

An Ant-based Routing Protocol using Unidirectional Links for Heterogeneous Mobile Ad-Hoc Networks

Taro Maekawa, Harumasa Tada, Naoki Wakamiya, Makoto Imase and Masayuki Murata
Graduate School of Information Science and Technology
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
Email: {maekawa, tada, wakamiya, imase, murata}@ist.osaka-u.ac.jp

Abstract

Because of the heterogeneity in radio transceiver capabilities and influence of surroundings, unidirectional links arise in wireless mobile ad-hoc networks. Since most of existing routing protocols for MANETs assume that all links are bidirectional, they would fail in establishing a path or cannot prepare sufficient number of paths for the purpose of multipath routing in a network with unidirectional links. In this paper, based on an ant-based routing algorithm, we propose a routing protocol which uses unidirectional links for establishing paths to improve the connectivity of networks and take advantage of multipath routing strategy. However, using unidirectional links introduces new problems such as detection of link disconnection. In order to solve these problems, we propose three mechanisms; detouring around a unidirectional link, detecting link failures by ants, and blind retransmission. Simulation experiments showed that the connectivity increased and more packets could be delivered than AODV and AntHocNet which did not use unidirectional links. It was also shown that the path establishment delay was shorter and the load of control packets per data packet was lower than AntHocNet.

1 Introduction

Mobile Ad-hoc Networks (MANETs) consist of a collection of wireless mobile nodes which communicate with each other over radio. There is no fixed infrastructure such as routers, wired links, and access points. Due to limited transmission range of wireless interfaces, in most cases packets have to be relayed over intermediate nodes by multi-hop communication, where each node plays a role of router. For MANETs, it is important to design routing algorithms that are adaptive to changes in radio environments and network topology, robust to failures of nodes and links, and self-organizing. Self-organizing systems observed in

the nature, such as insect societies inherently have these desirable properties. They are adaptive to changes in their environment and robust to collapse of the nest and death of nestmates. The intelligent behavior which emerges from the collection of simple behavior of small agents is called *swarm intelligence* [3].

Routing protocols taking inspiration from the swarm intelligence, especially Ant Colony Optimization (ACO) algorithms are called *ant-based routing protocols* [1, 4, 5, 2]. In natural environment, ants leave chemical substances, called *pheromone*, between their nest and food which they find. Ants trace pheromones deposited by their nestmates to reach the food and bring it back to their nest leaving their own pheromones. In ant-based routing protocols, each node generates control packets, i.e., ants to find or maintain paths from a source node to a destination node. Depending on a protocol, ants are disseminated over a network, wander around to find a path to the destination, or move toward the destination. When an ant successfully reaches the destination, it then goes back to the source node to reinforce the taken path with much pheromones to attract more ants and data packets. Data packets are routed stochastically to a destination node. An intermediate node choose a neighbor node as the next-hop node with the probability calculated from pheromone values.

Most of ant-based routing protocols are multipath routing protocols. With multipath routing protocols, multiple paths are established between a single source node and a single destination node [8]. Data packets are transmitted to a destination node on a path chosen among multiple paths based on some criteria. If a primary path gets worse or disconnected, another path is immediately chosen and compensates the failed path. Multipath routing protocols have advantages in connectivity, robustness, reliability, adaptability, and load balancing.

In MANETs, not only bidirectional links but also unidirectional links arise. One of the major causes of such link is different transmission ranges of heterogeneous nodes as illustrated in Fig. 1. We call MANETs including nodes

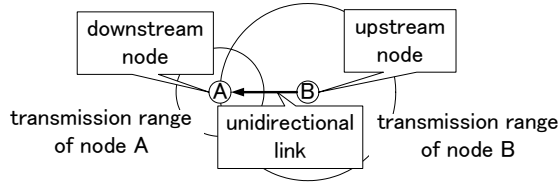


Figure 1. An example of a unidirectional link

of different transmission ranges as *heterogeneous Mobile Ad-Hoc Networks (heterogeneous MANETs)*. Heterogeneous MANETs include bidirectional links and unidirectional links. One end of a unidirectional link is called a *downstream node*, which receives packets from the other but cannot send ones back. The other end is called an *upstream node*. Most of routing protocols designed for MANETs work with the assumption that all nodes in a network have the same transmission range or all wireless links are bidirectional. However this assumption does not always hold. If unidirectional links exist, those protocols may decrease their performance or at worst cannot establish any path. In order to tackle the problem, some protocols first eliminate unidirectional links from components of paths in path computation procedure by, for instance, BlackListing, Hello Messages to detect unidirectional links, and Reverse Path Search [7]. Then, paths consisting of only bidirectional links are established. Although they can avoid the problem of unidirectional links, they establish a longer path or fewer number of paths, which leads to weaker network connectivity and lower packet delivery ratio than protocols using unidirectional links. Therefore, some other routing protocols use unidirectional links [13, 12, 11, 6]. In [7, 14], they evaluated the benefit of using unidirectional links and the results indicate that the benefit of using a high-overhead routing protocol to use unidirectional links is questionable or proves to be costly. However, they assumed establishing only one path for a source-destination pair. If we incorporate unidirectional links with the multipath strategy, the performance can be improved because of more number of paths.

In this paper, we propose an ant-based routing protocol which establishes paths including unidirectional links based on AntHocNet [2]. We extended AntHocNet to use unidirectional links and establish more paths, which enhances advantages of multipath strategy such as reliability, adaptability, and connectivity.

The rest of this paper is organized as follows. We propose a novel routing protocol in Section 2, and then show some simulation results in Section 3. Finally, we conclude the paper in Section 4.

2 Ant-based Routing Protocol using Unidirectional Links

In this section, we propose a routing protocol for heterogeneous MANETs based on AntHocNet, which is an ant-based routing protocol to establish multiple paths to a destination in a reactive way and maintain them in a proactive way. Data packets are stochastically transmitted over multiple links. We extended AntHocNet to effectively incorporate unidirectional links by proposing three extensions. They are *detouring unidirectional links*, *blind retransmission*, and *detection of link failure by ants*.

2.1 Reactive Path Setup and Detouring Unidirectional Links

When source node s wants to send data packets to destination node d whose corresponding routing information is not available, it broadcasts a reactive forward ant F_d^s to establish paths to node d . A reactive forward ant has the address of both a source node and a destination node, a unique identifier, and list P whose entry is a pair of the address of a node which F_d^s has visited and the time F_d^s arrived at the node. List P is called *visited nodes list*. A set of replicas which are generated from the same ant by broadcasting is called an *ant generation*. The generation is identified with the unique identifier.

If pheromone information is available at node i for destination d , a reactive forward ant chooses its next-hop node n with probability P_{nd} , which is defined as;

$$P_{nd} = \frac{(T_{nd}^i)^{\beta_1}}{\sum_{j \in N_d^i} (T_{nd}^j)^{\beta_1}}, \quad \beta_1 \geq 1, \quad (1)$$

where T_{nd}^i is the pheromone value of the entry of routing table T^i for neighbor node n as the next-hop node to destination node d . N_d^i is the set of neighbors of node i whose pheromone values are defined for destination node d , and β_1 is a constant. If β_1 is large, a neighbor node with a higher pheromone attracts a reactive forward ant more than in the case with a smaller β_1 . In our simulation experiments, β_1 is set to one. If no pheromone is available for destination node d , a reactive forward ant is broadcast to find paths to destination node d . To eliminate useless ants going toward a wrong direction or taking too long path, the lifetime of a reactive forward ant is defined by a source node as the maximum number of hops, i.e., TTL. In addition, when a node receives several reactive forward ants of the same generation, it decides whether to discard or accept the ants based on the length of the path they travelled. Assume that the i -th reactive forward ant has spent n_i hops and t_i time unit from a source to the node. If $n_i \leq a_1 \min_{j < i} (n_j)$ and $t_i \leq a_1 \min_{j < i} (t_j)$, the reactive forward ant is relayed to

neighbor nodes. Here, a_1 is called an acceptance factor and $0 < a_1 < 1$. Although this scheme contributes to eliminating too long paths, it brings a problem to form ‘kite-shaped’ paths [2] which are not disjoint. In order to establish disjoint paths for higher robustness against link failures and better load distribution, the first hop taken by a reactive forward ant is taken into account in thinning out ants. If the first hop of a newly received ant is different from those of all previously accepted ants, a higher acceptance factor $a_2 > a_1$ is applied.

When a reactive forward ant arrives at destination node d , a backward ant is generated. So that a backward ant can return to source node s by taking the same path that the corresponding reactive forward ant has travelled, it has a copy of visited nodes list P . A backward ant has its own unique identifier and the identifier of the corresponding reactive forward ant. As a backward ant moves one hop toward source node s , it computes an estimate \hat{T}_d^i of the time from the node i to destination node d .

At each node $i \in P$, a backward ant creates or updates an entry with pheromone value T_{nd}^i for destination node d in routing table T^i . According to [2], the pheromone value T_{nd}^i is updated as follows;

$$T_{nd}^i \leftarrow \gamma T_{nd}^i + (1-\gamma) \left(\frac{\hat{T}_d^i + h \cdot T_{hop}}{2} \right)^{-1}, \quad \gamma \in [0, 1], \quad (2)$$

where \hat{T}_d^i and h correspond to the estimated time and the number of hops from node i to destination node d via node n , respectively. T_{hop} is the fixed parameter indicating the time taken for one hop in unloaded conditions. γ is a smoothing parameter.

If a source node receives one or more backward ants, paths are successfully established to destination node d . Otherwise, source node s temporarily buffers data and the whole process is conducted again. The number of retries is limited, after which the buffered data are discarded.

To update pheromone values on the path that the corresponding forward ant traversed, a backward ant has to visit all intermediate nodes in path P in reverse order, i.e., from destination node d to source node s . However, if path P contains a unidirectional link from source node s to destination node d , a backward ant fails. In our proposal, we introduce a mechanism for a backward ant to detour around a unidirectional link. An example is illustrated in Fig. 2. When a backward ant encounters a unidirectional link from node 2 to node 3 on the way back to source node s , the detouring process by flooding is initiated. A *broadcast backward ant* is generated at node 3, i.e., the downstream node of a unidirectional link. A broadcast backward ant has the same information as the backward ant and the address of the originating node (node 3 in this example) and a unique identifier. A broadcast backward ant is spread around the originating node by broadcasting. To avoid wasting the net-

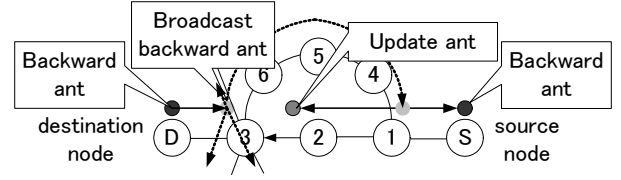


Figure 2. Detouring around a unidirectional link

work bandwidth, the maximum number of broadcasting is limited to n_b^{detour} . When a broadcast backward ant reaches any of unvisited nodes in visited nodes list P , it updates the routing table of the node. In the example, node 1 is the node at which a broadcast backward ant finishes the detouring. Then, a broadcast backward ant is converted into a backward ant and resumes the travel to source node s .

The problem is that nodes in the path which are bypassed by a backward ant cannot update their routing tables. In the example, node 2 is the node which a backward ant did not visit. In order to update routing tables of such nodes, an *update ant* is generated. An update ant moves toward destination node d in visited nodes list P while updating routing tables. When an update ant arrives at a node whose routing table has already been updated, it is discarded.

2.2 Stochastic Data Packet Forwarding and Blind Retransmission

The probability P_{nd} that neighbor node n is chosen as the next-hop node to destination node d is defined as;

$$P_{nd} = \frac{(T_{nd}^i)^{\beta_2}}{\sum_{j \in N_a^i} (T_{nd}^j)^{\beta_2}}, \quad \beta_2 \geq \beta_1 \geq 1. \quad (3)$$

However, an upstream node cannot confirm the success of transmission of data packets over a unidirectional link, since it cannot receive a link-level acknowledgement from a downstream node. Possible solutions are to tunnel an acknowledgement [9] and to relay an acknowledgement by the other nodes [12]. However, these solutions need to add more complex mechanisms to MAC protocol such as IEEE 802.11 DCF. We intend to control forwarding of data packet over a unidirectional link only with routing protocol, since we do not want to change MAC protocol. It takes much cost to change MAC protocol, because it is usually implemented on network devices.

In our proposal, we use broadcasting without acknowledgement to send a data packet over a unidirectional link. A data packet is encapsulated with a header consisting of the address of the intended receiver, i.e., the downstream node, and a unique identifier. On receiving an encapsulated packet, the downstream node takes out the original packet

and forwards it to destination node d . However, our scheme lacks the reliability of packet forwarding. To alleviate the shortage, we introduce a scheme which we call *blind retransmission*. In transmitting a data packet over a unidirectional link, an upstream node always sends the same packet twice. Although the load introduced by the blind retransmission increases in proportional to the number of unidirectional links in paths, the load is not much and it can be negligible for benefits in the smaller delay of path establishment, the higher delivery ratio of data packets, and the shorter delay of packet transmission as shown in Section 3.

2.3 Proactive Path Maintenance

A source node periodically sends proactive forward ants at the rate according to the data sending rate, that is, one ant every n_b^{proact} data packets, to maintain established paths and find better or alternative paths.

A proactive forward ant basically probes an established path by choosing the next-hop node by Eq. (1). It collects up-to-date information about the established path to refresh pheromone values of the path by a corresponding backward ant. In addition, a proactive forward ant is broadcast with a small probability p_b^{proact} at each intermediate node to explore a network for a better or alternative path. If a neighbor node receiving a proactive forward ant does not have the pheromone for the destination node, it then broadcasts the proactive forward ant again as far as the number of broadcasting is within the predetermined limit n_b^{proact} . In this way, a new path is found along the current path, but with a disjoint part, to offer a better or alternative path.

2.4 Route Repair and Detection of Link Failure by Ants

A link failure can be detected by a loss of packet sent to a neighbor over a bidirectional link or a mechanism of exchanging hello messages. If a node does not receive hello messages from a neighbor node to which data packet to a destination node are sent, i.e., a corresponding entry exists in a routing table, for a certain amount of time, defined as $t_{hello} \times allowed\text{-}hello\text{-}loss$, it considers that the link is disconnected.

In the case of a unidirectional link, a node cannot expect reception of hello messages from a downstream node at all. Therefore, a downstream node is not in a neighbor table, whereas a routing table has an entry for the downstream node. In addition, a node cannot detect a link failure by a loss of a packet either, since it cannot tell whether a packet is successfully received or not for the incapability of receiving an acknowledgement from the downstream node. Therefore, we use a proactive forward ant to confirm the stability and detect a failure of a unidirectional link.

When a proactive forward ant is sent over a unidirectional link, a node deposits its identifier in a *probing table*. If the node receives any of corresponding backward ant, broadcast backward ant, or update ant for n_{conf} times, the entry is removed. On the other hand, if the node does not receive n_{conf} ants for a certain amount of time T_{conf} , it concludes that the unidirectional link is disconnected.

When a node detects a failure of a link, it first removes the neighbor node of the other end of the failed link from a neighbor table and all associated entries from a routing table. Then, it starts either path repairing or path clearing depending on the reason of detecting the failure and the existence of alternative paths.

If the only path to a destination is lost due to a link failure which is detected by a failure of data packet transmission, a node which detected the failure first buffers data packets and then tries local repair of the path, while the preceding nodes still keeps sending data packets. The node broadcasts a path repair ant which explores an alternative path to a destination like a reactive forward ant. A path repair ant follows pheromones. If it reaches the node with no pheromones, it is broadcast. The number of broadcasting is also limited by the maximum value n_b^{repair} . After sending a path repair ant, the node waits for a reply, i.e., a backward ant, for a certain amount of time T_{repair} . If it does not receive any, buffered packets are discarded and a link failure notification message is broadcast.

On the other hand, if a reason of detection of a link failure is not a loss of data packet, the node detecting the failure broadcasts a link failure notification message to inform neighbor nodes of the link failure to clear path information. A link failure notification message contains addresses of destinations, to which the node lost the best or only path. The message also has new estimations of the delay and the number of hops to destinations, indicating the lost paths by invalid values, i.e., -1 . Nearby nodes receiving the notification update their routing tables accordingly, and then send link failure notification messages if needed.

3 Simulation Experiments

In this section, we evaluate the performance of our proposal from several aspects listed in 3.1.2. For comparison purposes, we also conduct simulation experiments with the proposal without blind retransmission (denoted as proposal-1), the proposal with blind retransmission (denoted as proposal-2), AODV [10], and AntHocNet [2].

3.1 Simulation Settings

3.1.1 Simulation Environment

Simulation experiments are performed with GloMoSim [15], which is widely used to evaluate the

performance of wireless networks.

We consider three simulation scenarios, i.e., static network of homogeneous nodes, static network of heterogeneous nodes, and dynamic network of heterogeneous nodes. In all scenarios, each simulation experiments is 900 seconds long in simulation time units. A source node and a destination are randomly chosen to initiate a CBR (constant bit rate) session. In a session, a source node generates a CBR traffic of 64-byte packet per second, with the same transmission rate as in [2].

A source node begins sending packets at a time randomly chosen from 10 second to 180 second. At first, the node emits a reactive forward ant or a RREQ message and tries to establish path. While waiting for a backward ant or a RREP message for a certain amount of time, data packets are temporarily buffered. After paths are established, buffered packets are sent to the destination. Then, a source node keeps sending packets until 800 second. If path establishment fails for three times, the buffered packets are discarded and the path establishment is aborted. If another packet comes from an application layer, a new reactive path establishment is initiated. A two-ray pathloss model is used as a radio propagation model. The radio noise is not taken into account. As a MAC layer protocol, IEEE 802.11b DCF The parameter setting is summarized in Table 1. Most of parameters are set according to [2]. Parameters specific to the proposal are set empirically.

3.1.2 Performance Measures

To evaluate the performance of routing protocols, we use the following measures.

Ratio of Successful Path Establishment The ratio of successful path establishment corresponds to the connectivity of a protocol and is defined as;

$$\frac{S_{suc}}{S_{all}}, \quad (4)$$

where S_{suc} is the number of sessions which successfully established a path to a destination, and S_{all} is the number of all sessions in a simulation experiment.

In the case of the proposal and AntHocNet, path establishment is considered successful when a source node receives a backward ant for a reactive forward ant and data packets sent from a source node are received at a destination. In the case of AODV, the case that a source node receives a RREP for a RREQ and data packets sent from a source node are received at a destination is considered successful path establishment.

Path Establishment Delay The path establishment delay is defined as the average of time between the transmission

of the first reactive forward ant or the first RREQ and the reception of the first backward ant or the first RREP at a source node.

Delivery Ratio of Packets Measure related to the reliability of communication, i.e., the delivery ratio R_{deliv} are defined as;

$$R_{deliv} = \frac{P_{rx}}{P_{tx}} \quad (5)$$

where P_{tx} is the number of packets that a source node generated at one packet per second, and P_{rx} is the number of packets that a destination node received.

End-to-End Packet Delay The end-to-end packet delay is defined as the average of time which data packets take from a source node to a destination node.

Control Overhead per Data Packet The load of control messages per data packet is defined as;

$$\frac{n_{ctrl}}{n_{data}}, \quad (6)$$

where n_{ctrl} is the number of transmissions of control packets in the whole network, and n_{data} is number of data packets received at all destination nodes .

3.2 Simulation Results

In the following, we show results scenario by scenario. Values averaged over 40 experiments are shown.

3.2.1 Scenario 1 : Static Network of Homogeneous Nodes

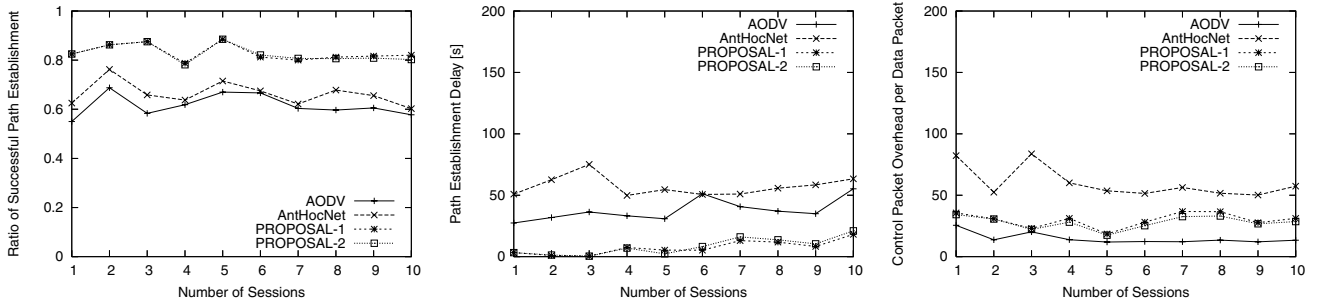
In this scenario, 100 nodes are randomly placed in an area of $3000m \times 1000m$. All nodes are homogeneous, where the transmission range of all nodes is identical and $300m$. All nodes are static and do not move. The number of CBR sessions is changed from 1 to 10. Since there is no unidirectional link in this scenario, we conduct simulation experiments with proposal-1, AntHocNet, and AODV. Details are not shown due to space limitation. Since no unidirectional link exists, proposal-1 shows similar results as AntHocNet.

3.2.2 Scenario 2 : Static Network of Heterogeneous Nodes

In this scenario, 100 nodes are randomly placed in an area of $1800m \times 600m$. 30 nodes have the transmission range of $300m$, 40 nodes have the transmission range of $150m$, and 30 nodes have the transmission range of $100m$. All nodes are static. The number of CBR sessions is changed from 1

Table 1. Common parameter setting in simulation experiments

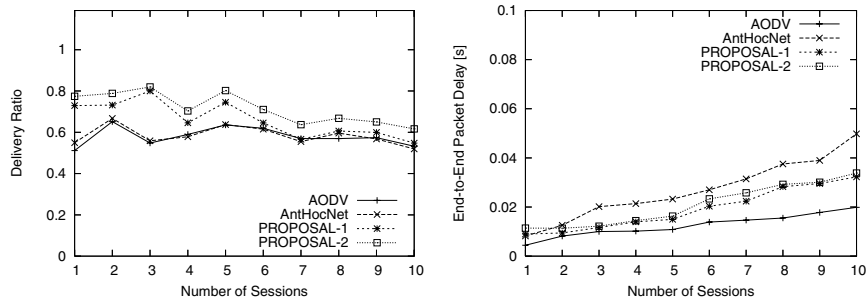
t_{hello}	interval of hello messages	1 [s]
$allowed_hello_loss$	the number of loss of hello message to detect link failure	2
β_1	parameter for ant in stochastic forwarding	1
β_2	parameter for data in stochastic forwarding	2
a_1	acceptance factor for the same first hop	0.9
a_2	acceptance factor for the different first hop	2.0
T_{hop}	time taken for one hop	3 [ms]
γ	smoothing parameter	0.7
$n^{proactive}$	sending rate of proactive forward ants	5 [packet / data]
$p_b^{proactive}$	broadcast probability of proactive forward ant	10 [%]
$n_b^{proactive}$	the maximum number of broadcast of proactive forward ant	2 [hops]
n_b^{repair}	the maximum number of broadcast in path repair	2 [hops]
T_{repair}	waiting time for repair	50 [ms]
TTL	TTL of reactive forward ants	35
n_b^{detour}	the maximum number of broadcast in detouring	5 [hops]
n_{conf}	the number of ants needed to remove an entry of the probing table	2
T_{conf}	waiting time to detect a failure of a unidirectional link	1000 [ms]



(a) Ratio of successful path establishment

(b) Average path establishment delay

(c) Control overhead per Data Packet



(d) Packet delivery ratio

(e) Average end-to-end packet delay

Figure 3. Result of Scenario 2

to 10. Proposal-1, proposal-2 (with blind retransmission), AntHocNet, and AODV are evaluated.

Figure 3 (a) shows the successful ratio of path establishment. The proposals outperform AODV and AntHocNet,

because they can use unidirectional links in path establishment.

Figure 3 (b) of the average delay of path establishment shows the proposals can establish a path faster than the others. The proposals can effectively use unidirectional links, but the others have to explore a network to find a path consisting of bidirectional links and they retry path establishment. As a result, the overhead of control packets in the proposal is reduced as shown in Fig. 3 (c)

The packet delivery ratio in Fig. 3 (d) shows that proposal-2 provides the best performance among protocols.

Figure 3 (e) shows the average of the end-to-end packet delay. The delay of all protocols is proportional to the number of CBR sessions, i.e., traffic of the network, where a network gets congested. The delay of AntHocNet and the proposals are a little worse than AODV. With the proposals, the end-to-end delay is lower than AntHocNet because packets take a shorter path through unidirectional links. However, because of the network load by control overhead, the delay is larger than that of AODV.

3.2.3 Scenario 3 : Dynamic Network of Heterogeneous Nodes

In this scenario, we use the same setting as scenario 2 except for the mobility of nodes and the number of CBR sessions. The number of CBR sessions is fixed at 10. Nodes have the mobility following the *random way-point* (RWP) model. In this scenario, the pause time is set at 30 seconds, the minimum speed is set at $0m/s$, and the maximum speed is changed from $10m/s$ to $50m/s$.

In this scenario, all protocols could successfully establish a path at least once, since nodes moved and there was at least one chance to establish a path to a destination node.

The average delay of path establishment of proposal-2 is much shorter than the others as shown in Fig. 4 (a). A source node can quickly establish a path to a destination node even if both are moving at the high speed of $50m/s$. On the other hand, the average end-to-end delay of data packets is slightly larger than that of AODV as shown in Fig. 4 (d).

Proposal-2 outperforms the others in terms of the delivery ratio as Fig. 4 (c) shows. Regardless of protocols, all ratios decreases as the node speed increases.

Figure 4 (b) shows the control overhead. The overhead of proposal-2 is comparable to that of AODV. The reasons are that the delivery ratio of proposal-2 is high, i.e., the number of data packets is large, and the number of control packets is kept small by quick and easy path establishment.

4 Conclusion and Future Works

In this paper, we have proposed a new routing protocol by detouring around a unidirectional link, detecting link failures by proactive forward ants, and blind retransmission. Simulation experiments showed that our proposal achieved the higher network connectivity, the higher delivery ratio, and the shorter path establishment delay, which are desirable for MANETs. It was also shown that the control overhead per data packet is reduced by easier path establishment and more data packets are received by a destination node.

Some research issues still remain. First, we will consider more efficient detouring by using pheromones left in the network. We also would like to evaluate AntHocNet and the proposal with other parameter settings. For example, we expect that smaller n_b^{proact} leads to more frequent update of paths and faster detection of failure of unidirectional links.

Acknowledgment

This research was partly supported by “New Information Technologies for Building a Networked Symbiosis Environment” (The 21st Century Center of Excellence Program) and a Grant-in-Aid for Scientific Research (A)(2) 16200003 of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

- [1] G. D. Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communication networks. *Journal of Artificial Intelligence Research*, 9:317–365, July-December 1998.
- [2] G. D. Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications (ETT)*, pages 443–455, September-October 2005.
- [3] M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(9):851–871, June 2000.
- [4] M. Gunes and O. Spaniol. Routing algorithms for mobile multi-hop ad-hoc networks. In *Proceedings of NGNT*, pages 20–24, Oct. 2002.
- [5] M. Heissenbüttel and T. Braun. Ants-based routing in large-scale mobile ad-hoc networks. In *Proceedings of KiVS '03*, pages 91–99, Feb. 2003.
- [6] J. G. Jetcheva and D. B. Johnson. On-demand multicast routing in ad hoc networks with unidirectional links. Technical report, School of Computer Science, Carnegie Mellon University, Dec. 2004.
- [7] M. K. Marina and S. R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proceedings of MobiHoc 2002*, pages 12–23, June 2002.

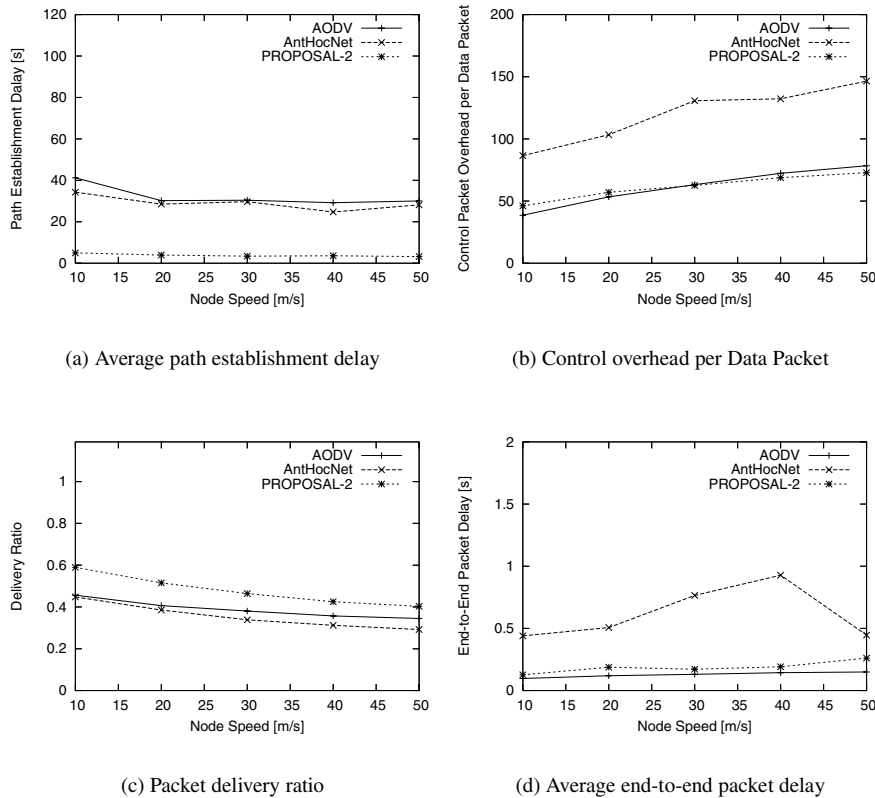


Figure 4. Result of Scenario 3

- [8] S. Mueller, R. P. Tsang, and D. Ghosal. Multipath routing in mobile ad hoc networks: Issues and challenges. *Performance Tools and Applications to Networked Systems*, 2965:209–234, Apr. 2004.
- [9] S. Nesargi and R. Prakash. A tunneling approach to routing with unidirectional links in mobile ad-hoc networks. In *Proceedings of IEEE ICCCN*, pages 16–18, Oct. 2000.
- [10] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector (AODV) routing, July 2003. RFC 3561.
- [11] R. Prakash. A routing algorithm for wireless ad hoc networks with unidirectional links. *Wireless Networks*, 7(6):617–625, Nov. 2001.
- [12] D. M. Shrestha and Y.-B. Ko. A new routing protocol in ad hoc networks with unidirectional links. In *Proceedings of IWDC'05*, Dec. 2005.
- [13] K. Venkataramanan, D. Aravindan, and K. Ganesh. On demand routing protocol to support unidirectional links in mobile ad hoc networks. In *Proceedings of ICOIN 2004*, pages 144–153, Feb. 2004.
- [14] J. Walter, J. Welch, and N. Vaidya. Unidirectional links prove costly in wireless ad hoc networks. In *Proceedings of DIALM 98*, Oct. 1998.
- [15] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Proceedings of PADS '98*, pages 154–161, May 1998.