Ad Hoc Networks 11 (2013) 1136–1149

Contents lists available at SciVerse ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Generic prediction assisted single-copy routing in underwater delay tolerant sensor networks

Zheng Guo [a,*,1], Bing Wang [b], Jun-Hong Cui [b]

[a] Xi'an University of Posts and Telecommunications, Xi'an, China
[b] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA

ARTICLE INFO

Article history:
Received 7 February 2012
Received in revised form 19 November 2012
Accepted 19 November 2012
Available online 3 January 2013

Keywords:
Delay tolerant networks
Underwater sensor networks
Single-copy
Routing
Algorithm

ABSTRACT

One challenge in delay tolerant networks (DTNs) is efficient routing, as the lack of contemporaneous end-to-end paths makes conventional routing schemes inapplicable. Although many DTN routing protocols have been proposed, they often have two limitations: many protocols are not mobility cognizant, so they only suit specific mobility models and become inefficient when the environment changes; some protocols employ multi-copy replication to accommodate mobility diversity for increased delivery probability or reduced delay, but they usually do not perform well in resource constrained networks. Due to the unique characteristics of underwater sensor networks (UWSNs), efficient DTN routing becomes even more challenging. In this paper, we propose a generic prediction assisted single-copy routing (PASR) scheme that can be instantiated for different mobility models. PASR first collects a short-duration trace with network connectivity information and employs an effective off-line greedy algorithm to characterize the underlying network mobility patterns, depict the features of best routing paths and provide guidance on how to use historical information. Then it instantiates prediction assisted single-copy online routing protocols based on the guidance. As a result, the instantiated protocols are energy efficient and cognizant of the underlying mobility patterns. We demonstrate the advantages of PASR in underwater sensor networks with various mobility models.

1. Introduction

Many routing protocols have been proposed to deal with the lack of contemporaneous end-to-end paths in delay tolerant networks (DTNs) [1–8]. These protocols, however, have the following limitations. First, many protocols are designed for specific mobility models [9–12]. For instance, the protocols in [13,14,7,15,16] are for networks in social environments; the protocols in [17,18] focus on random way point and random walk mobility models; and the protocol in [11] is for networks with pre-determined node trajectories. Although some other protocols are designed for general mobility models, they are not mobility model cognizant [19]. Since the underlying mobility dominates the contact and inter-contact pattern [20], these mobility incognizant protocols can have superior performance for one model while much degraded performance for another model [19]. Another drawback of most existing routing protocols is that they use multi-copy replication, which allows multiple replicas of a packet to exist in a network simultaneously. These protocols establish several virtual spatial temporal routes (either using flooding [4,19,21] or controlled flooding [15,22]) to increase delivery probability and decease end-to-end delay. On the other hand, they exhaust network resources (such as bandwidth, storage and power) much more quickly than single-copy routing strategies. Thus, multi-copy routing

* Corresponding author.
E-mail addresses: guozheng@engr.uconn.edu (Z. Guo), bing@engr.uconn.edu (B. Wang), jcui@engr.uconn.edu (J.-H. Cui).
[1] He was with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA. He is now with Xi'an University of Posts and Telecommunications, Xi'an, China.

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2012.11.012

schemes always lead to poor performance in resource stringent networks.

Underwater sensor networks (UWSNs), an area that has attracted significant attention from both academia and industry [23–27], can be treated as DTNs [28] due to node mobility and sparse deployment. Compared to other DTNs, UWSNs are extremely resource stringent since acoustic communication, the most practical communication method for UWSNs, has very limited bandwidth and very high power consumption. Furthermore, the mobility pattern in an UWSN can vary dramatically over time depending on the environment. These two characteristics render existing DTN routing protocols, especially multi-copy replication schemes, unsuitable in UWSNs. Therefore, an efficient single-copy routing scheme, which can also be self-adaptive to the varying mobility, is desirable.

In this paper, we propose a generic scheme, *prediction assisted single-copy routing* (PASR), for UWSNs to achieve minimum delivery delay at low energy consumption. PASR can be instantiated to efficient single-copy routing protocols under different mobility models. PASR consists of two phases, *learning phase* and *routing phase*. In the learning phase, PASR collects a short-duration trace with network connectivity information and employs an effective off-line greedy algorithm to characterize the underlying mobility patterns, depict the common features of best routing paths and provide guidance on how to use historical information. In the routing phase, it instantiates a prediction assisted single-copy online routing protocol based on the guidance. Our main contributions are: (1) we propose a short-duration trace-based greedy algorithm, named *aggressive chronological projected graph* (ACPG) in the learning phase and analyze the computational complexity; (2) we design a generic scheme on instantiating heuristic prediction assisted single-copy routing protocols based on the guidance from ACPG in the routing phase; and (3) we evaluate this generic scheme in UWSNs with three different mobility patterns through comprehensive simulation. Our simulation results show that ACPG indeed captures the relationship between historical information and best routing paths under different mobility patterns and provides effective guidance to instantiate heuristic single-copy routing protocols, that achieve close to optimal results and outperform other existing schemes. A preliminary version of this paper appeared in [29].

The rest of this paper is organized as follows. We first discuss related work in Section 2. We then introduce the network model and convert it to an extended space time graph in Section 3. Afterwards, we present the greedy algorithm ACPG and compare it with an optimal algorithm in Section 4. Section 5 describes the generic scheme PASR and how to instantiate prediction assisted single-copy routing protocols in UWSNs for different mobility models. Section 6 presents performance evaluation. Finally, Section 7 concludes the paper and proposes future research directions.

2. Related work

Since many types of networks (e.g., planet networks [30], sensor networks [24], village networks [31], vehicle networks [11]) can be treated as delay tolerant networks, much effort has been put on the challenging routing problem and many schemes have been proposed. Depending on whether the complete information of the networks is available or not, DTN routing can be categorized as *deterministic routing* and *heuristic routing*.

2.1. Deterministic routing

Deterministic routing schemes are to optimize a certain performance metric (e.g., shortest average delay, highest delivery ratio, minimum energy consumption, longest network lifetime and so on) when complete information is available. The complete information may include the location of nodes, the contacts between nodes, the power or storage status of nodes, the traffic demands and so on. Based on the complete information, deterministic routing schemes can achieve the optimal results in respect to a certain performance metric. Merugu et al. built a space time graph to select routing paths using dynamic programming and shortest path algorithm in [32]. Jain et al. formulated a linear programming problem upon the availability of all knowledge oracles in [5].

The rigorous requirement on the complete information of networks makes all algorithms above not practical in real networks, since even missing partial information can significantly increase the complexity [5]. Different from above schemes, we propose a practical prediction assisted scheme. In this scheme, we apply an off-line deterministic algorithm on a short-duration trace to guide the design of online heuristic routing schemes.

2.2. Heuristic routing

In most networks, it is impossible to obtain complete information in advance, thus only heuristic routing is suitable. According to how many replicas of a packet can exist in the network simultaneously, we classify existing heuristic routing schemes into two categories: *multi-copy routing* and *single-copy routing*.

2.2.1. Multi-copy routing

Multi-copy routing means that a node can replicate a packet on multiple relay nodes and expect any of them can reach the destination quickly. Epidemic [4] was a representative multi-copy routing scheme. To minimize the end-to-end delay, Epidemic replicated a packet to every node in the network. However, this flooding scheme consumed too many resources and made itself infeasible in harsh network environments. To avoid unconstrained resource consumption, many other multi-copy routing schemes that limit the number of copies have been proposed. These schemes forward packets according to some criteria, i.e. utility function, forwarding probability and so on. Harras et al. proposed several controlled flooding schemes in [21], such as basic probabilistic, time-to-live, kill time and passive cure. Spyropoulos et al. presented spray and wait [19], in which a certain number of copies of a packet were replicated to the first encountered nodes. This scheme only strictly controlled the total number of copies, but did not choose the relay nodes wisely, causing wasteful replication. Later, Spyropoulos et al. and Xue et al. improved the performance of spray and wait with better distribution schemes in [33,34] separately. In [35], Burgess et al. suggested a multi-copy routing scheme, called MaxProp, using priorities based on the path likelihood. Lindgren et al. proposed PROPHET to limit the number of copies [15], in which an intermediate node only forwarded a packet to the neighbors that have higher probabilities to reach the packet's destination in a short time. Balasubramanian et al. presented an intentional DTN routing protocol, Rapid, to optimize a specific routing metric [36]. Rapid treated DTN routing as a resource allocation problem by translating the targeting metric to per-packet utility and determining the packets replication in the network. Sandulescu et al. exploited the context of mobile nodes to estimate the size of a contact window and proposed OR-WAR to make better forwarding decisions [37]. In addition, Jones et al. utilized the contact history to find routes with minimum estimated expected delay [38]. Wu et al. proposed a scheme that forwarded packets to relays with increasing utility to increase reliability [22]. The scheme by Cardei et al. made routing decisions based on the probabilistic trajectory prediction [39]. Liu et al. presented optimal probabilistic forwarding (OPF) to maximize the expected delivery rate using forwarding thresholds as functions of remaining hop-count and residual time-to-live in [6]. Guo et al. adaptively classified packets with different priorities and assigned corresponding replicas in [28].

These multi-copy routing schemes replicate packets using some criteria, which are tailored for specific network mobilities and the performance can degrade significantly when the underlying mobility changes. So they require to know the underlying mobility in advance and cannot directly apply to other network scenarios. Moreover, multi-copy routing schemes consume too many resources (e.g., power, buffer and bandwidth) and are not suitable for resource stringent networks. In this paper, we propose a generic mobility cognizant single-copy routing scheme. This scheme does not require the underlying mobility in advance, while it applies a greedy algorithm on a short-duration trace collected to characterize the mobility, and guides to instantiate a corresponding heuristic single-copy scheme for this network. Therefore, our scheme is applicable to any network scenario, especially when the underlying mobility is unknown or not constant. In addition, our single-copy routing schemes are energy efficient.

2.2.2. Single-copy routing

In the literature, only few studies focus on single-copy routing. In connected networks, routing schemes take only single copy on a single path because it is easy to determine the *best* routing criteria. However, in DTNs, it is challenging, if not impossible, to decide which relay is the *best* candidate during each contact opportunity. Chen et al. only focused on the cross-cluster disconnection in cluster-based networks, which was only applicable to situations where the nodes inside a cluster were well connected and clusters were occasionally connected [40]. Musolesi et al. proposed Context-Aware Routing (CAR) to send messages to host with the highest delivery probability, which was predicted from available context (the set of attributes that described the aspects of the system and could be used to optimize the process of message delivery) [41]. Spyropoulos et al. discussed a number of basic single-copy routing protocols in [18]. Yuan et al. presented predict and relay (PER) to forward a packet based on the prediction of the probability distribution of future contact times [7]. They assumed that nodes could only move around a set of landmarks following a time-homogeneous semi-Markov model.

As a single-copy routing scheme, the forwarding criteria become even more strict and closely related to the underlying mobility. Thus the aforementioned routing schemes can only be used in special networks, because one criterion may be not meaningful, or even not available, in other networks. In this paper, we solve this problem by proposing a general prediction assisted scheme, which characterizes the network mobility and gives guidance on which information can be used for prediction and which forwarding criteria can be used for forwarding packets.

3. Preliminaries

3.1. Network model

We consider a data collection underwater sensor network, which consists of M layers. Multiple underwater sensors are deployed in each layer, and can passively move with water currents in the horizontal plane and vibrate slightly in the vertical direction. This kind of deployment can be achieved by simple buoyancy control of underwater sensors at certain depths [24]. Fig. 1 shows a simple example of such a network with three layers, where the dashed lines indicate the instantaneous connectivity. In this network, for simplicity, we assume that one data sink is anchored in the middle of the water surface.

Since underwater sensors float with currents (we assume passive mobility in the target UWSN model), their movements are driven by the movement of water currents and are tractable to some extent [42]. We adopt the kinematic model [43] to describe the mobility. The current field is assumed to be a combination of a tidal and a residual current fields. The tidal field is a spatially uniform oscillating current in one direction and the residual current field is assumed to be an infinite sequence of clockwise and counter-clockwise rotating eddies. The movement is

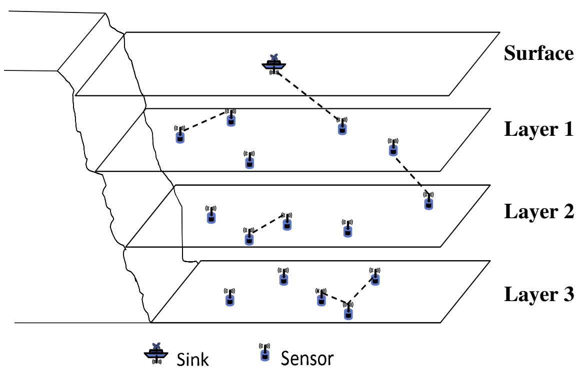


Fig. 1. A simple example of a three-layers underwater sensor network.

constrained in the horizontal plane and independent of the depth. The mobility model can be approximated as

$$\begin{cases} V_x = k_1\lambda v\sin(k_2x)\cos(k_3y) + k_1\lambda\cos(2k_1t) + k_4 \\ V_y = \lambda v\cos(k_2x)\sin(k_3y) + k_5 \end{cases} \tag{1}$$

where V_x and V_y are the instantaneous velocities on the X and Y axes respectively; k_i ($i = 1, \ldots, 5$), λ and v are variables related to the environment, such as tides and bathometry. We vary these parameters in Section 6 to instantiate three different mobilities and demonstrate that the proposed PASR can adapt to the corresponding underlying mobility.

Further we assume the network operates in a slotted manner, each slot of duration T. Sensors in the lowest layer generate packets to be transmitted to the sink using nodes in the middle layers as relays. All sensors use store-and-forward mechanism; a packet received or generated in a slot can be forwarded from the next slot. At the beginning of each slot, each sensor broadcasts a short *HELLO* message to its neighbors to declare its existence and exchange information. Each sensor is equipped with a buffer that can accommodate W packets and a battery that can transmit P packets. Sensors work in a half-duplex mode (i.e. they cannot transmit and receive simultaneously), and transmit or receive data at the rate of λ packets per second. The objective is to deliver packets to the sink with minimum delay at low energy consumption.

3.2. Extended space time graph

Since the connectivity changes over time, we can represent the evolving network in both spatial domain and temporal domain as a directed extended space time graph [44]. We assume the network contains M layers and N nodes in each layer (excluding the sink on the surface). We also model a super source v_s that generates packets and distributes them to the corresponding sources without delay, and a super sink v_d (the sink on the surface) to collect all packets. The slots are indexed as $0, 1, 2, \ldots$ Thus, we can construct a directed extended space time graph $G(V, E)$.

- V: the set of nodes, including v_s, v_d and node instances at different slots $v_{ij}^t, i = 1, 2, \ldots, M, j = 1, 2, \ldots, N, t = 0, 1, 2, \ldots$
- E: the set of edges, including *holding edge set* E_h and *forwarding edge set* E_f, i.e. $E = E_h \cup E_f$.
  - E_h: an edge $\left(v_{ij}^t, v_{ij}^{t+1}\right) \in E_h$ denotes that a packet can be held on node v_{ij} during the slot t; an edge $\left(v_s, v_{ij}^t\right) \in E_h$ denotes packets generation.
  - E_f: an edge $\left(v_{ij}^t, v_{kl}^{t+1}\right) \in E_f$ iff node v_{ij} can transmit to v_{kl} ($i \neq k$ or $j \neq l$) during slot t.
- $C_{(u,\ v)}$: the capacity of edge (u, v), defined as:

$$C_{(u,v)} = \begin{cases} +\infty, & (u, v) \in E_h \\ \tau\lambda, & (u, v) \in E_f \end{cases} \tag{2}$$

  where λ is the transmission rate, and τ is the contact duration between u and v in that slot, $0 \leqslant \tau \leqslant T$.
- C_v: the capacity of node v, which is defined as the available buffer space.

To completely describe the special constraints of the network, we introduce the following notations:

- $d(u, v)$: the cost of edge (u, v). Since our objective is to minimize the delay, the cost is defined as

$$d(u, v) = \begin{cases} 0, & u = v_s \\ T, & o.w. \end{cases} \tag{3}$$

which indicates a delay (cost) of T when a packet traverses an edge.

- P: the residual battery capacity of a node, indicating how many transmissions a node can still afford.
- λT: the transceiver capacity, i.e. the maximum number of packets a node can transmit and receive in one slot.
- q_v^i: the traffic demand, i.e. the number of packets that node v generates during the ith slot.

Fig. 2 gives an example extended space time graph for a three-layer network. A vertical vertex set at slot t (as circled) represents the node instances at the end of that slot. Vertices in the set are arranged according to the layer and node sequence, not related to the geographic locations. The horizontal direction represents the evolving network in temporal domain. The dashed lines are holding edges and the solid lines are forwarding edges. v_s and v_d represent the super source and super sink, respectively.

4. Deterministic routing algorithm

The directed extended space time graph $G(V, E)$ can be constructed based on a small trace, regarding the node connectivity, recorded during a short period. After that, we can use deterministic algorithm, such as *integer linear programming* (ILP), to find the optimal routing solution with minimum delay. However, ILP has high computational complexity and only provides the lower bound of achievable minimum delay. Hence, we develop a greedy algorithm, *aggressive chronological projected graph* (ACPG). ACPG not only produces routing results that are close to the minimum delay, but also characterizes the underlying mobility pattern and depicts the common features of routes.

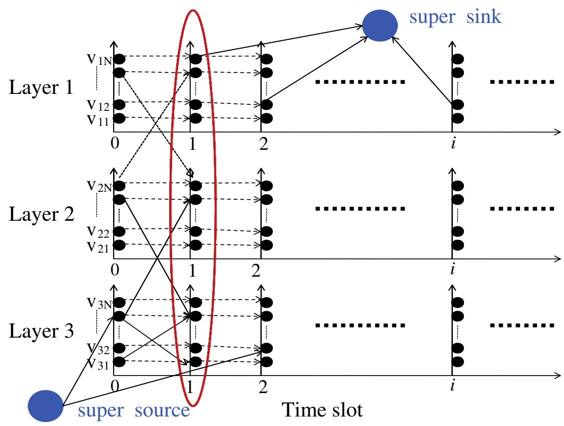


Fig. 2. An example of extended space time graph.

4.1. Integer linear programming

We assume $\mathcal{F}$ is a feasible flow on graph $G(V, E)$ and $f(u, v)$ is the flow through the edge (u, v). Different from Ref. [5], we require all flows contain an integer number of packets. Following the notations in the previous section, we formulate the minimum delay problem as an ILP:

$$\text{minimize}: \sum_{f(u,v)\in\mathcal{F}} d(u,v)f(u,v) \tag{4}$$

$$\text{subject to}: f(v_s, v^i) = q_v^i \tag{5}$$

$$\sum_{u\in V} f(u,v) = \sum_{w\in V} f(v,w), \quad v\in V\setminus\{v_s, v_d\} \tag{6}$$

$$f(u,v) \leqslant C_{(u,v)} \tag{7}$$

$$\sum_{u\in V} f(u,v) \leqslant C_v \tag{8}$$

$$\sum_{i=0,1,2,\ldots}\ \sum_{(u^i,v^{i+1})\in E_f} f(u^i, v^{i+1}) \leqslant P \tag{9}$$

$$\sum_{(u^i,v^{i+1})\in E_f} f(u^i, v^{i+1}) + \sum_{(v^i,w^{i+1})\in E_f} f(v^i, w^{i+1}) \leqslant \lambda T \tag{10}$$

$$\sum_{u\in V\setminus\{v_s,v_d\}} f(u, v_d) = \sum_v \sum_i q_v^i \tag{11}$$

where u^i or v^j is the instance of sensor u or v in the ith slot. The constraints from Eqs. (5) to (11) represent the traffic demand, flow conservation, channel capacity, node capacity, power capacity and transceiver capacity, respectively. By solving these equations, ILP can provide the optimal solution to satisfy all flow requirement with minimum average delay. However, ILP is NP hard with extremely high computational complexity.

4.2. Aggressive chronological projected graph

We next propose a greedy algorithm named *aggressive chronological projected graph* (ACPG) based on a much simpler undirected graph $G^o(V^o, E^o)$. The main inspiration is that the vertices at different time slots in $G(V, E)$ are not fully independent. They are different instances of the same node at different time slot, so they are spatially identical and temporally related. Moreover, the essentiality of routing is forwarding a packet to another appropriate candidate towards the destination, which happens over forwarding edges only in the extended space time graph. Therefore, we can construct a simple graph $G^o(V^o, E^o)$, where V^o only represents the overlay network nodes (the same vertex set in G) and E^o is dynamically updated by adding the connecting edges recorded in the short-duration trace chronologically. It can be seen as that $G^o(V^o, E^o)$ is compressed from the full extended space time graph $G(V, E)$ by projecting the forwarding edges in G to the same vertex set slot by slot from the beginning one. Optimal algorithms, such as ILP, need to construct the complete static graph G, while ACPG aggressively finds out possible flows in the dynamic graph G^o along with the graph construction. In the following, we present the construction of G^o and the operations of ACPG. For simplicity and comparison purpose, we construct G^o by compressing G. In practice, without constructing G in advance, we can directly construct G^o from the short-duration trace.

4.2.1. Construction of G^o

The projected graph $G^o(V^o, E^o)$ can be simply represented as the overlay graph of the network topology. Vertex $v_{ij} \in V^o$ ($i = 1,\ldots,M, j = 1,\ldots,N$) is the jth node in the ith layer, and edge $(v_{ij}, v_{kl}) \in E^o$ is the projection of edges $\left(v_{ij}^t, v_{kl}^{t+1}\right) \in E(t = 0, 1, \ldots)$. Since edges in different time slots in G can be projected to the same edge in G^o, we use (u, v, t, C) to represent an edge in G^o, where u^t and v^{t+1} are the connecting nodes in G and C is the capacity of that edge. The edge set E^o is initialized to be empty and updated in ACPG at each time slot.

We first introduce some basic concepts of the projected graph G^o:

Definition 1 (*Active node and inactive node*). At a time slot, a node $v \in V^o$ is active if there exists at least one route from the super source to it in G^o, which means that it is possible there could be packets arrived at this node from the super source until current time slot; otherwise, it is inactive. In G^o, an active node has at least one connecting edge and an inactive node has no associated edges.

Definition 2 (*Upstream node of a node* $v \in V^o$, U_v). Node u is the upstream node of v iff $(u, v, t, C) \in G^o$ has the smallest value t among all edges associated with v.

Initially, all nodes except for the super source v_s are *inactive*. Because we consider a real network scenario with limited transceiver capacity, buffer and power, we also maintain the following resource information for each node $v \in V^o$:

- $I_v(i)$: the maximum number of packets that can be transmitted or received during the ith slot, initialized to be λT.
- $C_v(i)$: the available storage in the ith slot, initialized to be buffer size W.
- P_v: the residual power for transmissions, initialized to be battery capacity P.

$I_v(i)$ and $C_v(i)$ are instant node properties at the ith slot, so we need to maintain multiple entries for a node. Each of them has an initial value and is updated in the iterative processes of ACPG. However, P_v is a node property, which cannot be recovered after consumption, so we only need to maintain one entry for a node.

4.2.2. Operations of ACPG

ACPG is operated in a slot by slot manner along with the construction of G^o. At each slot t, $t > 0$, it includes two routines: (1) *edge projection*, during which edges in G are projected to G^o (in practice, edges are added by examining the connectivity records at slot t in the short-duration trace) and (2) *routes reservation and graph update*, during which routes are discovered and G^o is updated.

4.2.2.1. Edge projection. To describe the edge projection process, we classify the edges $E \in E_f$ as *necessary* edges and *unnecessary* edges.

Definition 3 (*Necessary edge and unnecessary edge*). An edge $(u^t, v^{t+1}) \in E_f$ is necessary if it is a forwarding edge and either $u \in G^o$ or $v \in G^o$ is active. All holding edges and other forwarding edges in G are not necessary edges. Necessary edges mean that there are possible flows through these edges.

In the routine of *edge projection* at a time slot t, only the *necessary* edges in G will be projected to G^o, while *unnecessary* edges will be ignored directly because they cannot contribute for forwarding packets. The reasons are two-fold: (1) nodes do not forward packets through holding edges and (2) the forwarding edge, which connects two inactive nodes, $u \in G^o$ and $v \in G^o$, is impossible to be used for forwarding packets since no packets can arrive at either $u \in G^o$ or $v \in G^o$ before the time slot t when the edge is present, otherwise $u \in G^o$ or $v \in G^o$ should be active at this time slot t.

After the projection of $(u, v, t, C) \in E^o$, both u and v become active and update their upstream nodes respectively. If multiple edges have been projected to edge $(u, v) \in E^o$ up to the current slot t, only the one with the lowest time slot value can be used for the route discovery until it is replaced by the next earliest edge. Because we remove all holding edges and partial forwarding edges, ACPG can significantly reduce the problem size.

4.2.2.2. Routes reservation and graph update. The second routine finds possible routes up to slot t, reserves the resources and updates G^o. The second routine is only executed when the super sink $v_d \in G^o$ is active at slot t, because there must exist at least one flow from the super source v_s to v_d, which can be traced back along the upstream nodes from v_d with complexity $O(M \times N)$.

Theorem 4.1. *The discovered route by tracing back along upstream nodes is loop-free and the complexity of finding this route is upper bounded by $M \times N$, where M is the number of layers and N is the number of nodes in each layer.*

Proof. For a node $v \in G^o$, the direction to its upstream node U_v always points to another edge with smaller time slot, which further connects to the upstream node of U_v, until the super source. This property is guaranteed by the construction of G^o and the process of graph update in Section 4.2.2. Assume there is a loop of length k on the discovered route as $v_1 \to v_2 \cdots v_k \to v_1$. If $k = 2$, v_1 and v_2 become mutual upstream nodes (see Definition 4), which cannot exist in the graph. If $k > 2$, there must exist one node, for example v_1, which is the upstream node of v_k through the edge (v_1, v_k, t_k, C_k) and has its own upstream node u_2 through the edge (v_2, v_1, t_1, C_1). In this case, v_k should be the upstream node of v_1, which is contradictory to the fact that v_2 is the upstream node of v_1.

Since the discovered route by tracing back along upstream nodes is loop-free, this route can at most traverse all $(M \times N)$ nodes, which is the worst case. □

After finding a route, we reserve the necessary resources with the route capacity, which is the minimum node capacity, power capacity, transceiver capacity or edge capacity along the route. After reservation, nodes whose batteries are exhausted and edges whose capacities are reached become dead, and hence should be removed from G^o. Afterwards, related nodes should update their upstream nodes. The removal of nodes or edges and the update of upstream nodes may cause *mutual upstream nodes* and *outdated edges*, which are defined as follows.

Definition 4 (*Mutual upstream nodes and outdated edge*). Two nodes are mutual upstream nodes if they are the upstream nodes of each other and the edge connecting mutual upstream nodes is defined as an outdated edge.

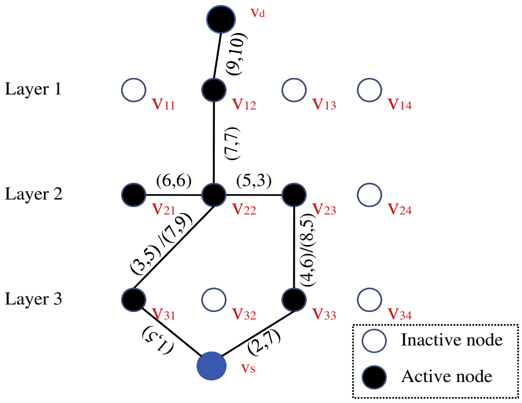
Both mutual upstream nodes and outdated edges cannot exist in G^o. Because if nodes u and v are mutual upstream nodes, then the edge (u, v, t', C) (t' is the earliest time slot among all projected edges between u and v) is outdated and impossible to be utilized since no packets can arrive at u or v before time slot t' through other routes. Therefore, outdated edges should be removed from G^o and the connecting nodes u and v should update their upstream nodes respectively until no mutual upstream nodes exist. If all edges associated with a node have been removed, this node goes back to be inactive. This update will guarantee that any route by tracing back along upstream nodes at any time in G^o is thus guaranteed to be loop-free.

These two routines are operated alternately until all traffic demands are satisfied. Through the aggressive route discovery along the earliest available edges, ACPG not only quickly finds low delay routes with significantly reduced complexity, but also summarizes the characteristics of the greedy routes, which reflect the properties of the underlying mobility pattern (see Section 6).

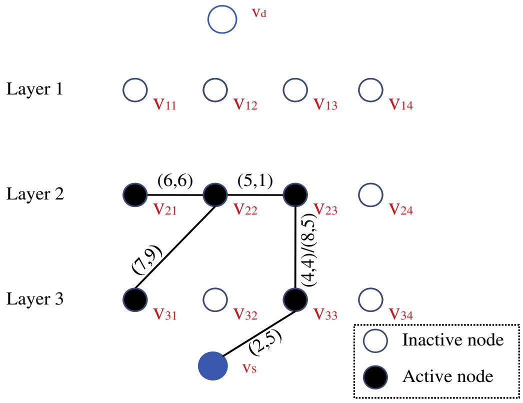
Fig. 3a shows an example of the projected graph for 12 nodes in 3 layers at slot 9 after adding forwarding edges in this slot, where we only mark the tuple (t, C) on an edge (overlapped edges are arranged chronologically and only the first edge is used for route discovery). There are possible flows on the graph since v_d is active. Following the upstream nodes, we can build the first route $\left(v_d \stackrel{9}{\leftarrow} v_{12} \stackrel{7}{\leftarrow} v_{22} \stackrel{3}{\leftarrow} v_{31} \stackrel{1}{\leftarrow} v_s\right)$ with capacity of 5 packets and delay of 8 slots, where $\stackrel{i}{\leftarrow}$ represents the packet is transmitted in the ith slot. After reserving the resources on nodes and edges on the route, edges $(v_{31}, v_{22}, 3, 5)$ and $(v_s, v_{31}, 1, 5)$ are removed and node $v_{22}(v_{31})$ changes its upstream node to $v_{23}(v_{22})$. Then we can find another route $\left(v_d \stackrel{9}{\leftarrow} v_{12} \stackrel{7}{\leftarrow} v_{22} \stackrel{5}{\leftarrow} v_{23} \stackrel{4}{\leftarrow} v_{33} \stackrel{2}{\leftarrow} v_s\right)$ with capacity of 2 packets and delay of 7 slots. This route exhausts edge $(v_{22}, v_{12}, 7, 2)$, so we remove edge $(v_{22}, v_{12}, 7, 2)$ and the outdated edge $(v_{12}, v_d, 9, 3)$, and make v_{12} and v_d inactive. The residual graph after routes reservation and graph update is shown as Fig. 3b.

4.3. Complexity analysis

In order to analyze the computational complexity, we assume the collected short-duration trace records the node connectivity during a period of S time slots in a network with M layers and N nodes in each layer. Therefore, we can construct the complete extended space time graph



(a) After edge projection.



(b) After graph update.

Fig. 3. An example of projected graph for 12 nodes in 3 layers at slot 9.

$G(V, E)$, where $|V| = M \times N \times (S + 1) \sim O(M \times N \times S)$, $|E_h| = M \times N \times S$ and $|E_f|$ is dynamic depending on the network situation. Although $|E_f|$ is not deterministic, we can assume $|E_f| \sim o(|E_h|)$ according to the definition of DTN.

Although the ILP aforementioned potentially provides the optimal solution, it is NP hard. Even when we relax the conditions to be a linear programming, the computational complexity is still as high as $O(n^6)$ (n is the number of vertex in G) using ellipsoid method [45]. In our extended space time graph, $n \sim O(M \times N \times S)$. Thus n may be huge, leading to extremely high computational complexity. Moreover, ILP is operated on the complete extended space time graph, causing significant spatial complexity.

In contrast, for ACPG, we build a much simpler graph $G^o(V^o, E^o)$, which only contains $M \times N$ vertices, and eliminate all holding edges. This simplification reduces the spatial complexity significantly.

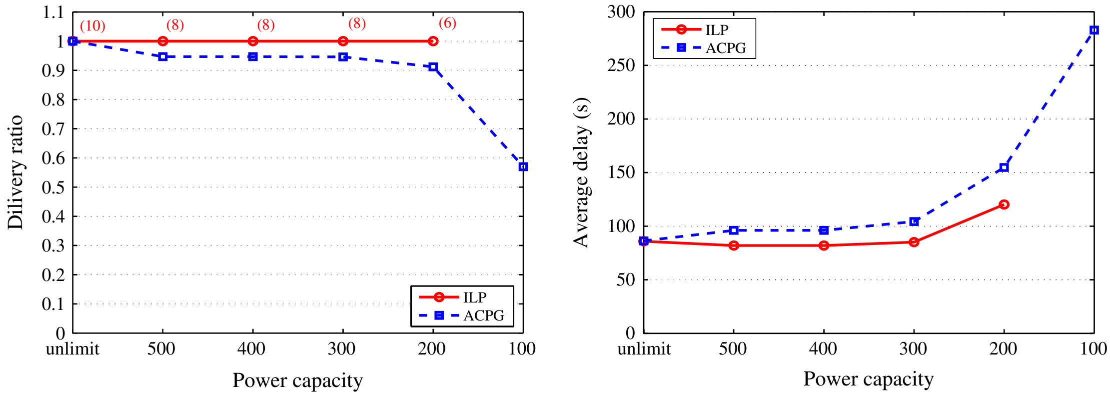
As we presented above, the operations of ACPG include two routines, (1) edge projection and (2) routes reservation and graph update. The first routine, edge projection, examines all forwarding edges in the trace and project them to G^o, which introduces the complexity of $|E_f|$. The second routine is executed only when there is a flow in the graph, and every execution finds out a flow of an integer amount which is at least 1. Thus there are at most F executions, where F is the total flow demand. In each execution, there are two steps, routes reservation and graph update. As proven in Theorem 4.1, the complexity of finding a route by tracing back along upstream nodes is upper bounded by $M \times N$, making the total routes reservation for all flows is upper bounded by $M \times N \times F$. Moreover, each route reservation leads to node status and edge update (a node might be changed to be inactive and an edge might be removed form G^o). During each execution, a node is updated at most once and each edge is removed at most once, leading to the total complexity of graph update for all flows is upper bounded by $(M \times N \times F) + |E_f|$. Therefore, we can conclude that the computational complexity for ACPG is $|E_f| + (M \times N \times F) + (M \times N \times F) + |E_f| \sim O(M \times N \times F) + o(M \times N \times S)$. Apparently, ACPG, compared to ILP, can also reduced the computational complexity significantly, which is demonstrated in Section 4.4.

4.4. Performance of ACPG

We evaluate the performance of ACPG by comparing it with optimal solutions from integer linear programming (ILP). The ILP formulation is based on an expanded space time graph. It, however, may not be able to provide feasible solutions (because not all traffic demands can be satisfied). ACPG, on the other hand, is a greedy algorithm that aggressively searches for routes using the earliest available edges. As we shall see, it not only achieves close to optimal solutions, but also characterizes the properties of the near optimal routes according to the mobility pattern.

We consider a three-layer underwater sensor network. Each layer covers a 600 m $\times$ 600 m horizontal area and the distance between two adjacent layers is 90 m. Four nodes are initially randomly deployed in each layer and move following the deterministic mobility model described in Eq. (1) (i.e. $k_4 = k_5 = 0$). The simulation in a small network is due to the high complexity of ILP (the proposed ACPG is scalable, we investigate larger networks in Section 5). Each sensor has buffer size of 30 packets and transmission range of 100 m. Sensors in the 3rd layer take turns to generate packets from the 500th second to the 1000th second with the rate of one packet per second in a round-robbin manner. The simulation runs to the 3000th second and the slot interval is chosen to be 10 s. We vary the power capacity from 500 to 100 transmissions, and *unlimit* means both the buffer and power capacities are not constrained.

Fig. 4 compares the performance of ILP and ACPG. The curves are plotted based on the average results of 10 runs and the numbers associated with the solid line in Fig. 4a indicate the number of runs in which ILP obtains feasible solutions. Although ILP is not feasible in some simulation runs because not all traffic demands can be satisfied, ACPG always provides solutions to deliver as many packets as possible. We observe that ILP and ACPG overlap for both delivery ratio and average delay in the unlimited condition, indicating that ACPG can perform as well as the optimal algorithm when there are no power, buffer and bandwidth constraints. It is interesting to note that ILP fails to provide feasible solutions in more and more runs when the power



(a) Delivery ratio. (b) Average delay.

Fig. 4. Performance comparison of ILP and ACPG.

capacity decreases, while ACPG provides good solutions with only slightly lower delivery ratio and higher average delay. Especially when the power capacity is as low as 100 transmissions, ILP provides no feasible solutions for all the runs, but ACPG provides results for almost 60% of all packets. If we exclude the unfeasible runs in ILP, ACPG can perform as well as the optimal ILP.

In Fig. 5, we plot the runtime of different algorithms, which represents the computational complexity. It is clear to see that ILP requires more and more time when the power capacity is lower (more flow splits and conjunctions), but ACPG always renders solutions in short time (around 0.1 s).

In summary, for all the scenarios we investigate, ACPG provides results close to the optimal ones. Moreover, ACPG can explore the relationship between the close to optimal routes and available historical information, which is related to the underlying mobility pattern, and provide guidance on how to utilize the historical information for instantiating prediction assisted single-copy routing protocols (see Section 5).

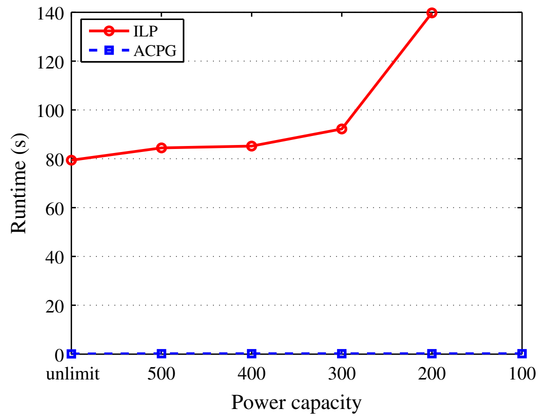


Fig. 5. Runtime comparison of ILP and ACPG.

5. Prediction assisted single-copy routing

Many routing protocols designed for DTNs predict future contacts based on some historical information. The more precise the prediction is, the better performance the routing achieves. This is especially true for single-copy routing. However, even the same historical information and prediction methods may lead to completely diverse performance under different mobility environments. Thus, what information can be used and how to predict the future become the main challenges. To address these challenges, we propose *prediction assisted single-copy routing* (PASR), that utilizes ACPG in a learning period to capture the characteristics of the mobility pattern, and provide guidance on route selection. Actually, ACPG does not examine the mobility model directly. In ACPG, we first demonstrate that ACPG can provide close to optimal solution by comparing with ILP, which means the routes discovered in ACPG are appropriate under current mobility model. Then we characterize the common features of these greedy routes and explore their relationship with the historical information, which are both closely related to the underlying mobility model. Finally we can conclude which historical information can be used and how to be used to predict future contacts and build heuristic single-copy routing schemes. The routes formed from the predication assisted single-copy routing schemes share the same features with the routes discovered in ACPG. In the following, we first present the generic scheme of PASR, and then describe how to instantiate PASR to construct two specific protocols for underwater sensor networks with different mobility patterns.

5.1. How PASR works?

5.1.1. Historical information

If the mobility pattern is stable for a long time, the history can tell the future. Widely used historical information includes:

- Recent trajectory: the geographic locations just visited.
- Average contact duration: the average duration of a contact.

- Average inter-contact duration: the average duration between two contacts. A contact coupled with the next inter-contact interval is called a *period*.
- Last contact time: the last time two nodes contacted.
- Contact frequency (or contact probability): the average contact frequency with another node or a landmark.

Not all the above information is available in a network or related to the mobility pattern. ACPG captures what historical information predicts the future with current mobility model.

5.1.2. Guidance from ACPG

The following properties of routes and node contacts, which are closely related to the underlying mobility model, can be captured by ACPG:

- Geographic preference: this is defined as the common feature of geographic locations that the greedy routes prefer. It is useful in geographic-related networks where nodes prefer certain geographic areas, or in landmark-based networks where nodes visit some landmarks frequently.
- Contact periodicity: this describes whether any pair of nodes have periodic contacts. Two nodes may have strict/weak contact period with fixed/variable contact and inter-contact durations. This periodicity may not last for the whole network lifetime.
- Inter-contact time distribution: e.g., uniform or exponential distribution. It can be obtained through curve fitting.
- Contact probability: the contact probability with another node or one landmark in a certain time interval.

5.1.3. Predict the future

After ACPG characterizes the mobility pattern, it suggests what historical information can be used for prediction.

- If the guidance exhibits geographic preference, a node can use it to determine whether to forward packets to a neighbor or not. For example, if the current neighbor will travel to a location which is preferred in ACPG with high probability, then it is qualified to be the next relay.
- If mobility shows contact periodicity, we can utilize the average contact duration and average inter-contact duration to estimate the future periods using linear prediction, and utilize the last contact time to estimate the next contact time with high accuracy.
- If the inter-contact time follows some well-known distribution, then the last contact time can be used to predict whether a node is approaching or departing away from another node. Several models have been exploited in [20].
- If a node contacts another node or landmark with a certain probability, the future contacts may be modeled as a semi-Markov process as described in [7].

In summary, ACPG connects history and future through proper information selection and prediction. Hence, efficient PASR can be instantiated following this procedure.

5.2. Instantiating PASR

We next describe how to instantiate PASR for different mobility models. For illustration, we consider three models in an underwater sensor network. This network consists of 3 layers and 15 nodes in each layer (in addition, the surface layer has one sink in the middle). Each layer covers a $800\ \text{m} \times 800\ \text{m}$ square area and the distance between two adjacent layers is 40 m. Each node has a buffer of 100 packets, a transmission range of 50 m and a transmission rate of 50 packets per second. The power capacity varies from 300 to 30 transmissions, and the slot duration is 10 s. We examine UWSNs with three mobility patterns.

5.2.1. UWSN in regular currents

The first mobility model we investigate assumes all nodes in the network float with the regular currents following Eq. (1) with $k_4 = k_5 = 0$. We first obtain guidance about the underlying mobility pattern from ACPG, then propose a PASR protocol accordingly.

5.2.1.1. Guidance from ACPG. Since our network consists of three layers which can be treated as three geographic areas and nodes move with regular currents, we focus on two properties: geographic preference and contact periodicity.

The geographic preference in this network means nodes in which layer are preferred. Results from ACPG show that most nodes highly prefer forwarding packets to an upper layer node directly even when having previous contacts with many nodes in the same and lower layers. Thus we obtain the first guidance: *an upper layer node is more preferred than other layer nodes*.

The contact periodicity for a pair of nodes is declared if a certain contact period repeats more than four times within 150 slots. We find that more than 80% of pairs observe periodicity. This leads to the second guidance: *nodes pairs have periodic contacts*. We should note that ACPG only indicates periodicity, the period durations for different pairs are different. This periodic contact property dominates the feasibility and accuracy of future contact prediction and the length of prediction window.

5.2.1.2. Protocol following ACPG. Based on the detailed depiction of the network mobility and the guidance obtained above, we propose a specific PASR for this network, *energy efficient history prediction assisted routing* (EEHPA). This scheme includes two essential operations: *prediction* and *per-contact forwarding decision*.

Both prediction and forwarding decision at a node u are built based on an *prediction vector* (PV), donated as E_u, as shown in Fig. 6. PV is a vector of tuples (i, v, D_v) to predict the potential contacts in the following T slots (T is the length of prediction window) and estimate delay to the sink. In each tuple, i is the prediction slot, v is the best potential relay in this slot and D_v is the estimated delay through this relay v to the sink. For example, a tuple $(5, v_1, 20)$ means that the current node u expects to contact the node v_1 at the 5 slots later with an estimated delay of 20 slots through this node v_1. In addition to E_u, node u also maintains an individual PV (iPV) E_u^v for each neighbor node v it met recently. E_u is updated from all iPVs it

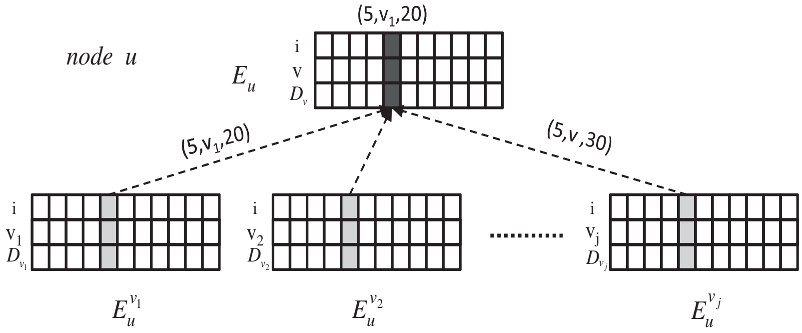


Fig. 6. PV and entries in node u.

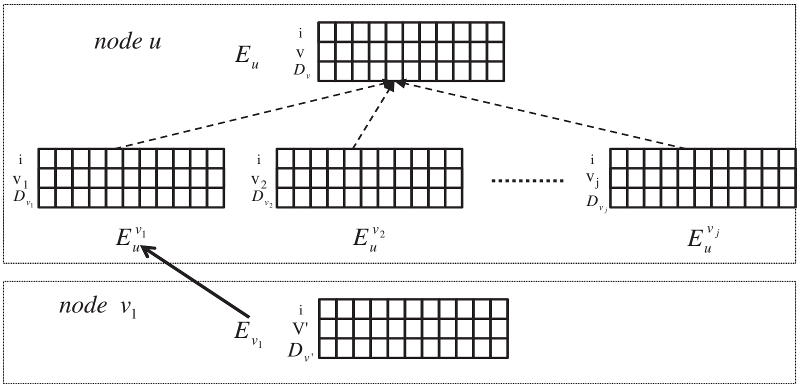


Fig. 7. The prediction update procedure for node u with node v_1.

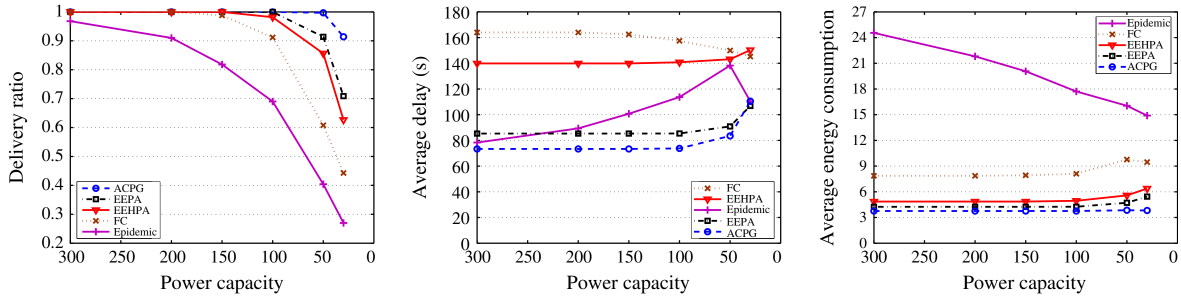
maintains. For a time slot t in the prediction window, the entry in E_u is selected from entries at the corresponding slot in all iPVs with the minimum estimated delay. As illustrated in Fig. 6, the node u expects to contact nodes v_1 and v_j at 5 slots later, and the estimated delay is 20 slots if forwarding a packet to v_1 or 30 slots if forwarding a packet to v_j. Thus, E_u can choose the tuple (5, v_1, 20) from v_1 as its entry at time slot t. The PV E_u is passively updated every time an iPV is updated.

Fig. 7 represents the update procedure when u contacts its neighbor v_1 (u and v_1 are in the communication range of each other) at a time slot. During the communication, v_1 will upload its prediction vector E_{v_1} to u, which is used by u to update the corresponding entry $E_u^{v_1}$ with the predicted future contacts between them. The update procedure is as follows: node u first clears all tuples in $E_u^{v_1}$ and predicts the future contacts with this node v_1 in the following T slots. Since ACPG indicates weak periodicity in this network, we choose the previous two periods (two contact durations and two inter-contact durations) to estimate the future periods using linear prediction. For each predicted contact time slot t', u will find the smallest estimated delay $D_{v'}$ in E_{v_1} after t', then update the corresponding D_{v_1} as $D_{v'}$ for the time slot t' in $E_u^{v_1}$. This tuple then becomes (t', v_1, $D_{v'}$), which means node u can forward packets to node v at time slot t' and expect the delay of $D_{v'}$. Once D_{v_1} is updated, node u will check the corresponding item with time slot t' in E_u: if the current estimated delay in this slot is larger than D_{v_1}, then this tuple will be updated as (t', v_1,D_{v_1}). PVs are exchanged only when they are updated, and can be piggybacked to the *HELLO* message at the beginning of each slot. Therefore, the update is not frequent owing to the sparse connectivity in DTNs.

Per-contact forwarding decisions are made based on the PV and the guidance from ACPG every time a node encounters a neighbor. Since results from ACPG indicate that few nodes forward packets to a lower layer, we only allow forwarding to nodes in the upper or the same layer for simplicity. We take node u to illustrate the decision. When u can communicate with v, it searches the expected delay D_v through v from E_u^v and the minimum expected delay $D_{v'}$ through the predicted best relay v' from E_u. Then node u makes positive forwarding decisions to v under two conditions: (1) $D_v \in [D_{v'}, D_{v'} + \delta_1)$ and (2) $D_v \in [D_{v'} + \delta_1, D_{v'} + \delta_2]$ and node v is in the upper layer. The parameters δ_1 and δ_2 are called *prediction error tolerances*. These tolerances, whose values are small with accurate predictions or large otherwise, are used to compensate the prediction error.

5.2.2. UWSN in currents with randomness

The second mobility model we exploit involves random movements by following Eq. (1) with non-zero k_4 and k_5. The randomness simulates the impact from environment, which may lead to estimation errors and prediction errors in real systems. We notice that PASR can tolerate these



(a) Delivery ratio. (b) Average delay. (c) Average energy.

Fig. 8. Performance comparison of various routing schemes in UWSN with regular currents.

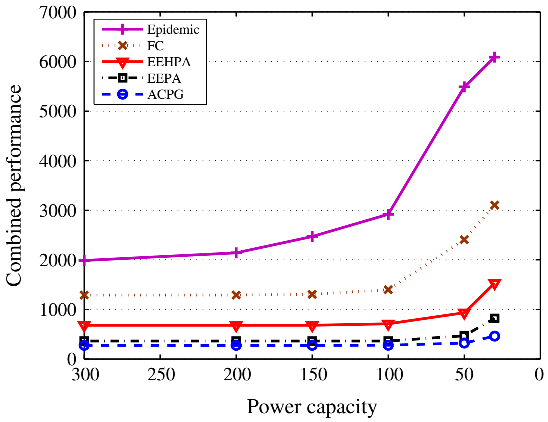


Fig. 9. Combined performance in UWSN with regular currents.

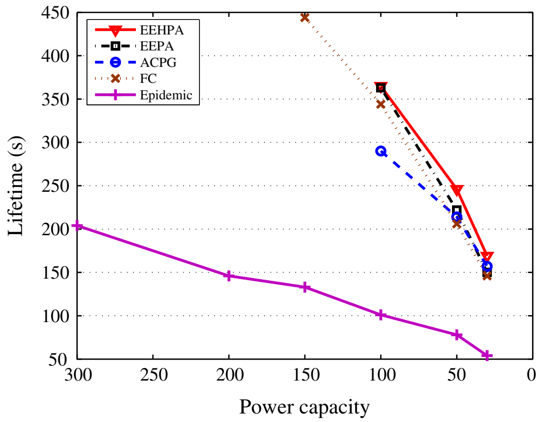


Fig. 10. Lifetime in UWSN with regular currents.

errors to some extent since ACPG just captures the general properties of the majority of nodes, who exhibit similar mobility patterns. In this setting we use EEHPA with larger δ_1 and δ_2, and demonstrate its performance in Section 6.

5.2.3. UWSN in irregular currents

The last mobility model incorporates irregular currents which will significantly change the underlying mobility pattern. Through this setting, we can evaluate whether ACPG discovers this change and reacts correspondingly.

We assume that nodes in the first two layers will be affected by an irregular water current, which drifts nodes away from the center of the network area. The nodes affected switch between the regular current and irregular current every 10 s. To provide connectivity to the sink, one node is anchored in the middle of the first two layers, which are not affected by the irregular current.

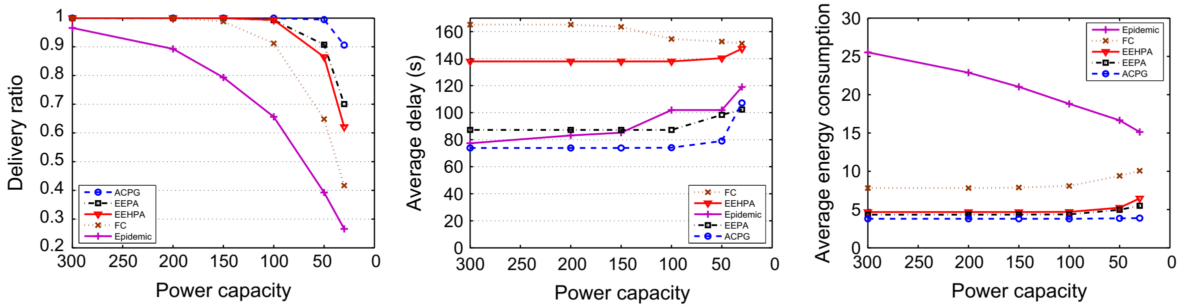
After executing ACPG under this mobility, we find that, affected by the irregular current, most nodes in the bottom layer route packets to the center area through nodes in the same layer to take advantages of anchor nodes. We also notice that only the nodes in the same layer have contact periodicity. Thus we obtain the following two guidance from ACPG: (1) a node in the same layer is preferred and (2) only predict for nodes in the same layer.

Therefore, we modify EEHPA to obtain a new prediction assisted single-copy routing scheme, named iEEHPA, according to the new guidance. In iEEHPA, we adopt the techniques and operations in EEHPA, but only predict future contacts for nodes in the same layer in the prediction update phase, and prefer nodes in the same layer in the forwarding decision phase.

6. Performance evaluation

In this section, we evaluate the performance of the instantiated PASR protocols for the UWSN described earlier. In this network, nodes in the bottom layer randomly generate 300 packets from the 500th second with the total generation rate of one packet per second. In each mobility model, we compare instantiated PASR with the following schemes:

- ACPG: serves as the lower-bound.
- EEPA: *energy efficient prediction assisted* routing. It differs from EEHPA by precise predictions using the deterministic terms in Eq. (1). This is an idealized scheme for UWSN since we do not know the precise mobility model in practice. EEPA is executed in the first two mobility models, and we show that EEHPA performs as well as EEPA although it only uses historical information.



(a) Delivery ratio. (b) Average delay. (c) Average energy.

Fig. 11. Performance comparison of various routing schemes in UWSN with regular currents and randomness.

- First Contact (FC): The single-copy routing by forwarding packets to the first node encountered without any prediction [5,18]. If multiple nodes are contacted at the same time, one in the upper layer is preferred.
- Epidemic: a flooding scheme [4]. To save energy, we allow epidemic ACK to be broadcasted through the network, which is used to delete useless copies.

To compare performance, we adopt the following metrics:

- Delivery ratio: the ratio of packets delivered.
- Average delay: the average delay for all delivered packets.
- Average energy consumption: the average number of transmissions needed to successfully deliver a packet.
- Combined performance: defined as $\frac{\text{Energy consumption} \times \text{Delay}}{\text{Delivery ratio}}$, used as a comprehensive performance measurement [8]. The smaller the value, the better performance is.
- Lifetime: the time when the first node dies owing to power exhaustion after packets generation.

6.1. UWSN in regular currents

We first compare various routing schemes in Fig. 8. When the network resources change from loosely to stringently constrained, it is not surprising to see that the delivery ratio of Epidemic drops rapidly to 0.3 when the power capacity is 30. This is because Epidemic uses too much energy during the flooding as shown in Fig. 8c and exhausts sensors quickly. This indicates that Epidemic is not suitable for resource constrained networks. Meanwhile, FC performs better than Epidemic with higher delivery ratio, but it degrades quickly especially from the power capacity 100 to 30 since the aimless forwarding not only delays the packets, but also wastes energy. ACPG provides the best results under all criteria. With the guidance from ACPG, the performance of both EEPA and EEHPA approaches the results of ACPG. It is interesting to notice that EEHPA only causes a slightly higher delay than EEPA since its prediction based on historical information is not as precise as EEPA.

Fig. 9 shows the comprehensive performance comparison considering all criteria. We clearly observe that the proposed PASR protocols are close to ACPG and outperform others significantly. This confirms that ACPG leads to efficient PASR.

We also plot the network lifetime under different routing schemes in Fig. 10. We can see that single-copy schemes significantly extend the network lifetime with much fewer energy consumption compared to Epidemic, and almost overlap with the result from ACPG. We believe that the lifetime of multi-copy schemes fall into the between. It confirms that single-copy routing scheme with proper prediction is more energy efficient in resource constrained networks.

6.2. UWSN in currents with randomness

We now examine the performance under a non-deterministic mobility pattern with randomness by setting non-zero k_4 and k_5 in Eq. (1). We assume both k_4 and k_5 follow uniform distributions in the interval $[-\gamma, \gamma]$ and vary γ from 0.1 m/s to 1 m/s. Each node has the buffer size of 100 packets and the power capacity of 100 transmissions. EEPA still uses the deterministic terms in Eq. (1) for prediction.

Fig. 11 shows the performance comparison of different schemes in UWSN with regular currents and randomness. The results are similar to that in regular currents without randomness, and the conclusions are still valid even the current movement is not deterministic. Fig. 12 exhibits

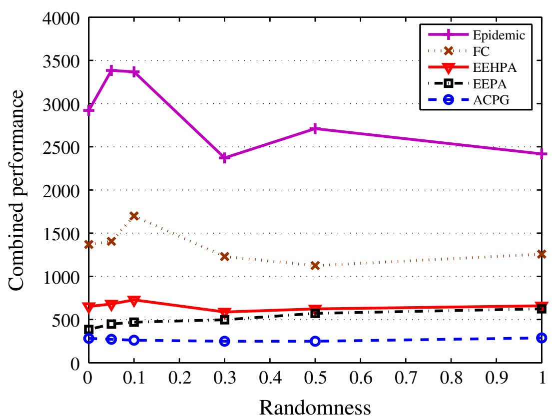
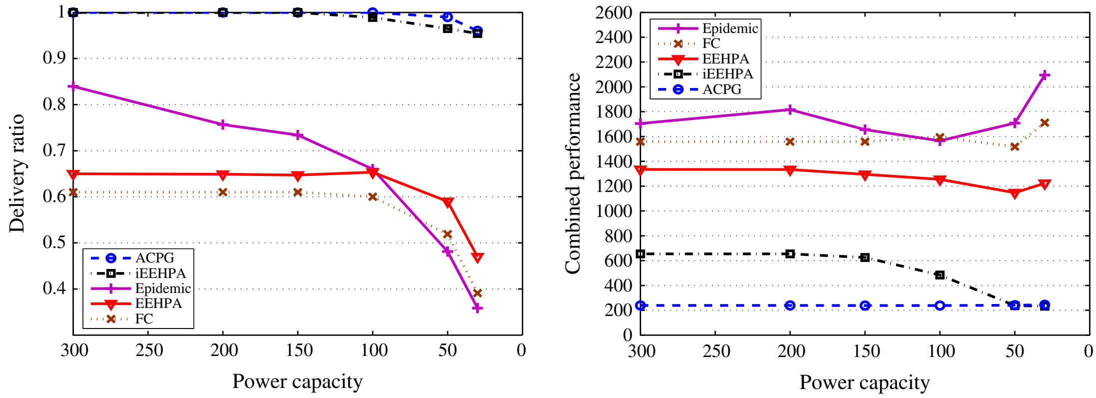


Fig. 12. Combined performance in UWSN with regular currents and randomness.



(a) Delivery ratio. (b) Combined performance.

Fig. 13. Performance comparison of various routing schemes in UWSN with irregular movements.

the comprehensive performance of various schemes under different levels of randomness. We can see that PASR outperforms Epidemic and FC and PASR can tolerate large randomness: its performance only slightly degrades when increasing the amount of randomness.

6.3. UWSN in irregular currents

We now explore the performance under impact from irregular currents as shown in Fig. 13. We observe that EEHPA is no longer suitable for this network and is even no better than FC (which uses no predictions). This indicates that inappropriate prediction may significantly degrade the performance. The modified iEEHPA performs much better and approaches ACPG. Therefore, we conclude that ACPG captures the properties of different mobility models and provides corresponding guidance for instantiating prediction assisted single-copy routing schemes.

7. Conclusions and future work

In this paper, we present a generic scheme *prediction assisted single-copy routing* (PASR) for UWSNs. We first propose *aggressive chronological projected graph* (ACPG). It is a greedy algorithm that provides results close to optimal and characterizes the properties of the underlying mobility pattern. We then design online heuristic protocols by choosing appropriate historical information and forwarding criteria based on the guidance from ACPG. We investigate an UWSN with various mobility patterns and randomness using two instantiated prediction assisted single-copy routing schemes, EEHPA and iEEHPA. Simulation shows that ACPG captures the properties of various mobility patterns and provides corresponding guidance, and the instantiated routing schemes outperform others.

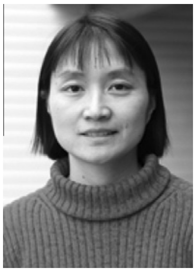
As future work, we would like to pursue the following directions: (1) we plan to explore the performance with different slot intervals, prediction window and prediction techniques and (2) we will extend PASR to other DTNs with different network structures and mobility models.

References

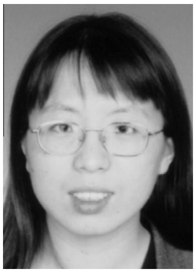
[1] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proc. of SIGCOMM, Karlsruhe, Germany, 2003.
[2] Z. Zhang, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges, IEEE Communications Surveys and Tutorials 8 (1) (2006).
[3] R.J. D'Souza, J. Jose, Routing approaches in delay tolerant networks: a survey, International Journal of Computer Applications 1 (17) (2010) 8–14.
[4] A. Vahdat, D. Becker, Epidemic Routing for Partially Connected Ad Hoc Networks, Tech. Rep., Duke University, April 2000.
[5] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, in: Proc. of SIGCOMM, Portland, Oregon, USA, 2004.
[6] C. Liu, J. Wu, An optimal probabilistic forwarding protocol delay tolerant networks, in: Proc. of ACM MOBIHOC, New Orleans, LA, USA, 2009.
[7] Q. Yuan, I. Cardei, J. Wu, Predict and relay: an efficient routing in disruption-tolerant networks, in: Proc. of ACM MOBIHOC, New Orleans, LA, USA, 2009.
[8] S.C. Nelson, M. Bakht, R. Kravets, A.F. Harris, Encounter based routing in DTNs, ACM SIGMOBILE Mobile Computing and Communications Review 25 (1) (2009).
[9] T. Spyropoulos, A. Jindal, K. Psounis, An analytical study of fundamental mobility properties for encounter-based protocols, International Journal of Autonomous and Adaptive Communications Systems 1 (1) (2008).
[10] W. Hsu, T. Spyropoulos, K. Psounis, A. Helmy, Modeling time-variant user mobility in wireless mobile networks, in: Proc. of IEEE INFOCOM, Anchorage, Alaska, USA, 2007.
[11] X. Zhang, J. Kurose, B.N. Levine, D. Towsley, H. Zhang, Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing, in: Proc. of ACM MOBICOM, Montréal, Québec, Canada, 2007.
[12] J. Leguay, T. Friedman, V. Conan, Evaluating mobility pattern space routing for DTNs, in: Proc. of IEEE INFOCOM, Barcelona, Spain, 2006.
[13] T. Hossmann, T. Spyropoulos, F. Legendre, Know thy neighbor: towards optimal mapping of contacts to social graphs for DTN routing, in: Proc. of IEEE INFOCOM, 2010, pp. 866–874.
[14] A. Mtibaa, M. May, C. Diot, M. Ammar, Peoplerank: social opportunistic forwarding, in: Proc. of IEEE INFOCOM, 2010, pp. 111–115.
[15] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, ACM SIGMOBILE Mobile Computing and Communications Review 7 (3) (2003).
[16] C. Lee, D. Chang, Y. Shim, N. Choi, T. Kwon, Y. Choi, Regional token based routing for DTNs, in: Proc. of ICOIN, Chiang Mai, Thailand, 2009.
[17] H. Dubois-Ferriere, M. Grossglauser, M. Vetterli, Age matters: efficient route discovery in mobile ad hoc networks using encounter ages, in: Proc. of the ACM MOBIHOC, Annapolis, Maryland, USA, 2003.

[18] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Efficient routing in intermittently connected mobile networks: the single-copy case, IEEE/ACM Transactions on Networking 16 (1) (2008).
[19] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proc. of SIGCOMM, Philadelphia, Pennsylvania, USA, 2005.
[20] H. Cai, D.Y. Eun, Aging rules: what does the past tell about the future in mobile ad-hoc networks? in: Proc. of ACM MOBIHOC, New Orleans, LA, USA, 2009.
[21] K.A. Harras, K.C. Almeroth, E.M. Belding-Royer, Delay tolerant mobile networks (DTMNs): controlled flooding in sparse mobile networks, Lecture Notes in Computer Science, 2005.
[22] J. Wu, M. Lu, F. Li, Utility-based opportunistic routing in multi-hop wireless networks, in: Proc. of IEEE ICDCS, Washington, DC, USA, 2008.
[23] J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, Research challenges and applications for underwater sensor networking, in: Proc. of IEEE WCNC, Las Vegas, Nevada, USA, 2006.
[24] J.-H. Cui, J. Kong, M. Gerla, S. Zhou, Challenges: building scalable mobile underwater wireless sensor networks for aquatic applications, IEEE Network, Special Issue on Wireless Sensor Networking 20 (3) (2006) 12–18.
[25] I.F. Akyildiz, D. Pompili, T. Melodia, State of the art in protocol research for underwater acoustic sensor networks, ACM SIGMOBILE Mobile Computing and Communications Review 11 (4) (2007) 11–22.
[26] J. Partan, J. Kurose, B.N. Levine, A survey of practical issues in underwater networks, Speical issue of ACM SIGMOBILE Mobile Computing and Communications Review 11 (4) (2007) 23–33.
[27] E.M. Sozer, M. Stojanovic, J.G. Proakis, Underwater acoustic networks, IEEE Journal of Oceanic Engineering 13 (1) (2000).
[28] Z. Guo, G. Colombo, B. Wang, J.-H. Cui, D. Maggiorini, G. Rossi, Adaptive routing in underwater delay/disruption tolerant sensor networks, in: Proc. of IFIP WONS, Garmisch-Partenkirchen, Germany, 2008.
[29] Z. Guo, B. Wang, J.-H. Cui, Prediction assisted single-copy routing in underwater delay tolerant networks, in: Proc. of IEEE GLOBECOM, Miami, FL, 2010.
[30] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: an approach to interplanetary internet, IEEE Communications Magazine 41 (6) (2003) 128–136.
[31] A. Pentland, R. Fletcher, A. Hasson, DakNet: rethinking connectivity in developing nations, IEEE Computer 37 (1) (2004) 78–83.
[32] S. Merugu, M. Ammar, E. Zegura, Routing in Space and Time in Networks with Predictable Mobility, Tech. Rep., Georgia Institute of Technology, 2004.
[33] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility, in: Proc. of IEEE PERCOM, White Plains, NY, USA, 2007.
[34] J. Xue, X. Fan, Y. Cao, J. Fang, J. Li, Spray and wait routing based on average delivery probability in delay tolerant network, in: Proc. of NSWCTC, Wuhan, China, vol. 2, 2009.
[35] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: Proc. of IEEE INFOCOM, Barcelona, Spain, 2006.
[36] A. Balasubramanian, B.N. Levine, A. Venkataramani, Replication routing in DTNs: a resource allocation approach, IEEE/ACM Transactions on Networking 18 (2) (2010) 596–609.
[37] G. Sandulescu, S. Nadjm-Tehrani, Adding redundancy to replication in window-aware delay-tolerant routing, Journal of Communications, Special Issue: Delay Tolerant Networks, Architecture, and Applications 5 (2) (2010) 117–129.
[38] E.P. Jones, L. Li, P.A. Ward, Practical routing in delay-tolerant networks, in: Proc. of WDTN, Philadelphia, PA, USA, 2005.
[39] I. Cardei, C. Liu, J. Wu, Q. Yuan, DTN Routing with probabilistic trajectory prediction, in: Proc. of WASA, Dallas, Texas, USA, 2008.
[40] X. Chen, A.L. Murphy, Enabling disconnected transitive communication in mobile ad hoc networks, in: Proc. of Workshop on Principles of Mobile Computing, colocated with PODC01, 2001, pp. 21–27.
[41] M. Musolesi, S. Hailes, C. Mascolo, Adaptive routing for intermittently connected mobile ad hoc networks, in: Proc. of IEEE WoWMoM, Washington, DC, USA, 2005, pp. 183–189.
[42] Z. Zhou, J.-H. Cui, A. Bagtzoglou, Scalable localization with mobility prediction for underwater sensor networks, in: Proc. of IEEE INFOCOM 2008, Mini-Conference, Phoenix, Arizona, USA, 2008.
[43] S.P. Beerens, H. Ridderinkhof, J. Zimmerman, An analytical study of chaotic stirring in tidal areas, Chaos, Solitons and Fractals (1994).
[44] L. Ford, D. Fulkerson, Flows in Networks, Princeton University Press, 1962.
[45] D. Bertsimas, J.N. Tsitsiklis, Introduction to Linear Optimization, Athena Scientific, 1997.

Zheng Guo is currently a Professor of Xi'an University of Posts and Telecommunications, Xi'an, China. He received his bachelor degree in Electronic Engineering from University of Science and Technology of China (USTC) in 2005, and Ph.D. degree in Computer Science and Engineering from University of Connecticut in 2010. His research interests are network coding, channel coding, routing and delay/disruption tolerant network in the area of Underwater Sensor Network (UWSN).



Bing Wang is currently an Associate Professor of the Computer Science & Engineering Department at the University of Connecticut. She received her B.S. degree in Computer Science from Nanjing University of Science & Technology, China in 1994, and M.S. degree in Computer Engineering from Institute of Computing Technology, Chinese Academy of Sciences in 1997. She then received M.S. degrees in Computer Science and Applied Mathematics, and a Ph.D. in Computer Science from the University of Massachusetts, Amherst in 2000, 2004, and 2005 respectively. Her research interests are in Computer Networks, Multimedia, and Distributed Systems. She received NSF CAREER award in 2008.



Jun-Hong Cui received her B.S. degree in Computer Science from Jilin University, China in 1995, her M.S. degree in Computer Engineering from Chinese Academy of Sciences in 1998, and her Ph.D. degree in Computer Science from UCLA in 2003. Currently, she is an Associate Professor in the Computer Science and Engineering Department at University of Connecticut (UConn). She also serves as the Assistant Dean for Graduate Studies and Diversity of School of Engineering at UConn.

Her research interests cover the design, modelling, and performance evaluation of networks and distributed systems. Currently, her research mainly focuses on algorithm, protocol and system design and development in underwater sensor networks, autonomous underwater vehicle (AUV) networks, and distributed cyber-aquatic systems. She has also been conducting active research on smart health and smart transportation systems. At UConn, she leads UbiNet (Ubiquitous Networking) Lab and UWSN (UnderWater Sensor Network) Lab. See http://www.cse.uconn.edu/~jcui/ for her recent projects and publications.

She is actively involved in the community as an organizer, a TPC member, and a reviewer for numerous conferences and journals. She now serves as an Associate Editor for Elsevier Ad Hoc Networks. She co-founded the first ACM International Workshop on UnderWater Networks (WUWNet'06), and she is now serving as the WUWNet steering committee chair. She received 2007 NSF CAREER Award and 2008 ONR Young Investigator Award. She also received the United Technologies Corporation (UTC) Professorship in Engineering Innovation award at UConn in 2008 and UCLA Engineering Distinguished Young Alumnus Award in 2010.