A Routing Protocol for Power Constrained Networks with Asymmetric Links

Guoqiang Wang, Yongchang Ji,
Dan C. Marinescu
School of Computer Science
University of Central Florida
Orlando, FL 32816-2450
{gwang,yji,dcm}@cs.ucf.edu

Damla Turgut
Department of Electrical and Computer
Engineering
University of Central Florida
Orlando, FL 32816-2450
turgut@cpe.ucf.edu

ABSTRACT

In many instances, an ad hoc network consists of nodes with different hardware and software capabilities as well as power limitations. This is the case of ad hoc grids where devices such as desktops, laptops, robots, palmtops, sensors, and actuators collaborate to solve computational problems. In such a heterogeneous environment, the nodes have various degrees of mobility and range and the communication links are asymmetric: node i may be able to reach node j, but j may not be able to reach i. A^4LP, is a Location-Aware and Power-Aware routing protocol designed primarily for heterogeneous Ad hoc networks with Asymmetric links.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Algorithms

Keywords

m-party proxy set, m-limited forwarding, forwarding fitness function

1. INTRODUCTION AND MOTIVATION

In many instances, an ad hoc network consists of nodes with different hardware capabilities and power limitations. Yet, routing protocols for ad hoc networks routinely assume that all communication links are bidirectional. In this paper we consider the case when the nodes of an ad hoc network have various degrees of mobility and range and the communication links are asymmetric; node i may be able to reach node j, but j may not be able to reach i. Power consumption is a major issue in an ad hoc network as nodes become inoperative once they have depleted their power. *Power-aware* routing algorithms minimize the power consumption and, whenever possible, avoid the nodes with a low level of residual power. The task of minimizing power consumption can be facilitated the location of individual nodes are known. Global Positioning Systems (GPS) [3] are routinely embedded into mobile systems and *location-aware* routing algorithms have the potential to extend the life of individual nodes of an ad hoc network.

An *ad hoc grid* is a heterogeneous system without a fixed infrastructure, all its components are mobile [12]; it consists of a hierarchy of systems with different hardware, software, and communication capabilities. Informally, we identify four classes of systems: $C1, C2, C3$, and $C4$ (Figure 1). Powerful laptops or regular desktop computers with significant amounts of secondary storage, a variety of I/O devices, high speed network access, and long life batteries, mounted on all-terrain vehicles, airplanes, ships are called $C1$ systems. Robots (terrestrial or airborne) and laptops with infrared, wireless, and/or satellite communication, are examples of $C2$ systems. Wearable computers, PDAs with little or no secondary storage and crossover devices (e.g., portable phones with PDA) are included in the $C3$ systems category. A $C4$ system is a low-cost smart sensor, e.g., a video camera, an infrared detector, a source of light, a generator of acoustic signals, or another type of sensor or actuator. Most systems are expected to have GPS capabilities, or, as in the case of sensors, to record their approximate position at the time when they were installed. The hierarchy presented here serves only an illustrative purpose as the number of programmable mobile devices increases every day (see for example http://www.fawcette.com/wireless/sherman/).

There are many potential applications of ad hoc grids for wild fire prevention and control, disaster management, peace keeping operations in a remote part of the world, sporting events, discovery expeditions, natural resource exploration, and battlefield management. Olympic Games, Tour de France, the Trans-Africa car rally, the dog sled race in Alaska, archeological excavations, an expedition to K9, and underwater oil exploration, are just a few practical cases when a grid-like environment is necessary to reliably support the coordinated effort of a group of individuals working under extreme conditions. Communication and computing facets of an ad-hoc grid are deeply intertwined. One node may request computations distributed over a set of nodes (e.g., running models to determine the humidity and combustion in an area affected by a wild fire, population evacu-

PE-WASUN'04, October 7, 2004, Venezia, Italy.

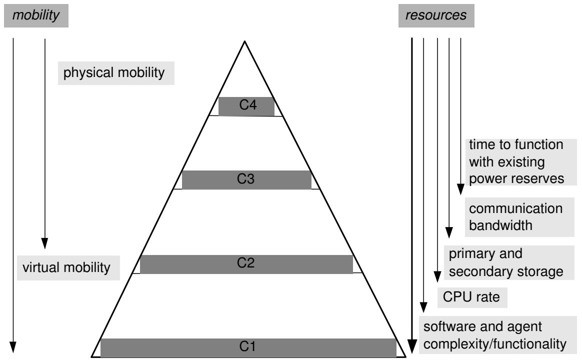

mobility
resources
physical mobility
C4
C3
C2
C1
virtual mobility
time to function with existing power reserves
communication bandwidth
primary and secondary storage
CPU rate
software and agent complexity/functionality

Figure 1: An ad hoc grid is a heterogeneous system without a fixed infrastructure, all its components are mobile; we illustrate a four-level hierarchy of $C1, C2, C3$, and $C4$ systems and their properties.

ation models, and so on).

In this paper, we concentrate on the communication aspect of ad hoc grids and we introduce A^4LP, a Location-Aware and Power-Aware routing protocol designed primarily for heterogeneous MANET with Asymmetric links. Most of the existing protocols focus on a single aspect such as either location or power-awareness and do not generally consider the existence of asymmetric links even though a heterogeneous ad hoc network is mostly composed of asymmetric links. Therefore, the novelty of this work comes from the combined effects of not only location and power-awareness but also the usage of asymmetric links throughout.

2. RELATED WORK

In a heterogeneous MANET, assorted devices with different computation and communication capabilities co-exist. As opposed to static networks, where symmetric links are the standard, routing in a heterogeneous MANET is dominated by asymmetric links. There are several reasons. (i) Due to the varying transmission ranges of the devices, only the devices with stronger communication capability may reach the devices with weaker communication capability. (ii) In order to achieve power-awareness, it is assumed that devices adjust their transmitting power according to their residual power such that their lifetime are extended. In the process, some of the symmetric links may become asymmetric when the communication capability of a node degrades due to decrease in the residual power. (iii) The transmission ranges of devices with same communication capabilities may vary due to fading and random transient phenomena.

Ad hoc routing protocols are generally classified into two categories: (1) *table-driven*, or *proactive*, such as DSDV[14], CGSR[2], DREAM[1], and OLSR[7]; (2) *on-demand*, *reactive*, or *source-initiated*, such as DSR[8], AODV[15], LAR[10], and TORA[13]. In proactive routing protocols, nodes periodically propagate routing update advertisements with their neighbors in order to maintain up-to-date routing information. Routes are immediately available upon a node's request. In reactive routing protocols, a route is found on demand when the source needs to send a packet to a destination. Routes are valid only for a limited period, after which routes are considered to be obsolete. No periodical route information propagation is required. Proactive protocols are energy inefficient because (i) the control message overhead grows at $O(n^2)$ since the routing information advertisement is introduced into the network by frequent system-wide broadcasting; (ii) nodes maintain routes for each destination in the network, which is almost impossible for most of the nodes of class $C3$ and $C4$ [12]; (iii) a considerable portion of the routes are never used and maintaining them causes unnecessary energy consumption. Compared to proactive protocols, reactive protocols save the bandwidth and energy, while higher latency is incurred due to discovering routes on demand.

Protocols combining the features of both proactive and reactive protocols are sometime considered as a third category, *hybrid protocols*, such as Zone Routing Protocol (ZRP) [4]. In a hybrid protocol, routes for a subset of nodes are maintained in a routing table proactively while routes for the remaining nodes are discovered on on-demand basis. In ZRP, routes within a zone, centered at a node, are kept up-to-date by IntrA-zone Routing Protocol (IARP) [6], and routes outside the zone are discovered upon request by IntEr-zone Routing Protocol (IERP) [5].

With the emergence of GPS on mobile devices, protocols, called *location-based protocols*, are devised with the aid of location information to improve the performance further. LAR[10] and DREAM[1] are examples of such protocols.

Saving on power consumption is yet another important aspect in wireless communication protocols [9, 11]. Mobile devices are generally operated by battery power, and in some situations, such as sensors, the power source may not be rechargeable due to inaccessable terrains. Most of the current protocols calculate paths by minimizing either the hop count or the transmission delay. Nodes along the critical paths are chosen more often causing the depletion of their energy sooner than the other nodes. Since these nodes are essential in maintaining the network connectivity, a set of protocols, called *power-aware*, are proposed such as [16, 17].

Our proposed protocol, A^4LP, is both *location-* and *power-aware* routing protocol supporting asymmetric links that may be suitable for heterogeneous MANET. We classify A^4LP as a hybrid protocol due to the following aspects. The routes to In-, Out-, and In/Out-bound are maintained by periodic neighbor update and immediately available upon request, while the routes to other nodes in the network are obtained by a path discovery protocol. A^4LP proposes an advanced flooding technique - *m-limited forwarding.* Receivers can re-broadcast a packet only if it qualifies a certain *fitness* value specified by the sender. The flooding cost is reduced and shortest high quality path is likely to be selected by using m-limited forwarding. Moreover, the metrics used to choose from multiple paths are based on the *power consumed per packet* and *transmission latency.*

3. THE MODEL OF THE SYSTEM

Let $\mathcal{N}$ be the set of nodes in an ad hoc grid/network. We make several assumptions:

a. The number of nodes is relatively small, say $| \mathcal{N} |\leq 10^4$.

b. All nodes use the same frequency to communicate with one another. Once a node sends a message all the nodes in its transmission range hear the broadcast.

c. The mobility of individual nodes is limited and differs from one node to the other.

d. Nodes are able to adjust their transmitting power according to their residual power so that their lifetime are extended.

Every node $i \in \mathcal{N}$ is characterized by a minimal set of attributes, two invariants in time and two time-dependent:

1. Id, Id_i; unique string used for node identification.

2. Class, C_i; the nodes of a heterogeneous MANET are classified in several classes based on the hardware and software resources. Throughout this paper we assume a four level hierarchy.

3. Location at time t, $L_i(t)$; the geographical coordinates of the position of the node at time t, and

4. Residual power at time t, $P_i^{res}(t)$; the amount of power available at time t. If P_i^{max} is the maximum level of power that can be stored at node i there are two water marks (e.g., $highWaterMark_i = 0.7P_i^{max}$ and $lowWaterMark_i = 0.2P_i^{max}$) and say that at time t the node operates at
- *full-power level* if $P_i^{res}(t) \geq highWaterMark_i$,
- *normal-power level* if $P_i^{res}(t) > lowWaterMark_i$ and $P_i^{res}(t) < highWaterMark_i$, or
- *low-power level* if $P_i^{res}(t) \leq lowWaterMark_i$.

Other attributes can be derived from the ones in the minimal set.

The *transmission range* of node N_i at time t, $R_i(t)$, is a function of the residual power and possibly other factors including the configuration of the terrain, atmospheric conditions, and so on. For simplicity we assume that

$$R_i(t) = constant \times P_i^{res}(t).$$

The *distance* between two nodes, $i, j \in \mathcal{N}$, $d_{ij}(t)$ is a function of the position of the two nodes

$$d_{ij}(t) = d_{ji}(t) = f(L_i(t), L_j(t)).$$

The *average velocity* of a node i over an interval of time $\Delta t = t_2 - t_1, \quad t_2 > t_1$ is

$$v_i^{\Delta t} = f(L_i(t_2), L_i(t_1))/\Delta t.$$

The *mobility region* of node i over an interval of time $\Delta t = t_2 - t_1, \quad t_2 > t_1$ is a circle of radius

$$M_i^{\Delta t} = v_i^{\Delta t} \times \Delta t$$

centered at $L_i(t_1)$.

The Boolean *reachability function* $\mathcal{R}_{ij}(t)$ is defined as

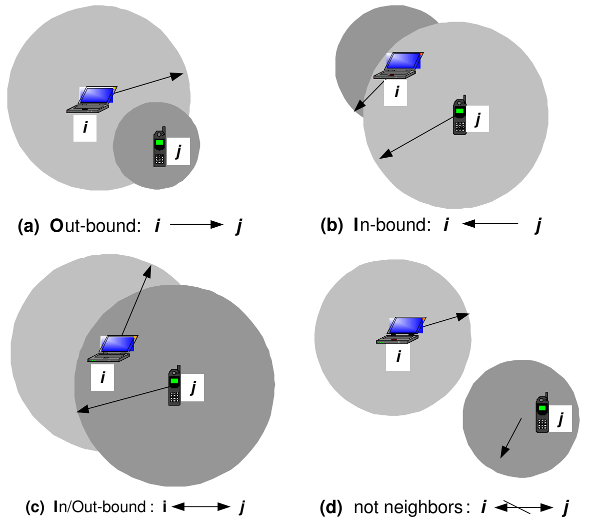
$$\begin{array}{lcl} \mathcal{R}_{ij}(t) = \text{true} & \Longleftrightarrow & R_i(t) \geq d_{ij}(t); \\ \mathcal{R}_{ij}(t) = \text{false} & \Longleftrightarrow & R_i(t) < d_{ij}(t). \end{array}$$

Definition 1. Two nodes $i, j \in \mathcal{N}$ are in *neighbor* relationship at time t if there is a direct communication link between them. We recognize several types of neighbors:

1. *Out-bound neighbor:* j is the out-bound neighbor of i, if i can reach j but j cannot reach i, as seen in Figure 2(a). In this case the link L_{ij} between the two nodes is unidirectional

$$\mathcal{R}_{ij}(t) = \text{true} \qquad \text{and} \qquad \mathcal{R}_{ji}(t) = \text{false}.$$

Call $Out_i(t) \subset \mathcal{N}$ the set of Out-bound neighbors of i at time t.



(a) Out-bound: $i \longrightarrow j$ (b) In-bound: $i \longleftarrow j$

(c) In/Out-bound : $i \longleftrightarrow j$ (d) not neighbors : $i \nleftrightarrow j$

Figure 2: The neighbor relationship between two nodes. (a) *j* is an Out-bound neighbor of *i*. (b) *j* is an In-bound neighbor of of *i*. (c) *j* is an In/Out-bound neighbor of *i*. (d) *i* and *j* are not neighbors.

2. *In-bound neighbor:* j is the In-bound neighbor of i, if j can reach i but i cannot reach j, as seen in Figure 2(b). In this case the link L_{ji} between the two nodes is unidirectional

$$\mathcal{R}_{ij}(t) = \text{false} \qquad \text{and} \qquad \mathcal{R}_{ji}(t) = \text{true}.$$

Call $In_i(t) \subset \mathcal{N}$ the set of In-bound neighbors of i at time t.

3. *In/Out-bound neighbor:* j is the In/Out-bound neighbor of i, if i and j can reach each other, as seen in Figure 2(c). In this case the link L_{ij} between the two nodes is bidirectional

$$\mathcal{R}_{ij}(t) = \text{true} \qquad \text{and} \qquad \mathcal{R}_{ji}(t) = \text{true}.$$

Call $InOut_i(t) \subset \mathcal{N}$ the set of In/Out-bound neighbors of i at time t.

The scenario that two nodes are not neighbors is illustrated in Figure 2(d).

Definition 2. If node i is an Out-bound neighbor of node j, we call i the *high-range node* (H-node) and j the *low-range node* (L-node) of the asymmetric link L_{ij}.

Definition 3. A set of m nodes $i_1, i_2, \ldots i_m \in \mathcal{N}$ are in an *m-party proxy set* if each node can reach the other $m - 1$ nodes either directly or through a subset of the other $m - 2$ members of the set.

PROPOSITION 1. *At least one of the links of an m-party proxy set must be bidirectional.*

PROOF. Suppose Proposition 1 is false, that is, there exists an m-party proxy set with no bidirectional link. Let the m nodes in the m-party proxy relationship be $i_1, i_2, \cdots, i_m \in \mathcal{N}$ and arbitrarily pick up an asymmetric link (i_u, i_v) where $1 \leq u, v \leq m, u \neq v$. Thus, node i_u can reach node i_v directly, but the reciprocal is not true, $\mathcal{R}_{i_u i_v}(t) = true$ and $\mathcal{R}_{i_v i_u}(t) = false$, which is equivalent to

$$R_{i_u}(t) \geq d_{i_u i_v}(t) > R_{i_v}(t) \tag{1}$$

By the definition of m-party proxy set, there exists at least a path for node i_v to reach node i_u. Let us choose the shortest path from node i_v to node i_u. There are no duplicate nodes on this path, otherwise a shorter path can be obtained by removing the sub-path consisting of all the nodes connecting the two duplicate nodes. Call the set of nodes on the shortest path $(i_v, i_{N_1}, i_{N_2}, \cdots, i_{N_p} \cdots, i_{N_k}, i_u)$, where $N_p \neq N_q, N_p \neq u, N_p \neq v, 1 \leq p, q \leq k \leq m-2, p \neq q$. Similar to (1), we have

$$\begin{array}{rcl} R_{i_v}(t) & \geq & d_{i_v i_{N_1}}(t) > R_{i_{N_1}}(t) \\ & \geq & d_{i_{N_1} i_{N_2}}(t) > R_{i_{N_2}}(t) \\ & \geq & \cdots > R_{i_{N_p}} \geq d_{i_{N_p} i_{N_{p+1}}}(t) > R_{i_{N_{p+1}}}(t) \\ & \geq & \cdots > R_{i_{N_k}} \geq d_{i_{N_k} i_u}(t) > R_{i_u}(t) \end{array} \quad (2)$$

Equations (1) and (2) are contradictory, thus, Proposition 1 must be true. □

Figure 3(a) and 3(b) show two possible configurations of a three-party proxy set with unidirectional links only. The configuration in Figure 3(a) is infeasible according to Proposition 1, while the configuration in Figure 3(b) is infeasible because k cannot reach either i or j.

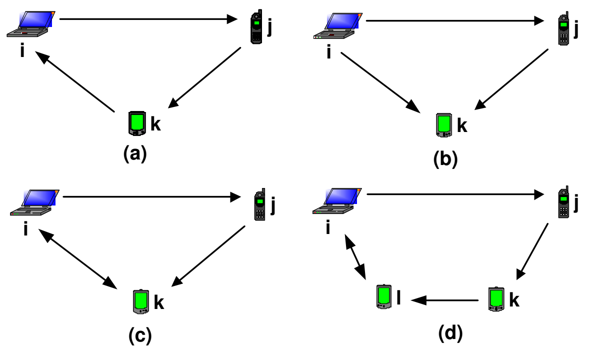


Figure 3: Three-party and four-party proxy sets. (a) An infeasible scenario for a three-party proxy set involving three unidirectional links. (b) A second infeasible scenario for a three-party proxy set involving three unidirectional links. (c) A feasible scenario for a three-party proxy set with one bidirectional link. (d) A feasible scenario for a four-party proxy set with one bidirectional link.

Proposition 2. *There is at least one configuration of an m-party proxy set with one bidirectional link.*

The configuration in Figure 3(c) and any configuration obtained by a permutation of the Ids of the nodes in the set has one bidirectional link. In this configuration i can reach j and k directly, j can reach k directly and i via k, and finally k can reach i directly and j via i. The ranges and distances among the nodes of the configuration in Figure 3(c) are:

$$\begin{cases} R_j < d_{ij} \leq R_i, \quad R_k < d_{jk} \leq R_j; \\ d_{ik} \leq R_i, \quad d_{ki} \leq R_k. \end{cases}$$

To show that there is at least one configuration of four-party proxy set with one bidirectional link we consider the configuration in Figure 3(d). Since there is a loop $i \rightarrow j \rightarrow k \rightarrow l \rightarrow i$, every node can reach the other nodes in the set. The ranges and distances among the nodes of the configuration in Figure 3(d) must satisfy the following constraints:

$$\begin{cases} R_j < d_{ij} \leq R_i, \quad R_k < d_{jk} \leq R_j, \quad R_l < d_{kl} \leq R_k; \\ d_{li} \leq R_i, \quad d_{li} \leq R_l; \\ d_{ik} > R_i, \quad d_{jl} > R_j, \quad d_{ki} > R_k, \quad d_{lj} > R_l. \end{cases}$$

In the general case of an m-party proxy set consider the nodes $i_1, i_2, \ldots i_m \in \mathcal{N}$ connected as follows: nodes i_k and i_{k+1} $(1 \leq k \leq m-2)$ are connected by unidirectional links from node i_k to node i_{k+1}, and nodes i_1 and i_m are connected by a bidirectional link. Thus the ranges and distances among nodes must satisfy the following constraints

$$\begin{cases} R_{k+1} < d_{k,k+1} \leq R_k, \quad \text{where } 1 \leq k \leq m-2; \\ d_{1m} \leq R_1, \quad d_{m1} \leq R_m; \\ d_{k,(k+j \mod m)} > R_k, \quad \text{where } 2 \leq j \leq m-1, \ 1 \leq k \leq m. \end{cases}$$

4. THE A^{4}LP PROTOCOL

The A^4LP protocol consists of an initialization phase when each node discovers its In-, In/Out-, and Out-bound neighbors, a path discovery phase using m-limited forwarding, and a path maintenance phase.

Information Maintained by a Node and Packet Types. A node $i \in \mathcal{N}$ maintains several data structures, a routing table (see Table 1), a path request sequence number and a node sequence number. The information in different packet types used by the A^4LP protocol is summarized in Table 2, and abbreviations can be found in Table 3.

1. *Routing Table at node j (RT_j)*: caches information for all neighbors and for most recently used destination (Table 1). The field *dstNeighborType* takes one of the following values: In-bound, Out-bound, In/Out-bound, or Not-neighbor. *expTime* records the expiration time for an entry after which it is no longer valid.

2. *Request Sequence Number* (`reqSeq`): a counter, uniquely identifies a path request packet sent by the the node with nodeId. The `reqSeq` is incremented every time a route request is sent.

3. *Node Sequence Number* (`seq`): a counter revealing the freshness of a node, incremented when the node detects the change of location, residual power, transmission range, routing table, and so on.

Table 1: The fields of a routing table

Field	Description
dstId	Destination node id
dstLoc	Destination location information
dstClass	Destination class
dstPow	Destination residual power
dstRange	Destination transmission range
dstSeq	Destination sequence number
dstNeighborType	Neighbor type of destination
nextHop	Next hop to forward a packet
expTime	Expiration time

Table 2: The fields of packet types used in the A^4LP protocol

Packet Type	Fields		
Hello	srcId srcClass	srcLoc srcRange	srcPow srcSeq
Convey	srcId lowLoc lowRange	dstId lowClass lowSeq	lowId lowPow
Update	srcId srcClass	srcLoc srcRange	srcPow srcSeq
PErr	srcId	dstId	srcSeq
FwdReq	srcId dstId reqSeq weakHops	srcLoc dstLoc cutoff powCons	srcSeq dstSeq fwdPath
BackReq	srcId dstId reqSeq backPath	srcLoc dstLoc cutoff weakHops	srcSeq dstSeq fwdPath powCons
FwdReply	srcId backPath	dstId	fwdPath
BackReply	srcId	dstId	backPath
FwdReqAck	srcId	dstId	fwdPath
BackReqAck	srcId	dstId	

Table 3: The abbreviations used in A^4LP

Field	Description
srcId/dstId/lowId	source/destination/L-node node id
srcLoc/dstLoc/lowLoc	source/destination/L-node location
srcClass/lowClass	class of source/L-node
srcPow/lowPow	residual power of source/L-node
srcRange/lowRange	range of the source/L-node
srcSeq/dstSeq/lowSeq	seq of source/destination/L-node
reqSeq	path request sequence number
cutoff	forward cutoff
fwdPath/backPath	forward/backward path
powCons	power consumed per packet
weakHops	number of weak hops

4.1 Neighbor Discovery Process

In-bound Neighbor Discovery Process.

The In-bound neighbor discovery (which, incidentally, leads also to the discovery of neighbors which will later turn out to be In/Out-bound) is initiated when a node joins the network. Each node broadcasts periodically a Hello packet to inform all the neighbors in its range of its current location, residual power, and transmission range. The time between two such transmissions is called a *hello interval*.

Upon receiving a Hello packet a node either updates an existing entry in its routing table or creates a new one. Acknowledgements are not required (actually not possible for In-bound neighbors). A node deletes the entries of Out- and In/Out-bound neighbors if it does not receive their Hello packets for several hello intervals. A Hello packet is a broadcast packet with a life time of one hop. The Hello packet provides the location, the class, the residual power, and the range of the sender, see Table 2.

Out-bound Neighbor Discovery Process.

Due to the nature of asymmetric links, Out-bound neighbors are not detected directly as their signals cannot be heard. For example, in the three-party proxy set in Figure 3(c), the Hello packet from node j cannot reach node i, thus node i cannot know that node j is an Out-bound neighbor. However, node k, which is an In/Out-bound neighbor of node i and an Out-bound neighbor of node j, is aware that link L_{ij} is asymmetric with i as the H-node and j as the L-node. Thus, node k sends a Convey packet to node i with the information of node j, and, at the same time, records node i as the next hop to reach node j.

In the Out-bound discovery, a node periodically checks the link relationship between its neighbors, sets up the route to its In-bound neighbor if a three-party proxy set is detected, and informs the H-node of an asymmetric link, when it detects one. The time between two Out-bound neighbor discovery is called a *convey interval*. The Convey packet contains the Id of the sender and of the destination (the H-node of an asymmetric link), the Id, the location, the class, the residual power, the range, and the sequence number of the L-node of the asymmetric link, see Table 2.

4.2 Location and Power Update

Dissemination of the approximate node location as well as its residual power are critical for any location-aware and power-aware routing scheme, yet it is not the focus of this paper. It can be achieved by (i) gossiping algorithms, (ii) a broadcast scheme, in which updates are sent infrequently and locally. (iii) a hierarchical scheme - nodes form clusters around *head of a cluster*, who covers a relative large area and is able to exchange information collected from members of the cluster, or some other scheme.

In A^4LP a node sends location and power updates only when (i) it joins the network, (ii) has moved significantly since the last reported location, (iii) its residual power goes below *low water mark*.

4.3 m-Limited Forwarding

Flooding, a common technique to disseminate information in traditional networks, is not suitable for power-constrained ad hoc networks because it requires excessive bandwidth and high power consumption. Our proposed m-limited forwarding is a technique to reduce the cost of disseminating information in a power-constrained environment by limiting the cardinality of the subset of nodes that will retransmit a packet. The path discovery in A^4LP is based upon m-limited forwarding.

In case of flooding, when node j transmits a packet at time t, all its Out- and In/Out-bound neighbors, the nodes in the set $H_j(t) = Out_j(t) \cup InOut_j(t)$ will retransmit the packet. We wish to limit the size of the subset of nodes that forward the packet to at most $m <\mid H_j(t) \mid$ in general case. Moreover, the nodes in this subset $F_j(t) \subset H_j(t)$ should be the ones optimally positioned vis-a-vis the packet destination and with the most favorable power level.

The sender of the packet, node j should provide a “hint”, which is called a *forwarding cutoff* $\kappa_j(t)$, and broadcast it together with the original packet. Each node $k \in H_j(t)$ is able to determine whether it belongs to the selected subset, $k \in F_j(t)$, by evaluating a *forwarding fitness function*, $\eta_k(t)$ and comparing the value of this function with the *forwarding cutoff*. Node k should forward the packet if and only if $\eta_k(t) \geq \kappa_j(t)$. If the location of the destination is unknown, the sender sets $\kappa_j(t) = -1$ and all nodes in H_j will retransmit the packet. If $\mid F_j(t) \mid < m$ then $\kappa_j(t) = 0$ and in this case individual nodes in H_j make their own decision whether to forward or not. The destination will recognize its own id and does not forward the packet further.

4.4 Forwarding Fitness Function

The path selection for the A^4LP protocol is based on the following principles. First, the forwarding process should favor nodes with a high level of residual power. A path from the source to the destination including only nodes operating at or above *normal-power level* (i) consists of fewer hops, (ii) has a lower latency, (iii) is less likely to fail, and (iv) has a greater change to be available and reused in the future. Second, we should avoid nodes which are about to deplete their power reserves. When a node in the *cut set* (a minimal set of nodes whose removal will cause the network to partition) becomes inoperable, the network is partitioned and works abnormally. Third, if we know the location of the source and the destination we should choose intermediate nodes optimally positioned around the straight line connecting the source with the destination.

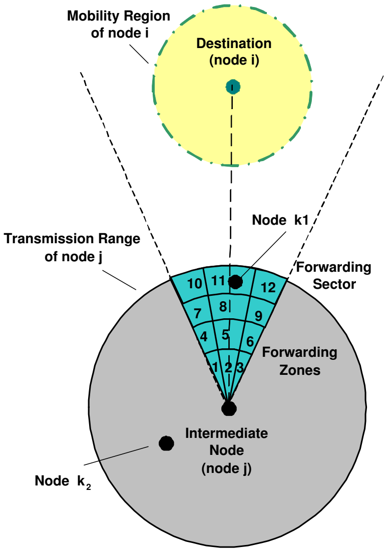


Figure 4: Forwarding sector and forwarding zones. Shown are the transmission range of the sender j, the destination i and its mobility region, and two candidates k_1 and k_2 as the next hop. The *forwarding sector* is a symmetric sector with an angle 2α with the axis of symmetry the straight line connecting nodes i and j. A set of *forwarding zones* are illustrated.

Let us now discuss some geometric considerations to locate the optimal next hop in path selection. In Figure 4, the shaded region defines the *forwarding sector* of node j for a packet with node i as its destination. A forwarding sector is a symmetric sector centered at the current location of the node and with a radius of the transmission range of the node. The axis of the forwarding sector is the straight line connecting nodes i and j and the angle of the sector is 2α, in which α is a function of the mobility region of the destination node and the density in the vicinity. The next hop on the optimal path is more likely within the forwarding sector. Moreover, a forwarding sector can be divided into a set of *forwarding zones* according to the distance to the sender and the angle relative to the destination, see Figure 4 with 12 forwarding zones. The optimal forwarding zones are the farthest from the sender and the closest to straight line from the sender and the destination. In our example the forwarding zone labeled 11 seems to be the best one.

For the case illustrated in Figure 4 we expect node k_1 to be a "better" choice than node k_2 as the next hop on the path from j to i, based solely on their locations. However, this may not be true when residual powers (or transmission ranges) of the two nodes are considered.

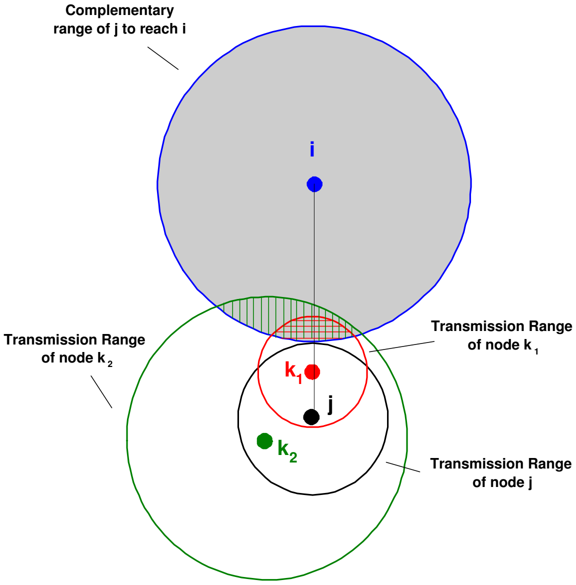

Complementary range of j to reach i
i
Transmission Range of node k_2
Transmission Range of node k_1
k_1
j
k_2
Transmission Range of node j

Figure 5: A configuration including the sender j, the destination i, and two candidates nodes k_1 and k_2 as the next hop on the path from j to i. The circles centered at the current location with the radius equal to the range of these nodes, Π_j, Π_{k_1}, and Π_{k_2} are also shown. The circle centered at the current position of node i and with a radius equal to $d_{ij} - R_j$ is called the "complementary range of j to reach i" and denoted by $\Gamma_{i,j}$. Call $\gamma_{i,j,k}$ the area of the intersection of $\Gamma_{i,j}$ and Π_k. We see that $\gamma_{i,j,k_1} < \gamma_{i,j,k_2}$.

A *forwarding fitness function*, $\eta_k(i,j,t)$, measures the fitness of node k as the next hop on the path from node j to node i. Different heuristics can be used when design a forwarding fitness function. In Figure 5 we see a configuration including the sender j, the destination i, and two candidates nodes k_1 and k_2 as the next hop on the path from j to i. The circles centered at the current location with the radius equal to the range of these nodes, Π_j, Π_{k_1}, and Π_{k_2} are also shown. The circle centered at the current position of node i and with a radius equal to $d_{ij} - R_j$ is called the "complementary range of j to reach i" and denoted by $\Gamma_{i,j}$. Call $\gamma_{i,j,k}$ the area of the intersection of $\Gamma_{i,j}$ and Π_k. In our example in Figures 4 and 5, $\gamma_{i,j,k_1} < \gamma_{i,j,k_2}$.

If we assume that the number of nodes in a given area is proportional to the size of the area, the number of nodes that could be potentially the next hop on the path to the destination is larger for node k_2 than those for node k_1.

We define the forwarding fitness function of node k as the next hop on the path from node j to node i to maximize the area $\gamma_{i,j,k}$ at time t as follows:

$$\eta_k(i,j,t) = \gamma_{i,j,k}(t) \times \beta(C_k)$$

with

$$\gamma_{i,j,k}(t) = \begin{cases} -1 & \text{if } Loc_i(t) \text{ unknown to } j; \\ 0 & \text{if } R_k(t) \leq d_{ik}(t) - r_{ij}(t); \\ r_{ij}^2(t)(\varphi - sin\varphi)/2 & \text{if } \ d_{ki}(t) - r_{ij}(t) < R_k(t) \\ +R_k^2(t)(\theta - sin\theta)/2 & \ < d_{ki}(t) + r_{ij}(t); \\ \pi \cdot r_{ij}^2(t) & \text{if } R_k(t) \geq d_{ki}(t) + r_{ij}(t). \end{cases}$$

where $r_{ij}(t) = d_{ij}(t) - R_j(t)$, $\theta = 2\arccos(\frac{R_k^2(t)+d_{ik}^2(t)-r_{ij}^2(t)}{2\cdot R_k(t)\cdot d_{ki}(t)})$, and $\varphi = 2\arccos(\frac{r_{ij}^2(t)+d_{ik}^2(t)-R_k^2(t)}{2\cdot r_{ij}(t)\cdot d_{ik}(t)})$, and

$$\beta(C_k) = \begin{cases} 1.5 & \text{if } C_k \equiv C1 \\ 1.0 & \text{if } C_k \equiv C2 \\ 0.5 & \text{if } C_k \equiv C3 \\ 0.1 & \text{if } C_k \equiv C4 \end{cases}$$

$\gamma_{i,j,k}$ favors nodes with a better chance to find optimally placed neighbors; the factor β favors nodes of higher class and lowers the chance of nodes of lower class as the next hop on the path from the current node to the destination. The *forwarding cutoff* is set as the m-th largest fitness value.

4.5 Path Discovery

When the sender does not have a route to the destination, it initiates *path discovery*. Traditional reactive ad hoc protocols consider only symmetric links, thus, the *forward/backward path* from source/destination to destination/source consists of the same set of nodes, but in reverse order. Once a path is found as a result of a *path discovery*, an acknowledgment is sent to the node originating the request.

In A^4LP, the forward and backward path are not necessary to be the same, thus we need four phases to find and confirm both paths: *forward path request*, *backward path request*, *forward path reply*, and *backward path reply*. Moreover, we need two additional phases, namely, *forward path request acknowledgement* and *forward path reply acknowledgement*, if the destination has a current route to the source, as can be seen in Figure 6. The fields of each packet type and their abbreviations can be found in Tables 2 and 3.

Forward Path Request. The source initiates a forward path request (`FwdReq`) packet, and uses *m-limited forwarding* to send this packet to its Out- and In/Out-bound neighbors. *fwdPath* accumulates the sequence of hops taken by route request packet as it travels through the network. Each intermediate node appends its node id onto *forward path* before forwarding the `FwdReq` packet. *cutoff* is the *forwarding cutoff* as described in section 4.3. *powCons* accumulates the power consumption rate of hops taken by route request packet. Both transmitting and receiving power rate are counted. *weakHops* accumulates the number of hops who operate at *low-power level*, which is an indicator of the path quality. Upon receiving a `FwdReq` packet, an intermediate node synchronizes information of the source and the destination according to their sequence numbers, such that both the packet and the intermediate node contain up-to-date information about the source and the destination. If the intermediate node is allowed to forward the packet, *fwdPath*, *powCons*, *weakHops* may need to be modified. The destination node continues to accept `FwdReq` packets for a given time interval, after which it chooses a forward path from the set of received `FwdReq` packets based on the power consumed per packet, arrival time, and quality of path. The destination initiates either the forward path request acknowledgement phase if it has a valid route to the source, or the backward path request phase otherwise.

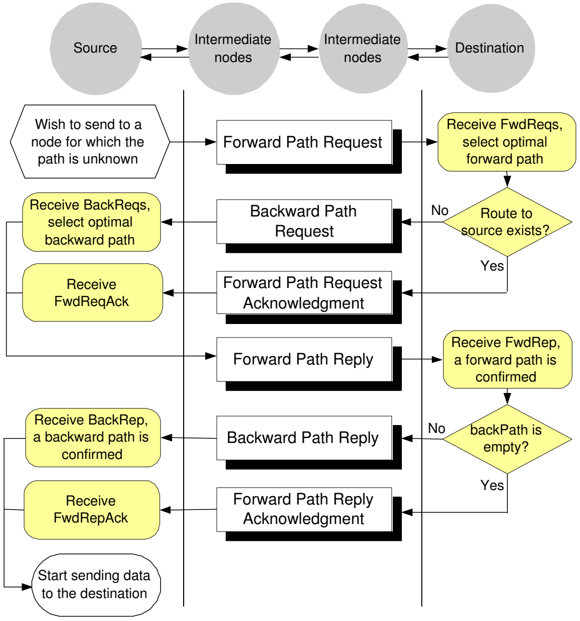

Source
Intermediate nodes
Intermediate nodes
Destination
Wish to send to a node for which the path is unknown
Forward Path Request
Receive FwdReqs, select optimal forward path
Receive BackReqs, select optimal backward path
Backward Path Request
No
Route to source exists?
Yes
Receive FwdReqAck
Forward Path Request Acknowledgment
Forward Path Reply
Receive FwdRep, a forward path is confirmed
Receive BackRep, a backward path is confirmed
Backward Path Reply
No
backPath is empty?
Yes
Receive FwdRepAck
Forward Path Reply Acknowledgment
Start sending data to the destination

Figure 6: The path discovery in A^4LP involves the following phases: Forward Path Request, Backward Path Request, Forward Path Reply, Backward Path Reply, Forward Path Request Acknowledgement, and Forward Path Reply Acknowledgement.

Forward Path Request Acknowledgement. The destination initiates a forward path request acknowledgement phase when the destination receives a `FwdReq` packet and has a route to the destination. The forward path is piggybacked into a forward path request acknowledgement (`FwdReqAck`) packet. When the sender receives this packet, the forward path reply phase is triggered.

Backward Path Request. The destination initiates a backward path request (`BackReq`) packet, and uses *m-limited forwarding* to send the packet to its Out-, In/Out-bound neighbors, with the forward path piggybacked. Intermediate nodes handle a `BackReq` packet similar to a `FwdReq` packet. The source chooses the best backward path based on the `BackReq`s received in a given interval, and initiates the forward path reply phase to confirm the forward path.

Forward Path Reply. The source initiates a forward path reply phase by unicasting a *forward path reply* (`FwdReply`) packet along the forward path to the destination. If the forward path reply phase is triggered by a `BackReq` packet, the source piggybacks the backward path. *backPath* is set to be empty if the forward path reply phase is initiated by a

FwdReqAck packet. A route to the destination is established during the traversal of a FwdReply packet. Each node on the forward path updates its next hop neighbor towards the destination in the *fwdPath* field of the FwdReply packet. To get the maximum benefit from the forward path, sub-paths can be established between intermediate nodes. For instance, suppose an intermediate node N_i receives a FwdReply packet with the forward path as $[N_0, N_1, \cdots, N_i, N_{i+1}, \cdots, N_k]$ where node N_0 is the source and node N_k is the destination. Node N_i could establish route to $N_{i+1}, N_{i+2}, \cdots, N_k$ with next hop as N_{i+1}. The destination initiates either the backward path reply phase if *backPath* is not empty, or initiates the forward path reply acknowledgement phase otherwise.

Backward Path Reply. The destination initiates a backward path reply phase by unicasting a *backward path reply* (BackReply) packet along the backward path to the source. A route to the source is established during the traversal of a BackReply packet. All intermediate nodes handle a BackReply packet similar to a FwdReply packet. When the source receives a BackReply packet, the path discovery process is completed successfully. Thus, the source is capable of exchanging data packets with the destination.

Forward Path Reply Acknowledgement. The destination initiates a forward path reply acknowledgement phase when it receives a FwdReply packet in which *backPath* field is empty, and unicasts a forward path reply acknowledgement (FwdReplyAck) packet to the source. The source is informed the end of the path discovery phase by the *FwdReplyAck* packet and the transmission of data packets are triggered.

4.6 Path Maintenance

Movement of nodes lying along an active path may cause a route to become invalid. In case a route becomes invalid, at least one of the links on the route fails. A link ($L_{i,j}$) failure could be detected, if all attempts to forward a packet from node i to the next hop j fails. All packets at a failed link will be discarded. The link failure has to be reported to the source to avoid massive retransmissions.

If the source detects the link from it to the next hop becoming unreachable, it disables the routing entry to the destination, and initiates a path discovery process to recover the route to the destination. If the link failure happens at an intermediate node, it reports to the source by sending a path error (PErr) packet. The source node recovers the route to the destination by initiating a path discovery process.

5. SUMMARY AND FUTURE WORK

In this paper we introduce a power- and location-aware routing protocol for ad hoc networks with asymmetric links. We recognize three types of neighbors of a node, In-, Out-, and In/Out-bound. We introduce the concept of an m-party proxy set as a subset of nodes that can reach each other either directly or through a subset of members. We then discuss the topological constraints for ad hoc networks with asymmetric links. Then we discuss forward and backward discovery phases based upon m-limited forwarding. We use a forwarding fitness function and a forwarding cutoff to allow individual nodes receiving a broadcast packet to decide if they will forward it or not. We then describe the protocol.

We are currently using the NS2 simulator to investigate various forwarding fitness functions and the performance of the A^4LP protocol.

6. ACKNOWLEDGEMENTS

The research reported in this paper was partially supported by National Science Foundation grants MCB9527131, DBI0296035, ACI0296035, and EIA0296179

7. REFERENCES

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). *ACM MOBICOM '98*, 1998.

[2] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *Proceedings of IEEE SICON'97*, pages 197–211, 1997.

[3] G. Dommety and R. Jain. Potential networking applications of global positioning systems (GPS). *Technical Report TR-24*, Ohio State University, 1996.

[4] Z. Haas and M. Pearlman. The zone routing protocol (zrp) for ad hoc networks. *Internet Draft*, 1998.

[5] Z. J. Haas, M. R. Pearlman, and P. Samar. The interzone routing protocol (ierp). *Internet Draft, draft-ietf-manet-zone-ierp-02.txt*, 2002.

[6] Z. J. Haas, M. R. Pearlman, and P. Samar. The intrazone routing protocol (iarp). *Internet Draft, draft-ietf-manet-zone-iarp-02.txt*, 2002.

[7] P. Jacquet, P. Muhlethaler, and A. Qayyum. Optimized link state routing protocol. *IETF MANET, Internet Draft*, 1998.

[8] D. Johnson and D. Maltz. Dynamic source routing in ad hocwireless networks. *Kluwer Academic*, 1996.

[9] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7:343–358, 2001.

[10] Y. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *ACM/Baltzer Wireless Networks (WINET) journal*, 6-4, 2000.

[11] M. Maleki, K. Dantu, and M. Pedram. Power-aware source routing protocol for mobile ad hoc networks. *ISLPED '02*, August 2002.

[12] D. C. Marinescu, G. M. Marinescu, Y. Ji, L. Bölöni, and H. J. Siegel. Ad hoc grids: Communication and computing in a power constrained environment. *Workshop on Energy-Efficient Wireless Communications and Networks (EWCN)*, 2003.

[13] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. *Proceedings of INFOCOM '97*, 1997.

[14] C. E. Pekins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *ACM SIGCOMM '94*, pages 234–244, 1994.

[15] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 99–100, 1999.

[16] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. *ACM MOBICOM '98*, 1998.

[17] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. *ACM MOBICOM '01*, 2001.