CERIAS Tech Report 2005-143

A02P: Ad Hoc On-Demand Position-Based Private Routing Protocol

by X Wu, B Bhargava
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

AO2P: *A*d Hoc *O*n-Demand *P*osition-Based *P*rivate Routing Protocol

Xiaoxin Wu and Bharat Bhargava, *Fellow*, *IEEE*

Abstract—Privacy is needed in ad hoc networks. An ad hoc on-demand position-based private routing algorithm, called AO2P, is proposed for communication anonymity. Only the position of the destination is exposed in the network for route discovery. To discover routes with the limited routing information, a receiver contention scheme is designed for determining the next hop. Pseudo identifiers are used for data packet delivery after a route is established. Real identities (IDs) for the source nodes, the destination nodes, and the forwarding nodes in the end-to-end connections are kept private. Anonymity for a destination relies on the difficulty of matching a geographic position to a real node ID. This can be enforced by the use of secure position service systems. Node mobility enhances destination anonymity by making the match of a node ID with a position momentary. To further improve destination privacy, R-AO2P is proposed. In this protocol, the position of a reference point, instead of the position of the destination, is used for route discovery. Analytical models are developed for evaluating the delay in route discovery and the probability of route discovery failure. A simulator based on ns-2 is developed for evaluating network throughput. Analysis and simulation results show that, while AO2P preserves communication privacy in ad hoc networks, its routing performance is comparable with other position-based routing algorithms.

Index Terms—Ad hoc routing protocol, anonymity, communication privacy, channel access mechanism.

1 Introduction

Protecting personal privacy is a prime concern for the emerging pervasive systems. As an important part of privacy, the user anonymity can improve security by making it difficult for adversaries to trace their potential victims and to conduct target-specific attacks. Achieving node privacy is challenging in ad hoc networks, where routing schemes rely on the cooperation and information exchange among the nodes. In routing algorithms such as AODV [1], DSR [2], and DSDV [3], a node has to disclose its identity (ID) in the network for building a route. Node activities, such as sending or receiving data, are highly traceable and, consequently, nodes are vulnerable to attacks and disruptions.

The privacy preservation approaches in the literature do not directly extend to ad hoc networks. The use of broadcast [4] or multicast [5] for receiver privacy are not suitable; as in ad hoc networks, the bandwidth is limited, and multicast itself is a challenging problem [6]. The K-anonymity algorithm [7] achieves anonymity by keeping the entity of interest within a group. Yet it is not easy to maintain such a group with a fixed proxy in an ad hoc network due to the node mobility and the continuous join-and-leave activities. The dynamic nature also makes it difficult to use the anonymity solutions based on trusted third parties [8]. In approaches applying the onion structure [9], [10], where anonymity can be realized in a multihop path by keeping each node along the path aware of only its previous hop and next hop, the cost of using public keys is high.

• *The authors are with the Department of Computer Sciences, 250 N. University St., Purdue University, West Lafayette, IN 47906. E-mail: {wu, bb}@cs.purdue.edu.*

Manuscript received 23 July 2004; revised 8 Nov. 2004; accepted 15 Nov. 2004; published online 27 May 2005.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0234-0704.

Geographic or position-based routing algorithms for ad hoc networks have been widely studied [11]. In addition to node ID, extra information, such as the positions of the nodes, is used for making routing decisions. Since it is unlikely that two ad hoc nodes are concurrently at exactly the same position, the match between a position and an ID is unique. Therefore, in position-based routing algorithms, if the positions have been exposed for routing, node IDs do not need to be revealed. If an adversary cannot match a position to a node ID correctly, node anonymity can be achieved.

However, using position instead of ID for route management in traditional positioning routing algorithms does not guarantee node anonymity. These algorithms rely on the position exchange among the neighboring nodes. A previous hop knows the positions of its neighbors, so that it can select the next hop that is the closest to the destination. Such an information exchange is normally through a periodic message that is locally broadcast by each node. The message is called a "hello" message and carries an updated position of the sender. These time-based position reports make a node highly traceable. An eavesdropper can determine whether the "hello" messages are from the same node based on the time they are sent out. The trajectory of a node movement can be well-known to other nodes even when its ID is intentionally hidden. It is much easier to obtain a node ID based on its trajectory. Furthermore, if a tracer has determined the node ID correctly, it can always stay close to this node and monitor its behavior. The transmission jitter for "hello" messages may make tracing a little more difficult, yet it is not sufficient to protect a node's trajectory from being discovered.

Lack of privacy in traditional positioning ad hoc routing algorithms is mainly caused by the extensive position information exposure. To achieve communication anonymity, a position-based ad hoc routing algorithm, named

AO2P, is proposed. AO2P works in the network with relatively high node densities, where the positions of destinations are the only position information disclosed in the network for routing. In AO2P, a route is discovered by delivering a routing request message from the source towards the position of the destination. However, AO2P does not rely on the local position information exchange. To determine the next hop with limited position information, an approach similar to Anycast [12] is developed, which relies on a proposed *receiver contention* channel access mechanism.

In AO2P, once a previous hop sends out a routing request, its neighboring nodes who receive the request will contend to access the channel to be the next hop. In the receiver contention mechanism, receiving nodes are divided into different classes according to how close they can bring the routing request toward the destination. A receiver geographically closer to the destination is assigned to a class with a higher priority, and it generally can win the contention. This results in the routes with a lower number of hops. Fewer forwarders are needed and, hence, the ad hoc channel is shared by fewer nodes. In a network with a fixed data rate, these routes generally have a better routing performance.

Once a route is built, pseudo IDs and temporary MAC addresses are used for the nodes in the routes, such as sources, destinations, and intermediate forwarders. Since the node identities are not disclosed, communication anonymity can be achieved. For a destination whose position is revealed, its privacy is preserved by hiding the match between a position and its ID through the secure position management scheme. Eavesdroppers or attackers only know that a node at a certain position will receive data, but they do not know which node it is. On the other hand, the routing accuracy is guaranteed because at most one node can be at a specific geographic position at one time. The position and the time are used as the inputs of the hash function, which generates a node's pseudo ID. The possibility that two nodes have the same pseudo ID so that data may be delivered to the wrong node is negligible.

AO2P mitigates the attacks on node anonymity from both external and internal attackers with the assistance of secure position services. Node authentication and encryption can prevent an external attacker from learning a node's position. For an internal attacker, position management policy will be enforced so that a node cannot abuse position information for tracing purposes. Such an attacker may obtain fractions of position information of its target. However, the information is incomplete and will not be enough for attacker to trace a moving target.

The contributions of this paper are the design and routing performance evaluation for the proposed anonymous positioning routing algorithm. We build analytical models for evaluating the performance metrics, such as the delay in route discovery and the probability of a route discovery failure with or without position errors. We use simulation to evaluate performance metrics, such as the impact of destination mobility, the hop counts in the discovered routes, and the network end-to-end throughput. Node anonymity can be evaluated in terms of the size of anonymity set [13], probability [10], [14], and entropy [15]. This is our on-going research and will be presented in a future paper.

The rest of the paper is organized as follows: Section 2 briefly presents related research. In Section 3, the details of the routing algorithm are presented. In Section 4, the delay for the routing discovery and the probability of routing discovery failure are analyzed. Section 5 shows analysis and simulation results. Section 6 provide the conclusions and future works.

2 Related Research

Anonymous communication in ad hoc networks has been studied in [16]. A novel untraceable on-demand routing protocol, named ANODR, is proposed. Onion structure is used for routing discovery. To reduce the cost and latency of the encryption/decryption, a symmetric key based *Boomerang Onions* is used. Once a route is found, pseudorandom numbers are used as temporary IDs for the nodes along the route. Each node only knows the pseudo numbers from its previous hop and next hop. The communication privacy is achieved because real IDs are not revealed. The protocol is robust to intrusion since the intrusion in a single node en route does not result in ID exposure.

A position-based ad hoc routing algorithm, named a greedy perimeter stateless routing (GPSR), is presented in [17]. A packet is always forwarded to the next hop that is geographically closest to the destination. Such an approach is scalable since it does not need route discovery and maintenance, and the position information is exchanged locally among neighbor nodes by periodically sending out a beacon. GPSR may not always find the optimum route. When nodes are not uniformly distributed in the network, there will be dead ends, in which a node cannot find any next hop closer to the destination. GPSR solves this problem by routing around the perimeter of its local region. Other approaches to solving the dead end problem, named Face Routing and GFG (Greedy-Face-Greedy) schemes, are proposed in [18]. Some other delivery-guaranteed methods are based on the single-path strategy [19], [20]. A route from the source to the destination is built before data packets can be delivered.

The position-based routing algorithms depend on the position availability. It is assumed that a source is able to get the position of its destination. The Global Positioning System (GPS) helps a node to get its own position. How a source gets the position of its destination is a challenging task. In an ad hoc/cellular integrated environment [21], the position of a destination can be obtained through paging or the short message service through the cellular network. A source node sends a position request to the cellular network. The cellular network pages the destination. The destination replies with its position, which is forwarded to the source. This *out-of-band* solution is simple since it has little signaling overhead and operational complexity. When an out-of-band server is not available, *in-band* position servers are designed. In [22], each node has a geographical region around a fixed center. The region is called a virtual home region (VHR) and the ad hoc node updates its position information to all the nodes residing in its VHR.

The relationship between a node ID and the fixed center of its VHR follows a hash function, so that other nodes can acquire a node's position by sending request to the right VHR. A similar distributed position service system, named DLM (Distributed Location Management), is studied in [23].

3 AO2P Routing Algorithm

In this section, we first introduce a secure position service system that is necessary for privacy preservation in positioning ad hoc routing algorithms. We then describe the proposed anonymous routing algorithm, where the details on AO2P route discovery and maintenance are given. Next, we present a receiver classification scheme, followed with the receiver contention scheme. Based on these two schemes, AO2P can process efficient route discovery. Finally, an enhanced algorithm that further improves destination anonymity is given.

3.1 Position Management

We propose a virtual home region (VHR)-based distributed secure position service, named DISPOSER. An ad hoc node is assumed to be able to obtain its own geographic position through GPS. Each node has a VHR, which is a geographical region around a fixed center. The relationship between a node ID and its VHR center follows a hash function. This function is predefined and known to all the nodes who join the network. A number of servers, which are also ad hoc nodes, are distributed in the network. A node updates its position to the servers located in its VHR, to which other nodes send position request acquiring this node's position.

A node updates its position to its VHR when the distance between its current position and the last reported position exceeds a threshold value. The threshold value is determined by finding out that, if a destination moves a certain distance away from the position known to the source, what the probability is of a routing failure caused by this position drift. Simulation is done to study the relationship between this threshold value and the probability of a routing failure in Section 5.2.2. Since the positions of VHRs are known, position update and position request can use the AO2P routing algorithm. In this way, DISPOSER message delivery does not require a node to process a time-based position update to its neighboring nodes.

DISPOSER enhances position security. Only a small number of trusted nodes can act as position servers. To obtain the position of a certain node, a requester has to send a signed position request. The position information is encrypted and will not be learned by other users during the position management. Positions are used for routing only. A mechanism has been designed, which constrains a node to use position for route discovery only. After obtaining a node position, the node requester has to prove to the servers that it has built a meaningful communication with that node as its destination, normally by showing a ticket assigned by the destination. The position abuse when a node continuously sends position requests for tracing a target node is prevented. More details on DISPOSER security procedures are in [24].

When the source gets the position of its destination, it also gets the time when the position is updated and an authentication code. The time is needed for routing accuracy. The secret code can be a random number, which is generated and sent to the position server by the destination along with its position update. The authentication code is used for destination authentication in the AO2P route discovery stage.

3.2 AO2P Routing Protocol

In AO2P, a source discovers the route through the delivery of a routing request to its destination. A node en-route will generate a pseudo node ID and a temporary MAC address. Once a route is built up, data is forwarded from the source to the destination based on the pseudo IDs. This section gives the details on AO2P routing discovery. Other issues, such as data delivery, route maintenance, and pseudo ID management, are addressed.

Once a source needs to find the route to its destination, it first generates a pseudo ID and a temporary MAC address for itself through a globally defined hash function using its position and the current time as the inputs. Such a procedure makes the probability that two active nodes (i.e., nodes involved in routing) have the same ID and MAC address small and negligible. The source then sends out a *routing request* (*rreq*) message.

The *rreq* message carries the information needed for routing, such as the position of the destination and the distance from this source to the destination, as well as the source pseudo ID. Since it is possible that another node has updated the same position (yet at a different time) to the position servers, a destination challenge message is carried in the *rreq* to make sure that the right destination will be reached. This message is also a result of a hash function, of which the inputs are the position of the destination and the time at which this position is updated. *rreq* carries the challenge message instead of the time for less information revelation. *rreq* also carries a Time-to-Live (TTL) number that deals with the possible loop. TTL is the maximum number of the hops a *rreq* can be forwarded. A source node can estimate the TTL value according to the distance from the source to the destination and the radio transmission range for each hop.

The neighboring nodes around the source, called *receivers*, will receive the *rreq*. A receiver checks the destination challenge message to find out whether it is the destination. If not, a receiver assigns itself to a receiver class following the rules in Section 3.3. Each receiver uses a hash function to generate a pseudo ID and a temporary MAC address. The inputs of the hash function are the receiver's position and the time it receives the *rreq*. The receivers then contend for the wireless channel to send out a *hop reply* (*hrep*) message in a so-called *rreq* contention phase. Details of this receiver-contention mechanism are described in Section 3.4. The receiver who has successfully sent out the *hrep* will be the next hop. Its pseudo ID is carried in the *hrep*.

On receiving the *hrep*, the source replies with a *confirm* (*cnfm*) message. Its next hop replies to this message with an *ack*. Upon receiving the *ack*, the source saves the pseudo ID and the temporary MAC address of the next hop in its routing table.

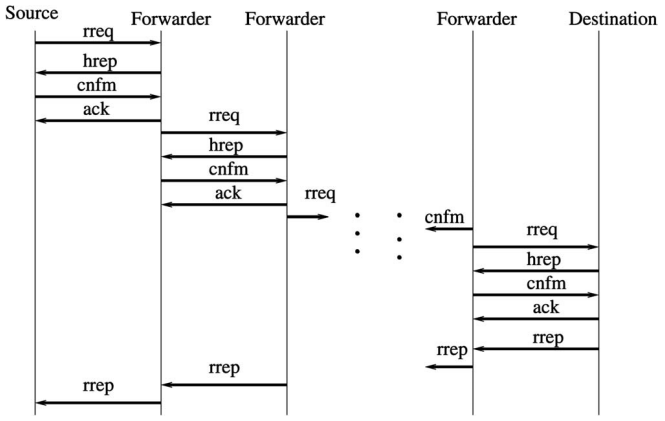


Fig. 1. Message flow in AO2P routing discovery.

After receiving the *cnfm*, the next-hop receiver becomes a *sender*.[1] It sends out the modified *rreq*, which carries the distance from itself to the destination. The TTL value is reduced by 1. Neighboring nodes around it will contend to be its next hop. Once the sender receives a *hrep*, it couples the pseudo ID and the temporary MAC address of its next hop with those of its previous hop and saves the pairs in the routing table.

The searching of the next hop is repeated until the destination receives the *rreq*. After identifying the destination challenge message, the destination sends out a *hrep*. Based on Section 3.4, it can always send out the *hrep* successfully. After receiving the *cnfm* from its previous hop, the destination sends a *routing reply* (*rrep*) message through the reverse path to the source. The destination also finds the corresponding authentication code according to the position carried in the *rreq* and encrypts the code with the secret key of its public/secret key pair. The encrypted result is included in the *rrep* and sent to the source. The source finds out whether it reaches the right destination by decrypting the information with the destination's public key and comparing the authentication code with the one it obtained through the position request.

The message flow in AO2P routing discovery is shown in Fig. 1. The frames for important control messages are shown in Fig. 2.

A route discovery failure will occur when a sender cannot find a legitimate next hop. Routing discovery failure may also be caused due to destination mobility. A typical case for this type of routing failure is that a *rreq* has been forwarded close to the position at which the destination was expected to be, yet the destination cannot receive the *rreq* because it has moved away. In both cases, a routing discovery failure report will be sent back to the source. The source will start a new route search based on the destination's most updated position after a backoff time.

After a route is built up, data packets are delivered following the pseudo ID and temporary MAC address pairs in the routing tables. Routing maintenance mechanisms in traditional ad hoc routing algorithms can be used for AO2P. When a route is broken, an *error* message will be sent back to the source by the node who has discovered the broken link. In AO2P, during the communication, the destination will update its new position to the source through the reverse route. The source can thus start a new routing discovery using the updated position information.

1. A *sender* is defined as the source or an intermediate node who forwards the *rreq* message.

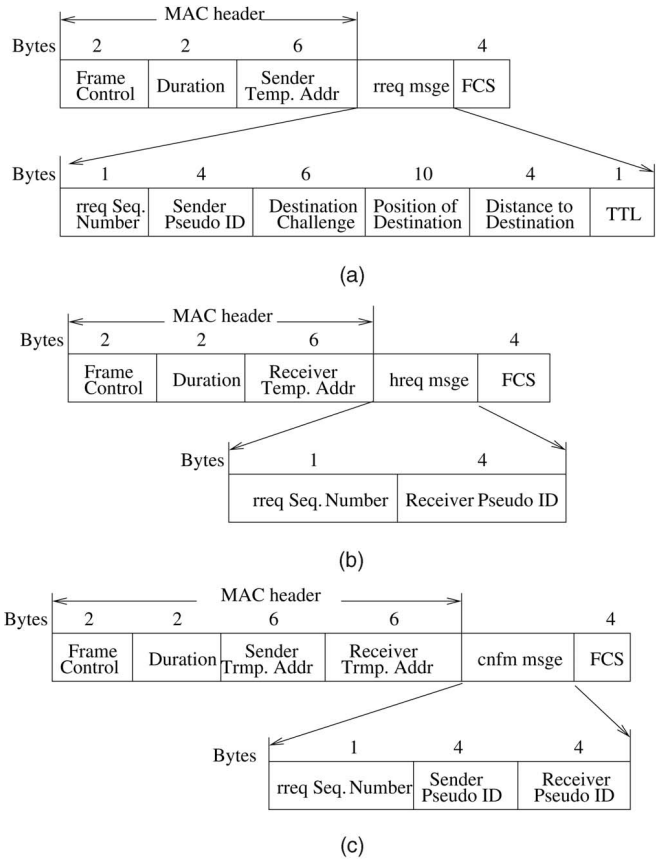


Fig. 2. Frames of control messages in AO2P routing discovery. (a) *rreq* frame. (b) *hrep* frame. (c) *cnfm* frame.

A node will generate a pair of a pseudo ID and a temporary MAC address only when it receives a *rreq*. If it wins the next hop contention and is included in the route, it will use the pair for data delivery. Otherwise, the ID and MAC address pair will be deleted. It is possible that a node is included in more than one route. In this case, only one pseudo ID and one temporary MAC address are used. A node deletes the pair of the pseudo ID and the temporary MAC address if the route in which the pair is used no longer exists. This happens when data delivery is finished, a routing *error* message is received, or the pair has not been used for a long time.

3.3 Receiver Classification

A receiver determines its node class by finding that, if it is the next hop, how much closer (this geographic distance is defined as Δd) it can move a *rreq* from the sender toward the destination. Δd can be calculated because the distance between the receiver and the destination is known based on their positions and the distance between the sender and the destination is carried in the *rreq*.

A simple illustrated example of node classification is shown in Fig. 3. In this example, all nodes except the destination are divided into four node classes. A distance of

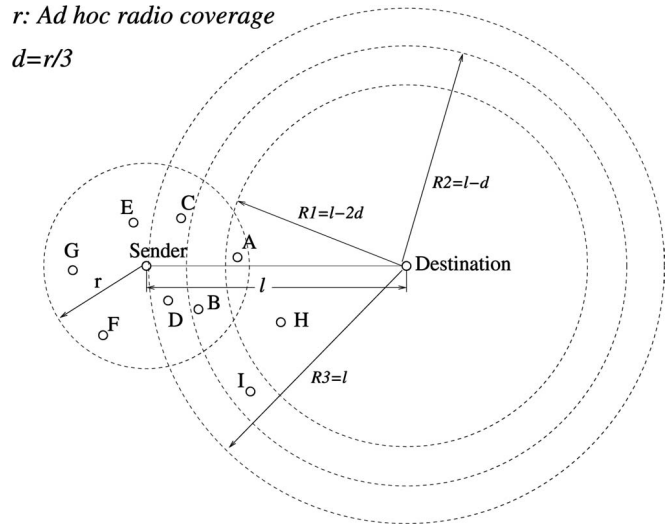


Fig. 3. Classifying nodes based on the positions.

d is calculated as $d = r/3$, where r is the maximum radio coverage of the ad hoc channel. Nodes with $\Delta d \geq 2d$ (e.g., node A, as it falls in the circle centered at the destination with a radius of $l - 2d$) belong to class 1, which has the highest priority. Nodes with $d \leq \Delta d < 2d$ (e.g., node B) and nodes with $0 \leq \Delta d < d$ (e.g., nodes C and D) belong to Class 2 and 3, respectively, and have lower priorities. For nodes E, F, and G, $\Delta d < 0$. They belong to class 4 and will lead the *rreq* away from the destination. Other nodes, such as H and I, are out of the sender's transmission range and cannot receive the *rreq*. Note that the destination is a special node. It has the highest priority to access the channel with a class of 0. In this paper, we investigated the algorithm in which only nodes of class 1, 2, and 3 will contend to be legitimate receivers. A node of class 4 will not attend the contention because it leads a *rreq* away from the destination.

The node classification scheme is used only for simplicity of presentation and will be used in the rest of the paper. In more complicated schemes, rules for node classification can be adaptive based on node density. When the density is high, only the nodes that can greatly reduce the distance between the *rreq* and the destination should be assigned to the class with a high priority. On the other hand, if the nodes are sparsely distributed, a node which leads the *rreq* away from the destination can also be a possible legitimate receiver. Such a rule adaptation, for example, can be made by adjusting the value of d. Besides the distance to the destination, other criteria, such as signal quality, the remaining power of a node, and node mobility, can also be considered in node classification.

3.4 AO2P *hrep* Contention Mechanism

The receiver-contention mechanism used in the *hrep* contention phase is *EY-NPMA* (Elimination Yield—Non-preemptive Priority Multiple Access), the channel access mechanism for HIPERLAN 1 [25], [26]. The main reasons for using *EY-NPMA* for *hrep* contention are: 1) *EY-NPMA* is a class-based channel access mechanism, while, in AO2P, receivers are divided into different classes, 2) the probability of a successful transmission for *EY-NPMA* is very high even when there are a large number of contending nodes, and 3) *EY-NPMA* has been widely used and tested. For better understanding of the entire AO2P algorithm and performance analysis, this section gives the details of the *hrep* contention mechanism.

Like *EY-NPMA*, the *hrep* contention phase of AO2P is further divided into three phases: the prioritization phase, the elimination phase, and the yield phase.

The prioritization phase starts a synchronization interval after receiving the *rreq*. It allows only the receivers with the highest channel access priority among the contending ones to participate in the next phase. A number of slots, the same as the number of different priority classes, are reserved for this phase. A receiver with a class of c can send a burst in slot c only if no burst is sent in the previous $c-1$ prioritization slots. This also means that it has the highest priority in this contending cycle. This receiver will then enter the next phase. If a receiver senses a burst in any of the previous slots, it will quit from *hrep* contention. In AO2P, the receivers that cannot enter the next phase will drop the *rreq*. The first slot of the prioritization phase is reserved for destination, which is called *Destination Acknowledgment Slot*. Only a destination can send a burst on this slot. In this way, the destination receiving a *rreq* can always have access to the channel successfully.

The elimination phase starts immediately after the transmission of the prioritization burst and consists of a number of slots. An AO2P receiver who enters this phase will transmit burst in a randomly selected number of continuous slots, starting from the first one in this phase. The receivers transmitting the longest series of bursts will survive. After the end of the burst transmission, each receiver senses the channel for the duration of the elimination survival verification slot. If the channel is sensed to be idle, the receiver is admitted to the yield phase; otherwise, it drops itself from contention. The length of the burst follows a truncated geometric probability distributed function. A transmission parameter, P_E, is used to adjust the burst length. Let m_{ES} to be the number of overall elimination slots and $P_e(n)$ be the probability that the burst is transmitted in the consecutive n slots. $P_e(n)$ is specified as:

$$P_e(n) = \begin{cases} (1 - P_E)P_E^n & 0 \leq n < m_{ES}, \\ P_E^n & n = m_{ES}. \end{cases} \quad (1)$$

The yield phase starts immediately after the end of the elimination survival verification interval. Before transmitting a *hrep*, a receiver will yield for a number of slots from 0 to m_{YS} with equal likelihood. It listens to the channel and, if the channel is sensed idle during the yield listening interval, it will send out the *hrep*. Otherwise, the receiver loses contention and drops the *rreq*. Let $P_y(n)$ to be the probability that a receiver will wait for n slots before it sends out the *hrep*. Then,

$$P_y(n) = 1/(m_{YS} + 1), 0 \leq n \leq m_{YS}. \quad (2)$$

When more than one receiver sends out an *hrep* at the same time, a *hrep* collision occurs. In this case, the sender will resend the *rreq*. On the other hand, if the sender cannot hear any burst in the prioritization period, it means there

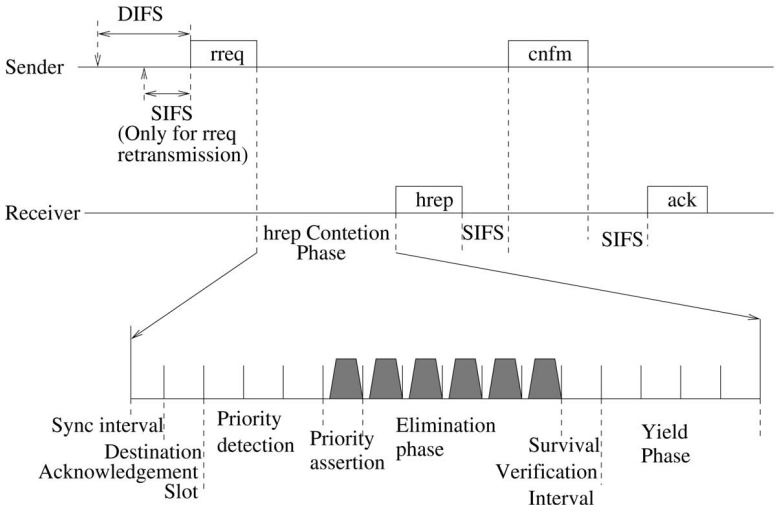


Fig. 4. Illustration of AO2P channel access mechanism in routing discovery.

are no potential next hops at that time. In that case, it will backoff and resend the *rreq* after a backoff time.

The complete channel access mechanism for AO2P routing discovery is summarized in Fig. 4. A traditional channel access mechanism such as CSMA/CA is considered to be used for a *rreq* or a data packet. Before a *rreq* message or a data packet is sent, the channel has to be sensed idle for a distributed interframe space (DIFS) of time. There is a short interframe space (SIFS) between message exchange for data packet forwarding and *rreq* forwarding, except for the time interval between a *rreq* and its *hrep*. This interval depends on the duration of the *hrep* contention phase. In AO2P, a *rreq* retransmission has a higher priority. The sender has to wait for only a SIFS before the retransmission. To avoid any other transmission during a *hrep* contention period, DIFS has to be longer than the longest idle period in the *hrep* contention phase. In AO2P, DIFS has to be longer than the entire yield phase.

3.5 Communication Anonymity and Privacy Enhancement

In AO2P, the identities for the two ends (source and destination) of a communication are anonymous to other nodes. AO2P also protects the privacy for nodes acting as intermediate forwarders, as they do not need to expose any information during data delivery. This is important for communication privacy in ad hoc networks. Unlike wired networks, in which a forwarder is normally a fixed router without the necessity to hide any information from others, in ad hoc networks, a forwarder is also a potential source or destination. The exposure of private information of a node during its action of forwarding may cause privacy loss in its previous or future communication sessions.

Destination has the lowest privacy because its position is revealed to the network for routing. Node movement can enhance the destination privacy because, if a node is mobile, the match between a position and the node ID is momentary. A single position release may not lead to severe privacy degradation.

Protecting destination position from adversaries can further improve destination anonymity. To hide this information from the eavesdroppers, a position of a *reference* point can be used in a *rreq* instead of the real position of the destination. The corresponding routing protocol is called AO2P with reference point, or R-AO2P. The reference point is on the extended line from the sender to the destination, as shown in Fig. 5a. The distance between the reference point and the destination is a large random value, based on which a tracer cannot estimate the real position of the destination. The node classification for receivers can use the similar rules as those in Section 3.3, with the difference that the class of a receiver is determined by how closer it can process a packet to the reference point. The rules for node classification based on destination and a reference point are compared in Fig. 5a and Fig. 5b. S_1, S_2, and S_3 are the areas where the nodes belonging to class 1, class 2, and class 3 are located. The solid lines in Fig. 5a and Fig. 5b are part of the circles centered at the reference point and the destination, respectively (referring to Fig. 3). Generally, a node closer to the reference point is also closer to the destination. Nodes at some special positions have the higher node class levels in R-AO2P than in AO2P. For example, node at position A has a class level of 3 in AO2P, yet it has a class level of 2 in R-AO2P. In R-AO2P, Node A has a better opportunity to win the *hrep* contention. The routes discovered by R-AO2P may then have larger hop counts. However, the hop count increase is not significant because nodes at most positions have the same class level. It should also be noted that, for R-AO2P, some nodes residing in S_3 may lead a packet away from the destination.

In R-AO2P, the next hop will obtain the position of the destination from the sender after it wins the *hrep* contention. The position is encrypted by a Diffie-Hellman key to keep it from being learned by other nodes. The Diffie-Hellman key, not the public key of the next hop, is used to prevent the identity exposure for the intermediate nodes. The Diffie-Hellman key is set up during *hrep* and *cnfm* exchange. *hrep* carries the initiation of the key and *cnfm* replies with the rest part of the key. *cnfm* also carries the encrypted position of the destination. After receiving the position of the destination, the next hop can generate a reference point at the extended line from itself to the destination and sends out a *rreq* carrying the position of the new reference point. The

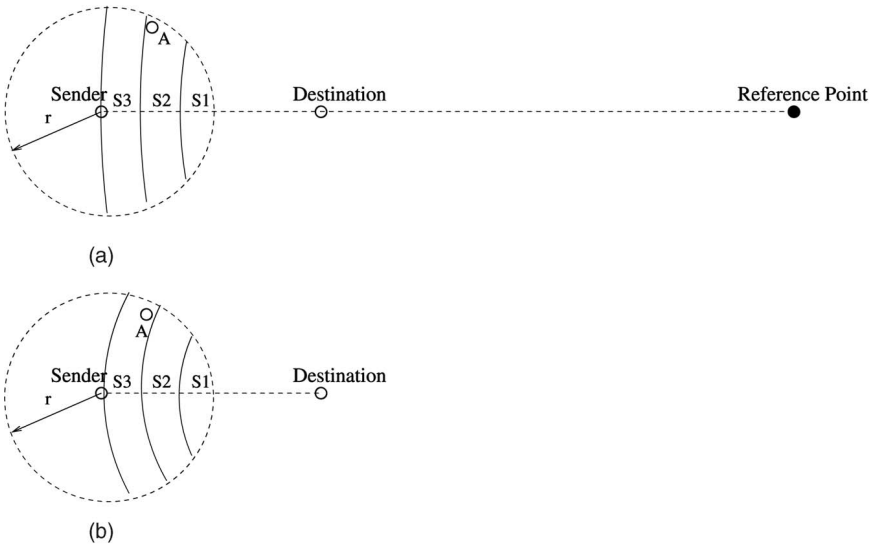


Fig. 5. Reference point and the receiver classification. (a) Node classification based on a reference point. (b) Node classification based on destination.

procedure is repeated until the *rreq* finally reaches the destination.

4 AO2P Performance Evaluation

In AO2P, a hop reply (*hrep*) contention period may cause extra delay in the route discovery. If such a delay is large, a routing failure or a route discovery failure may occur because the destination may have already moved away from the position known to the source. A route discovery failure may also be caused by inaccurate position information or the network topology where a next hop cannot be found. In this section, we first analyze the *hrep* average delay. Based on this delay, the average time needed for a successful next hop determination is calculated. We then present the analysis for the probability of a route discovery failure under different node distributions and position accuracy.

4.1 Delay for AO2P Next Hop Searching

The definitions of major symbols used in our analysis are listed in Table 1. We calculate $\overline{D}_{REQ}(n)$, the average time for next hop determination when there are initially n contenders.

Since we are considering a network with relatively high node density, for the simplicity of analysis, it can be assumed that, for a sender, neighbors belonging to the class with the highest priority are always available. Thus, in the *hrep* prioritization phase, the delay is approximately the time duration for two slots: the destination acknowledgment slot and the slot for the class with the highest priority. It is assumed that n receivers belong to the class with the highest priority and will enter the elimination phase upon receiving a *rreq*. The probability that a number of s receivers will succeed in the elimination phase and enter the yield phase, denoted as $P_E\{s|n\}$, is:

$$P_E\{s|n\} = \binom{n}{s} \sum_{j=0}^{m_{ES}} P_E\{B=j\}^s P_E\{B<j\}^{n-s}, \quad (3)$$

where $P_E\{B=j\}$ is the probability that a burst has a length of j, as that in (1). $P_E\{B<j\}$ is the probability that a burst has a length less than j and is specified as follows:

$$P_E\{B<j\} = \sum_{i=0}^{j-1} P_e(i) = 1 - P_E^j. \quad (4)$$

The receivers with the longest burst will enter the yield phase. Let $\bar{B}_s(s,n)$ be the average number of bursts a successful receiver sends, on the condition that there are n receivers entering the elimination phase and s receivers win. In this case,

$$\bar{B}_s(s,n) = \frac{\binom{n}{s} \sum_{j=0}^{m_{ES}} jP_E\{B=j\}^s P_E\{B<j\}^{n-s}}{P_E\{s|n\}}. \quad (5)$$

The average burst length that a successful receiver sends when there are n receivers in the elimination phase, denoted as $\bar{B}_E(n)$, is calculated as:

$$\bar{B}_E(n) = \sum_{s=1}^{n} \bar{B}_s(s,n) P_E\{s|n\}. \quad (6)$$

In the yield phase, let $P_Y\{T=1|s\}$ be the probability of a successful transmission when there are s receivers joining the contention. A successful transmission occurs only when one receiver waits for fewer yield slots than all of the others. In this case,

$$P_Y\{T=1|s\} = \binom{s}{1} \sum_{j=0}^{m_{YS}} P_Y\{L=j\} P_Y\{L>j\}^{s-1}, \quad (7)$$

where L is the number of yield slots a successful receiver will wait. $P_Y\{L=j\}$ depends on (2), and

$$P_Y\{L>j\} = \sum_{i=j+1}^{m_{YS}} P_y(i) = \frac{m_{YS}-j}{m_{YS}+1}. \quad (8)$$

The average number of slots a successful receiver will yield before transmitting *hrep* when there are s of

TABLE 1
Mathematical Symbols Used in the Analysis

Symbol	Meaning
i_{SYNC}	time duration for synchronization interval
m_{PS}	number of slots in prioritization phase
i_{PS}	time duration for a prioritization slot
m_{ES}	number of slots in elimination phase
i_{ES}	time duration for a elimination slot
m_{YS}	number of slots in yield phase
i_{ES}	time duration for a yield slot
$P_T(n)$	probability of a successful *hrep* attempt among n contenders
B	number of bursts transmitted in the elimination phase
L	number of slots yielded in the yield phase
$\bar{B}_E(n)$	average number of bursts in elimination phase when n receivers are in *hrep* contention
$\bar{L}_Y(n)$	average number of yield slots in a successful attempt when n receivers are in *hrep* contention
$\bar{L'}_Y(n)$	average number of yield slots in a failed attempt when n receivers are in *hrep* contention
$\overline{D}_T(n)$	average time for a successful *hrep* transmission cycle
$\overline{D}_{RT}(n)$	average time for a failed *hrep* transmission cycle
$\overline{D}_{REQ}(n)$	average time for next hop determination when there are initially n contenders

n receivers entering the yield phase, denoted as $\bar{L}_s(s,n)$, is calculated as:

$$\bar{L}_s(s,n) = \frac{\sum_{j=0}^{m_{YS}} jP_Y\{L=j\}P_Y\{L>j\}^{s-1}}{P_Y\{T=1|s\}}. \quad (9)$$

The average number of slots a successful receiver will yield before it sends the *hrep* when n receivers are in the elimination phase, denoted as $\bar{L}_Y(n)$, is as follows:

$$\bar{L}_Y(n) = \sum_{s=1}^{n} \bar{L}_s(s,n)P_Y\{T=1|s\}P_E\{s|n\}. \quad (10)$$

Let the time duration for a slot in the prioritization phase, the elimination phase, and the yield phase be i_{PS}, i_{ES}, and i_{YS}, respectively. Let i_{SYNC} be the time interval for synchronization before *hrep* contention. The average delay for a successful *hrep* contention in the first attempt under the condition that there are n receivers entering the elimination phase, denoted as $\overline{D}_T(n)$, is:

$$\overline{D}_T(n) = i_{SYNC} + 2i_{PS} + \bar{B}_E(n)i_{ES} + \bar{L}_Y(n)i_{YS} + i_{ES}, \quad (11)$$

where $2i_{PS}$ stands for the two slots in the prioritization phase and the i_{ES} at the end is for the survival verification interval.

The corresponding average time for a successful next hop searching cycle from *rreq* to the *ack*, denoted as $t_{succ}(n)$, is given as:

$$t_{succ}(n) = i_{rreq} + \overline{D}_T(n) + i_{hrep} + i_{SIFS} + i_{cnfm} + i_{SIFS} + i_{ack}, \quad (12)$$

where i_{rreq}, i_{hrep}, i_{cnfm}, i_{ack}, and i_{SIFS} are the time duration for *rreq*, *hrep*, *cnfm*, *ack*, and SIFS, respectively.

In *hrep* contention phase, a transmission failure is caused by *hrep* collision when more than one receivers send *hrep* simultaneously. After detecting a *hrep* collision, the sender will wait for a SIFS before it resends a *rreq*.

Let $\bar{L'}(w,s,n)$ be the average number of slots in the yield phase in a failed *hrep* contention when w of s receivers have the shortest yield phase and send the *hrep*. n is the number of contenders in the elimination phase. Let $P_Y\{T=w|s\}$ be the corresponding probability of such a case. It is given as follows:

$$P_Y\{T=w|s\} = \binom{s}{w} \sum_{j=0}^{m_{YS}} P_Y\{L=j\}^w P_Y\{L>j\}^{s-w}. \quad (13)$$

Note that $2 \le w \le s$. A failed *hrep* contention thus implies that there are at least two receivers entering the yield phase. Since, in *hrep*, the probability that no receiver enters the yield phase is 0, the probability that more than one receiver enters the yield phase is $(1 - P_E\{1|n\})$. $\bar{L'}(w,s,n)$ is calculated as follows:

$$\bar{L}'(w,s,n) = \frac{\binom{s}{w}\sum_{j=0}^{m_{YS}} jP_Y\{L=j\}^w P_Y\{L>j\}^{s-w}}{(1-P_E\{1|n\})P_Y\{T=w|s\}}. \quad (14)$$

In a failed transmission, the average number of the yield slots before any receiver transmits a *hrep* when there are s receivers in the yield phase, denoted as $\bar{L}'(s)$, is given as:

$$\bar{L}'(s,n) = \sum_{w=2}^{s} \bar{L}'(w,s,n)P_Y\{T=w|s\}. \quad (15)$$

The average number of yield slots before any receiver transmits a *hrep* when there are n receivers in the elimination phase, denoted as $\bar{L}'_Y(n)$, is given as:

$$\bar{L}'_Y(n) = \sum_{s=2}^{n} \bar{L}'(s,n)P_E\{s|n\}. \quad (16)$$

Assume that a previous *hrep* contention with n contenders fails and is followed by a new contention. Let $\bar{D}_{RT}(n)$ be the average contention time for a failed *hrep* contention, which can be calculated by:

$$\bar{D}_{RT}(n) = i_{SYNC} + 2i_{PS} + \bar{B}_E(n)i_{ES} + \bar{L}'_Y(n)i_{YS} + i_{ES}. \quad (17)$$

Let t_{fail} be the time needed for the sender to detect a *hrep* failure since it transmits *rreq*, which can be calculated by:

$$t_{fail}(n) = i_{rreq} + \overline{D}_{RT}(n) + i_{hrep} + i_{SIFS}. \quad (18)$$

The probability of a successful transmission among n contending receivers in the elimination phase, denoted as $P_T(n)$, is:

$$P_T(n) = \sum_{s=0}^{n} P_E\{s|n\}P_Y\{T=1|s\}. \quad (19)$$

Assume that a sender has to send the *rreq* k times before it receives a *hrep* successfully. The average delay for a sender to determine its next hop when there are n neighboring nodes contending to be the next hop is denoted by $\overline{D}_{REQ}(n)$ and can be calculated by:

$$\begin{aligned}\overline{D}_{REQ}(n) &= \sum_{k=1}^{\infty} P_T(n)(1-P_T(n))^{k-1}(t_{succ}(n)+(k-1)t_{fail}(n)) \\ &= P_T(n)t_{succ}(n)\sum_{k=1}^{\infty}(1-P_T(n))^{k-1} \\ &\quad + t_{fail}(n)P_T(n)(1-P_T(n))\sum_{k=1}^{\infty}(k-1)(1-P_T(n))^{k-2} \\ &= t_{succ}(n) + \frac{(1-P_T(n))}{P_T(n)}t_{fail}(n).\end{aligned} \quad (20)$$

Our observations and the data with some sample values for various parameters is given in Section 5.

4.2 Routing Failure and Impact of Inaccurate Position Information

The low-cost high-accuracy position service is not yet available. There are relatively large position errors for either the GPS system, or the cellular position (location) service system, especially when considering that such a facility used at the mobile device must be small and low power-consuming.

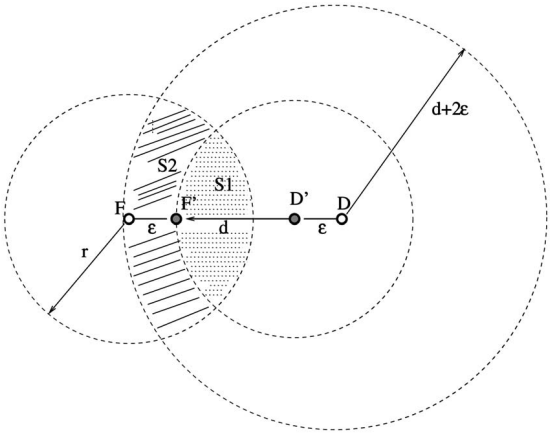


Fig. 6. Impact of position error on routing discovery.

In AO2P, the position error at senders or receivers may make the receivers assign themselves to the wrong classes. A node actually closer to the destination may lose *hrep* contention to a node actually farther away. The end-to-end multihop connection based on these inaccurate positions may not be the best in terms of number of hops. However, position errors will not cause a routing failure by generating the links that actually do not exist, because only a node receiving a *rreq* can possibly be the next hop. The connection between a sender and a receiver thus is always real regardless of wrong positions.

Assume that the maximum error of a position service is ϵ, which means the real position may be as far as ϵ away from the position given by the position service provider. Assume that the average geographic distance for each hop is d when the correct positions are used. When there is a position error, the percentile increase for the number of hops may at most be $100 \times (\frac{d}{d-\epsilon-1})$ percent.

More seriously, position error may cause extra routing discovery failure. Based on the wrong positions, a legitimate receiver may think it cannot process the *rreq* closer to the destination and does not participate in *hrep* contention. This potential link cannot be used by the sender even if it cannot find any other legitimate next hop.

Fig. 6 illustrates the worst case of how the position error affects AO2P routing discovery. r is the ad hoc radio coverage. Due to the position error of ϵ, a sender with the actual position of F thinks that it is at the position F'. The destination thinks that it is at D' instead of D. Thus, the sender thinks the distance between the destination and itself is d instead of the real distance of $d+2\epsilon$. A legitimate next hop can only be from S_1, since only the nodes in S_1 are within the radio coverage from F and has a shorter distance to D' than that from F' to D'. A route discovery failure will occur if there is no node in S_1. If both the sender and the receiver have the right positions, the probability of a routing failure will decrease. This occurs when there is no node in both of the areas S_1 and S_2.

If a large number of N nodes are uniformly distributed in an ad hoc network covering an area of S, the node density ρ

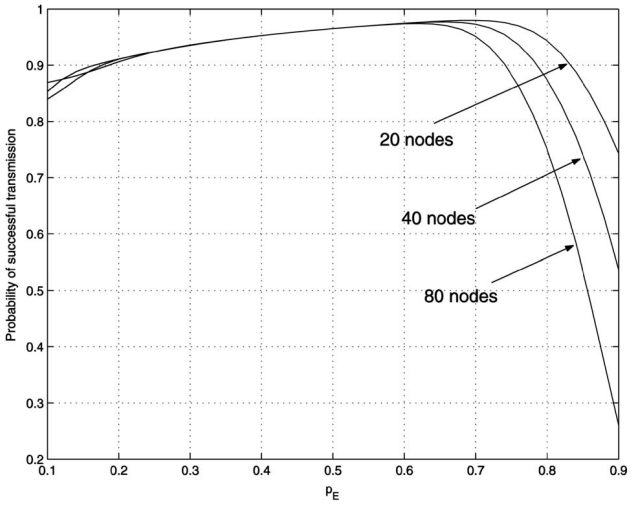


Fig. 7. Probability for successful *hrep* transmission at different P_E.

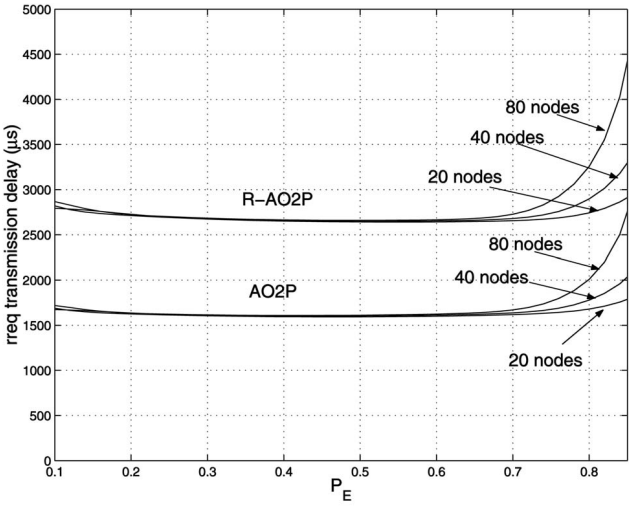


Fig. 8. Average time to determine a next hop at different P_E.

then is N/S. For any area of S_0, the probability that no node resides in it, denoted as $p_{S_0}(n=0)$, is as follows:

$$P_{S_0}(n=0) = \left(1 - \frac{S_0}{S}\right)^N = \left(1 - \frac{S_0}{N/\rho}\right) = \left(\left(1 - \frac{\rho S_0}{N}\right)^{\frac{N}{\rho S_0}}\right)^{\rho S_0}. \tag{21}$$

ρS_0 is the number of the nodes residing in the area S_0. When N is large enough compared to ρS_0,

$$P_{S_0}(n=0) \approx e^{-\rho S_0} \tag{22}$$

The probability that there is at least one node in area S_0, denoted as $P_{S_0}(n \geq 1)$, is:

$$P_{S_0}(n \geq 1) = 1 - P_{S_0}(n=0) = 1 - e^{-\rho S_0}. \tag{23}$$

Let us define p_1 as the probability of such a routing discovery failure for the worst case of position errors and p_2 as the probability of a routing discovery failure when there is no position error. Referring back to Fig. 6, when the nodes are uniformly distributed with a density of ρ, based on the previous analysis results,

$$p_1 = e^{-\rho S_1},$$
$$p_2 = e^{-\rho(S_1+S_2)},$$

where S_1 and S_2 are functions of ϵ and d.

In R-AO2P, $d \gg r$. At the source and each forwarding node, the worst case scenario (refer to Fig. 6) results in approximately the same S_1 and S_2. Therefore, the probability of a route discovery failure with and without position error, i.e., p_1 and p_2, are approximately the same. In a n-hop end-to-end connection, the probability of a routing failure with and without position errors can be calculated as $1-(1-p_1)^n$ and $1-(1-p_2)^n$.

5 Illustrative Data and Observations

In this section, we present both analysis and simulation studies. We first present the analytical results on route discovery delay and the probability of a failure in route discovery based on accurate or inaccurate position information. We then use a simulation model to study the probability of a route discovery failure under destination mobility and inaccurate positions. We also use simulation to study the average number of hops for the generated routes and the end-to-end throughput.

5.1 Analysis Results

5.1.1 The Average Delay for rreq Transmission Cycle

The major parameters in an AO2P *hrep* contention period are set the same as those in HIPERLAN1 standard. The number of slots in the prioritization phase, the elimination phase, and the yield phase are 5, 12, and 9, with the duration time of 7.2 μs, 9 μs, and 7.2 μs, respectively. Time duration for the synchronization interval is 11 μs. Time duration for SIFS and DIFS are 28 μs and 128 μs, as those in WLAN. Note that the duration of DIFS is longer than the duration of a complete yield phase. *rreq*, *hrep*, and *cnfm* are transmitted at the rate of 1 Mb/s, with the length shown in Fig. 2. An extra physical header of 128 $bits$ is added to each frame. *ack* has an overall length of 240 $bits$ and is also transmitted at the rate of 1 Mb/s.

Fig. 7 shows the probability of a successful *hrep* transmission. It shows that, when P_E, the parameter for a node to adjust the burst length in the elimination phase, is correctly chosen, the probability for a successful *hrep* transmission is very high even when there are a large number of contending receivers. The optimum successful transmission rate occurs at $P_E = 0.65$, where the probabilities of successful *hrep* transmissions are above 95 percent for different number of contending nodes.

The corresponding average delay for a node to determine its next hop (i.e., the average time for the completion of the *rreq* transmission cycle) is shown in Fig. 8. In R-AO2P, 64 $bytes$ are added in *hrep* and *cnfm* for Diffie-Hellman key exchange. The curves of the delays are like the reverse of those for the probabilities of successful transmission, which means a high transmission probability results in a low next hop searching delay. The results show that the average time for the next hop discovery is only a few milliseconds. This

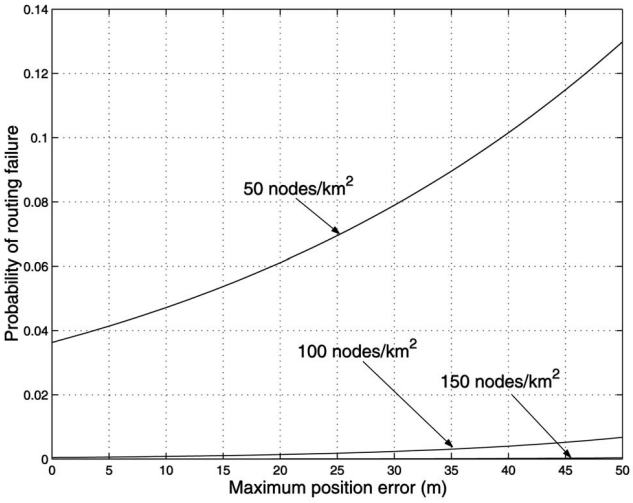


Fig. 9. Routing failure at different position accuracy.

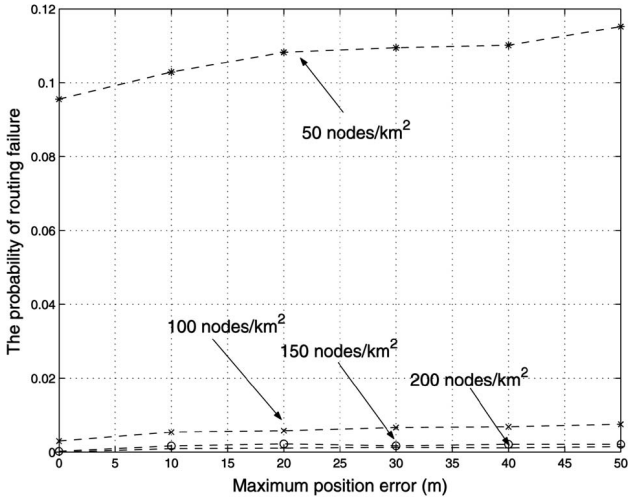


Fig. 10. Probability of routing failure at different position accuracy.

means that a large delay in AO2P routing discovery can only occur in a heavily-loaded network, where the delay for *rreq* channel access is high. Such a delay may lead to a route discovery failure if the destination moves away from its previously reported position (as shown in Fig. 12). Thus, AO2P can be looked at as a self-adaptive protocol as it impedes a new data source to join a heavily-loaded network through causing route discovery failure and prevents the network congestion from getting worse.

5.1.2 Impact of Position Error

Fig. 9 shows the probabilities of a routing failure for R-AO2P in the worst case scenario when there are different maximum position errors. The results are based on the assumptions that 3-hop connections are needed and nodes are uniformly distributed. Routing failure increases as the position error gets worse. The analysis results show that when node density gets higher, the impact of the position error is less significant.

5.2 Simulation Results

The simulation scenario is a network covering an area of $1,000\ m \times 1,000\ m$, where a number of nodes are uniformly deployed. The transmission range for the ad hoc channel is $250m$. The receivers are divided into 4 classes according to the rules in Section 3.3.

5.2.1 Impact of Position Error

Fig. 10 shows the increased probability of a routing failure in AO2P as the position error gets worse. The x-axis is the maximum position error ϵ_{max}. In simulation, the position of a node used for route discovery is a distance away from its real position. This distance is an assigned random value distributed uniformly between 0 and ϵ_{max}. The data shows that when node density is high, i.e., greater than $150/km^2$, the probability of a routing failure is very low even when the position error is large.

Fig. 11 shows the average number of hops for the discovered routes at different position errors. It shows that, as the value of maximum position error increases, the hop number increases. However, the increase in the hop number is not significant and will not have much impact on routing performance.

5.2.2 Impact of Destination Mobility

Fig. 12 shows the probability of a routing discovery failure caused by destination mobility. Such a routing failure will not occur if the destination stays at the position known to the source. We define the parameter *drift* of destination as the distance that the destination is away from the position carried in the *rreq* for routing discovery. It shows that, as the drift value increases, the probability of a routing failure increases. The impact of destination mobility is more obvious in the network with low node densities. Fig. 12 can provide the distance threshold value based on which a node has to update its position. For example, if the probability of a route discovery failure caused by position drift is required to be no more than 0.005, at low node densities (50 nodes$/km^2$), the threshold value should be no more than $30m$. At high node densities (100 nodes$/km^2$ or

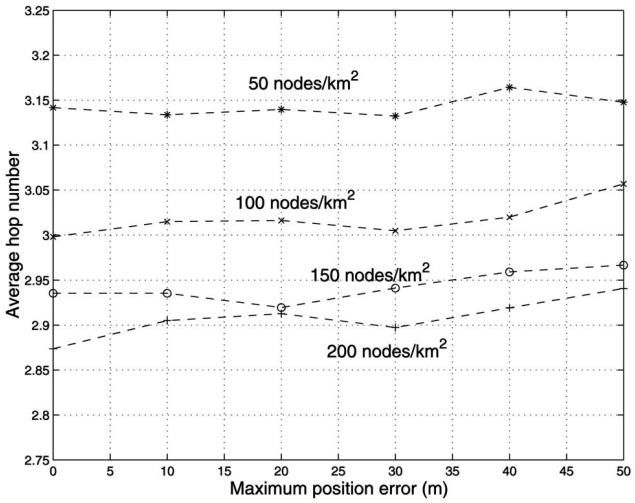


Fig. 11. Average number of hops in a discovered route at different position accuracy.

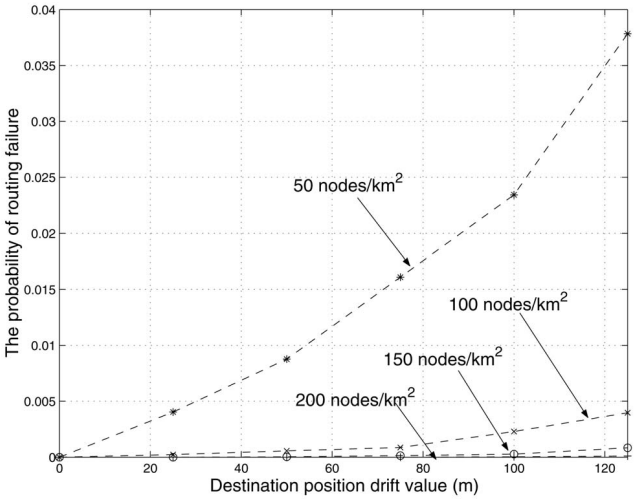


Fig. 12. Routing discovery failure caused by the drift of destination.

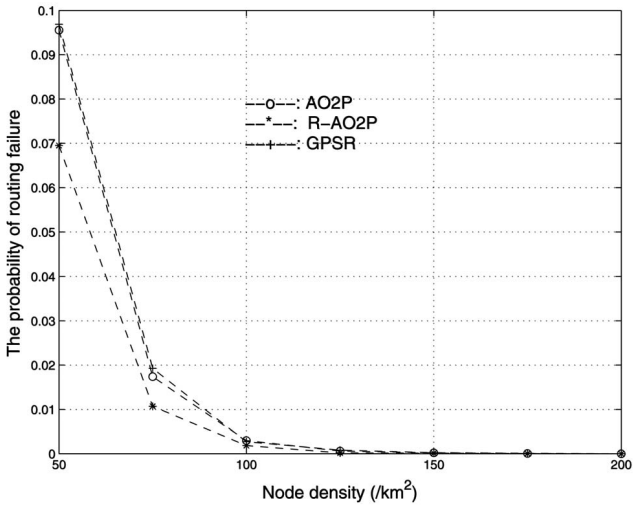


Fig. 14. Routing discovery failure in AO2P, R-AO2P, and GPSR.

above), a node need not update its position unless it is more than $125\ m$ away from the position it reported last time. $125\ m$ is the half of the ad hoc radio transmission range.

Other than causing routing failure, the drift of destination can also lead to inefficient routing. Fig. 13 shows that the average hop number increases as the drift value increases. The reason is that the *rreq* is forwarded to the destination position. Therefore, a shortest path (in terms of number of hops) to the destination normally cannot be found.

5.2.3 AO2P, R-AO2P, and GPSR Comparisons

Fig. 14 compares the probabilities of a route discovery failure in the networks where AO2P, R-AO2P, and GPSR are used for route discovery. For fair comparison, GPSR is modified so that a node can be the next hop of a sender only when this node is closer to the destination. It shows that AO2P and GPSR have approximately the same probability for a routing discovery failure. R-AO2P has a lower probability because it allows a node leading the *rreq* away from the destination to be the next hop. Note that, for R-AO2P, the probability of a route discovery failure in simulation is higher than the analytical results obtained and shown in Fig. 9. The reason is that in simulation, a lot of nodes close to the boundary of the network are included in the routes. It is more likely that these nodes cannot find the legitimate next hop, which results in a route discovery failure.

Fig. 15 compares the average number of hops for end-to-end connections. It is observed that GPSR has the smallest hop count, as it always uses the node that is closest to the destination as the next hop. R-AO2P has larger hop counts than AO2P, as nodes belonging to the class of lower priority (farther from the destination) and cannot win the *hrep* contention in AO2P may be assigned to the class of higher priority in R-AO2P and win the contention. In all cases, the average hop number of the routes decreases as the node density gets higher.

Fig. 16 compares the simulated delivery ratio in the network when routes are discovered by different protocols. ns-2 is used as the simulator, as it has the well-developed CSMA/CA model. Network has a medium size and density, with 100 low-mobility nodes uniformly distributed

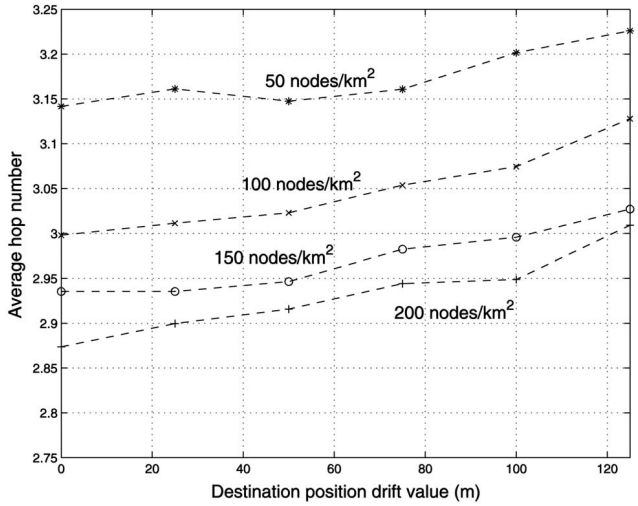


Fig. 13. Average number of hops at different drift values.

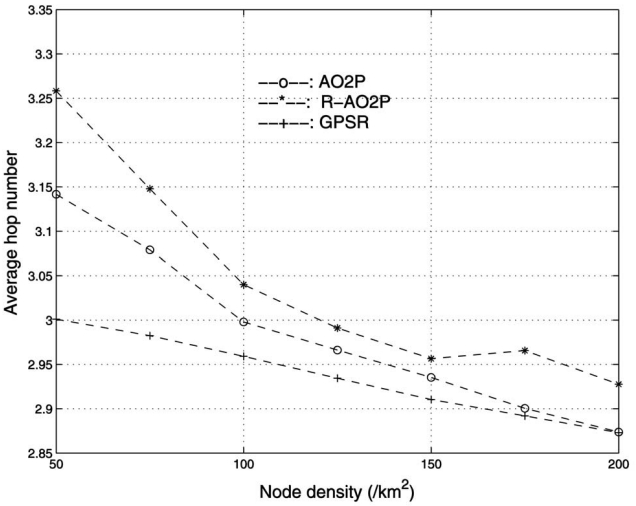


Fig. 15. Average number of hops for AO2P, R-AO2P, and GPSR.

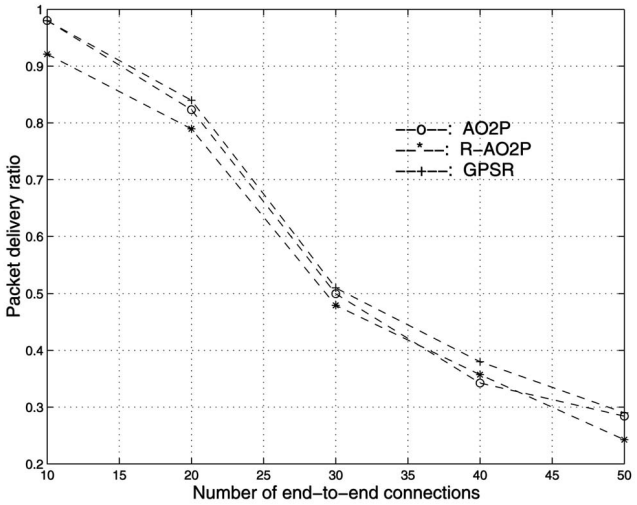


Fig. 16. Data packet delivery ratio in AO2P, R-AO2P, and GPSR.

in a $1,000\ m \times 1,000\ m$ area. The constant bit rate flow is used as data input for each connection, with the data packet arrival rate of four per second and a packet size of $512\ bytes$. The available data rate in an ad hoc channel is $1\ Mb/s$. It shows that, generally, GPSR has the highest delivery ratio as it has the minimum hop count. R-AO2P has the lowest delivery ratio. However, the routing performance degradation in AO2P and R-AO2P is not significant.

6 Conclusions and Future Works

This research proposes a routing algorithm, named AO2P, to achieve communication privacy in ad hoc networks. Node position, instead of identity, is used for route discovery. Only limited position information is revealed to the network to protect node anonymity. In an enhanced algorithm R-AO2P, the position of a reference point, instead of the position of the destination, is used to further improve destination anonymity. We use analysis and simulation to evaluate the routing performance for the proposed algorithms.

In the MAC layer, we build an analytical model to evaluate the extra delay caused by the proposed receiver contention scheme. We find that the delay is small and a search for the next hop in AO2P or R-AO2P takes only a few milliseconds. Such a delay does not result in the failure of a route discovery even if destinations are highly mobile.

In the network layer, we first use analysis and simulation models to evaluate the impact of position error on route discoveries. It is observed that a large error may cause inefficient routing, i.e., routes built up with a greater number of hops, or may even cause a route failure. However, this impact is less significant in the networks with high node densities. For example, in a network with a node density of $200/km^2$, the probability of a route discovery failure can be as low as 0.001 even if the position error is as high as the half of the maximum ad hoc radio coverage. We use simulation to study the impact of destination mobility. The movement of a destination makes its position known by the source incorrect. However, it is observed that a route discovery based on a "false" destination position may not necessarily leads to a route discovery failure. Again, in networks with high node densities, the impact of destination mobility is less significant. The results can be used to determine the distance threshold value in the distance-based position update system, where an ad hoc node updates its position when its current position is more than a distance away from the last reported one. Finally, we compare the routing performance between AO2P/E-AO2P and GPSR. We compare the hop counts in the routes discovered by these algorithms. It is observed that the routes in AO2P or R-AO2P have only marginally greater hop counts than in GPSR, yet GPSR requires much more position information. Simulation also shows that the corresponding end-to-end throughput degradation in AO2P and R-AO2P is not significant. Therefore, AO2P preserves communication privacy without significant routing performance degradation.

We propose the following two future research directions:

- **Privacy evaluation.** Internal attackers are able to obtain pieces of position information of their targets. Based on this information, they can estimate the trajectories of their target or reduce the anonymity set. The level of destination anonymity can then be quantified by a probability of matching a position to any node ID. This can be calculated if node mobility, traffic pattern, and the policies for position services are given. Future work will include building analytical models for mobility and traffic based on which node anonymity can be quantified. Other than probabilities, entropy and size of anonymity sets will also be considered as the metric for anonymity evaluation.
- **Security issues and mitigation techniques.** In AO2P, the next hop is determined by node contention. A malicious node can always use the most aggressive contention mechanism to become the next hop. Once it is included in a route, it can conduct different attacks, such as changing the position of destination in the routing request or dropping/fabricating data packets after the route is built up. More seriously, protecting the privacy of intermediate nodes in AO2P makes it almost impossible to identify these attackers. A modified channel contention scheme will be developed. The next hop cannot be decided by a receiver itself, but by both the sender and the receiver. Information exchange is needed based on which a required trust between a previous hop and its potential next hop can be assessed. The privacy degradation due to such information requires investigation. Incentives for a node to be a forwarder at the cost of the degraded privacy has to be provided.

Acknowledgments

The authors would like to thank Dr. Ninghui Li for his valuable, insightful advice and comments on the cryptographic application in this paper. They also thank Gang Ding for running simulation and collecting data. This work is supported by Information Infrastructure Protection (I3P) Fellowship, US National Science Foundation grants CCR-0001788, ANI-0219110, Motorola, and CISCO.

REFERENCES

[1] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. Second IEEE Workshop Mobile Computing Systems and Applications,* 1999.

[2] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Proc. ACM SIGCOMM-Computer Comm. Review,* 1996

[3] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM,* 1994.

[4] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "p5: A Protocol for Scalable Anonymous Communication," *Proc. IEEE Symp. Security and Privacy,* 2002.

[5] V. Scarlata, B. Levine, and C. Shields, "Responder Anonymity and Anonymous Peer-to-Peer File Sharing," *Proc. IEEE Int'l Conf. Network Protocols,* 2001.

[6] E. Bommaiah, A. McAuley, R. Talpade, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *Internet-Draft, IETF,* Aug. 1998.

[7] L. Sweeney, "K-Anonymity: A Model for Protecting Privacy," *Int'l J. Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[8] L. Xiao, Z. Xu, and X. Zhang, "Low-Cost and Reliable Mutual Anonymity Protocols in Peer-to-Peer Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 14, no. 9, pp. 829-840, 2003.

[9] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous Connections and Onion Routing," *IEEE J. Selected Areas in Comm.,* special issue on copyright and privacy protection, vol. 16, no. 4, pp. 482-494, 1998.

[10] M.K. Reiter and A.D. Rubin, "Crowds: Anonymity for Web Transactions," *ACM Trans. Information and System Security,* vol. 1, no. 1, pp. 6-92, 1998.

[11] I. Stojmenovic, "Position Based Routing in Ad Hoc Networks," *IEEE Comm. Magazine,* vol. 40, no. 7, pp. 128-134, 2002.

[12] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," *Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Comm.,* 2000.

[13] Y. Zhong and B. Bhargava, "Using Entropy to Trade Privacy for Trust," *Proc. Workshop Secure Knowledge Management,* 2004.

[14] D. Kesdogan, J. Egner, and R. Buschkes, "Stop-And-Go-MIXes Providing Probabilistic Anonymity in an Open System," *Proc. Int'l Information Hiding Workshop,* 1998.

[15] A. Serjantov and G. Danezis, "Towards an Information Theoretic Metric for Anonymity," R. Dingledine and P. Syverson, eds., *Proc. Privacy Enhancing Technologies Workshop,* 2002.

[16] J. Kong and X. Hong, "ANODR: Anonymous on Demand Routing with Untraceable Routes for Mobile Ad-Hoc Networks," *Proc. Fourth ACM Int'l Symp. Mobile Ad Hoc Networking and Computing,* 2003.

[17] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Network," *Proc. MOBICOM,* 2000.

[18] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad-Hoc Wireless Networks," *ACM Wireless Networks,* vol. 7, no. 6, pp. 609-616, Nov. 2001.

[19] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing Using Partial Information for Wireless Ad Hoc Networks," *IEEE Personal Comm. Magazine,* pp. 48-57, Feb. 2001.

[20] I. Stojmenovic, M. Russell, and B. Vukojevic, "Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks," *Proc. IEEE Int'l Conf. Parallel Processing,* 2000.

[21] B. Bhargava, X. Wu, Y. Lu, and W. Wang, "Integrating Heterogeneous Wireless Technologies: A Cellular-Assisted Mobile Ad Hoc Networks," *Mobile Network and Applications,* special issue on integration of heterogeneous wireless technologies, no. 9, pp. 393-408, 2004.

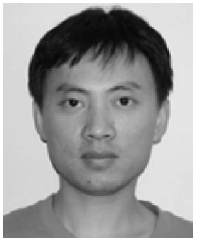
[22] L. Blazevic, L. Buttyan, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, "Self-Organization in Mobile Ad Hoc Networks: The Approach of Terminodes," *IEEE Personal Comm. Magazine,* pp. 166-174, June 2000.

[23] Y. Xue, B. Li, and K. Nahrstedt, "A Scalable Location Management Scheme in Mobile Ad-Hoc Networks," *Proc. 26th Ann. IEEE Conf. Local Computer Networks,* 2001.

[24] X. Wu, "DISPOSER: DIstributed Secure POSition SERvice in Mobile Ad Hoc Networks," Technical Report CSD TR # 04-027, Dept. Computer Sciences, 2004.

[25] ETSI HIPERLAN/1 Standard, http://portal.etsi.org/bran/kta/Hiperlan/hiperlan1.asp, 1998.

[26] G. Anastasi, L. Lenzini, and E. Mingozzi, "Stability and Performance Analysis of HIPERLAN," *Proc. IEEE Conf. Computer Comm. (INFOCOM),* 1998.



Xiaoxin Wu received the BE degree from the Beijing University of Posts and Telecommunications in 1990 and the PhD degree from the University of California, Davis, in 2001. After that, he joined Arraycomm Inc. as a protocol research engineer. Since 2002, he has been working as a postdoctoral researcher in the Department of Computer Science, Purdue University. He is currently supported by an Institute for Information Infrastructure Protection (I3P) research fellowship and working on wireless network privacy and security. His other research interests include designing and developing architecture, algorithms, and protocol for wireless network performance improvement.

Bharat Bhargava received the BE degree from the Indiana Institute of Science and the MS and PhD degrees in EE from Purdue University. He is a professor of computer sciences at Purdue University. His research involves host authentication and key management, secure routing and dealing with malicious hosts, adaptability to attacks, and experimental studies. Related research is in formalizing evidence, trust, and fraud. Professor Bhargava is a fellow of the IEEE and of the Institute of Electronics and Telecommunication Engineers. He has been awarded the charter Gold Core Member distinction by the IEEE Computer Society for his distinguished service. In 1999, he received an IEEE Technical Achievement award for a major impact of his decade long contributions to foundations of adaptability in communication and distributed systems.