEEE Communications Surveys*, volume 8, 2006.
18. W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 2, pages 1407–1418 vol. 2, 2005.