Proceedings of APCC2015 copyright © 2015 IEICE 14 SB 0087

Looping Issues and Solutions in AntHocNet

Shams ur Rahman[1], Jae-Choong Nam[2], Geon-Hwan Kim[2], You-Ze Cho[2], and Ajmal Khan[3]
Kyungpook National University, Daegu, Korea[1,2]
IQRA National University, Peshawar, Pakistan[3]
shamsuet@gmail.com[1]
{jch-nam, kgh76, yzcho}@ee.knu.ac.kr[2]
engajmal@yahoo.com[3]

***Abstract*— Routing is one of the most challenging issues related to mobile ad hoc networks. Researchers' attention is increasingly being attracted toward bio-inspired routing protocols, a representative of which is AntHocNet. This protocol incorporates congestion avoidance into its path construction mechanism and attempts to construct and maintain multiple paths. The source node and each intermediate node along the path forward data to the next hop stochastically such that the probability of a better path being chosen is high. However, the combination of multiple paths and the broadcasting of route-construction (or route repair) packets sometimes results in the formation of loops, which seriously degrades the performance of the protocol. In this paper, we identify some loop formation scenarios and propose solutions thereto. We also present some simulation results that compare the performance of the original AntHocNet with one in which the proposed solutions are incorporated.**

Keywords*— *MANETs; routing; bio-inspired; ant colony optimization; AntHocNet; loops

I. INTRODUCTION (*HEADING 1*)

Mobile ad hoc networks (MANETs) have attracted increasing research attention over the last few decades. A MANET is a network of mobile nodes communicating over a wireless medium. Because of the network's ad hoc nature, there is no infrastructure and, therefore, the mobile nodes themselves have to perform the routing function. Ad hoc networks are self-organizing and adaptive. On the one hand, the infrastructure-less nature of MANETs brings with it many advantages, such as allowing rapid deployment, but on the other, the dynamic nature of the topology of MANETs renders routing a challenging issue. Mobile nodes that are within transmission range of each other can communicate directly. However, for communication among nodes that are not in the transmission range of each other, intermediate nodes have to forward the packets toward the destination. A large number of routing protocols for MANETs have been proposed [1]. Researchers have also looked to nature for finding solutions to the problem of routing [2, 3] and a number of bio-inspired routing protocols have been proposed. AntHocNet [4, 5] is a representative bio-inspired protocol designed for MANETs. In this paper, we identify some loop formation scenarios in AntHocNet and propose solutions thereto. In the end we show some simulation results, which show that the performance of AntHocNet is considerably improved when the loop formation is prevented.

II. OVERVIEW OF ANTHOCNET

AntHocNet is a routing protocol inspired by the foraging behavior of ants. AntHocNet has both reactive and proactive features.

A. Reactive path setup

The source node broadcasts a reactive forward ant (FANT). Every intermediate node that receives a reactive FANT broadcasts the ant further if it has no path to the given destination; otherwise, it unicasts the ant by choosing the next hop stochastically according to

$$P_{nd} = \frac{(\mathrm{T}_{nd}^{i})^{\beta_1}}{\sum_{j \in N_d^i} (\mathrm{T}_{jd}^{i})^{\beta_1}}, \beta_1 \geq 1$$

where T_{nd}^{i} is the amount of pheromone on the link from *i* to *n* for destination *d*, while N_{d}^{i} is the set of neighbors of *i* over which a path to the destination *d* is known. The reactive FANT maintains a list of its visited nodes. When the ant reaches the destination, it is converted to a backward ant (BANT). The BANT takes the same path as its corresponding FANT, but in the reverse direction. The BANT sets up a path to the destination at every intermediate node by creating or updating an entry T_{nd}^{i} in the routing table. This entry indicates the goodness of traveling over next hop *n* toward destination *d* from the current node *i*, and is computed using

$$\tau_d^i = \left(\frac{\hat{T}_d^i + hT_{hop}}{2}\right)^{-1}$$

$$\mathrm{T}_{nd}^{i} = \gamma \mathrm{T}_{nd}^{i} + (1-\gamma)\tau_d^i, \gamma \in [0,1]$$

Where $\hat{\mathrm{T}}_{d}^{i}$ is the estimate of the time it would take a data packet to reach from the current node *i* to the destination *d*.

B. Stochastic data routing

Multiple paths are constructed from the source to the destination during the path setup phase and data are forwarded along these paths according to their goodness indicated by the pheromone values in the routing table. At each node, the next hop is chosen stochastically using

$$P_{nd} = \frac{(T^i_{nd})^{\beta_2}}{\sum_{j \in N^i_d} (T^i_{jd})^{\beta_2}}, \beta_2 \ge \beta_1$$

This stochastic data routing results in automatic load balancing with better and less congested paths having higher probability of being chosen.

C. Proactive path maintenance

Another type of ant, called proactive FANT, is regularly launched while the data session is running. In general, proactive FANTs are unicast by the intermediate nodes in the same manner as reactive FANTs. However, there is a small probability (0.1) that an intermediate node may decide to broadcast a proactive FANT. While unicasts result in existing paths be sampled and refreshed, a broadcast causes the exploration of new paths. However, a proactive ant cannot be broadcast more than twice, effectively controlling the flooding of the network with proactive FANTS. Nodes regularly broadcast hello messages that help each node determine its neighbors. If a node misses two consecutive hello messages from a neighbor, the node assumes it has lost that neighbor and removes its entry from the routing table.

D. Link failures

The loss of a neighbor indicates a link failure and therefore the node notifies this fact to all its neighbors. If a link fails during an ongoing data session, the node looks for alternate routes. If no alternate route is available, the node attempts to repair the route locally by broadcasting a route repair FANT toward the involved destination. Meanwhile, packets destined for the involved destination are buffered. If the local link repair is successful, the buffered packets are forwarded along the repaired path; otherwise, the packets are dropped and a link failure notification is broadcast.

III. Looping issues with AntHocNet

In AntHocNet, the distributed nature of the path setup mechanism, the construction and use of multiple paths, and the broadcasts of FANTS provide a recipe for loop formation. Any of the three types of ants – reactive, proactive, and route repair – can cause the formation of loops. Loops can be formed by the indirect interaction of ants of the same generation, as well of different generations.

A. Loops formed by forward ants of the same generation (reactive and route-repair forward ants)

Consider a network in which a source node *S* launches a FANT, reactive or route repair, toward a destination *D*. Since the ant is broadcast by the source node and it may also be broadcast by other nodes, a number of copies of the same ant, or, in other words, a number of FANTs of the same generation, travel along various paths.

1) Single-hop loop

Figure 1 shows the paths taken by two reactive FANTs of the same generation. The red ant traversed nodes m_1,m_2,m_3,m_4, whereas the blue ant traversed nodes n_1,n_2,n_3,n_4 on their way from source S to destination D. Nodes n_2 and m_2 are visited by both ants but in the reserve order, which results in a loop spanning these two nodes (we call this a single-hop loop).

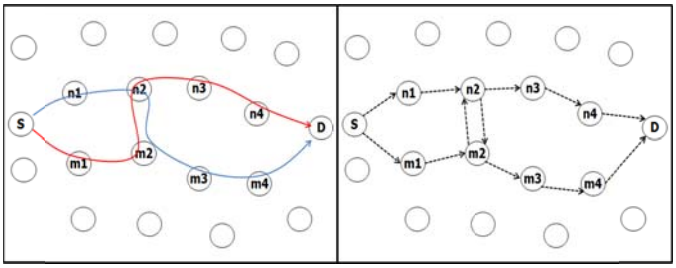

Fig. 1. Single-hop loop formation by ants of the same generation.

2) Multi-hop loop

In Figure 2 the path taken by the reactive FANTs of the same generation is slightly different from that shown in Figure 1. Again, nodes n_2 and m_2 are visited by both ants and in reverse order, but here n_2 and m_2 are not neighbors, or, more importantly, the visits to these two nodes by the two ants are not consecutive. Here, a loop is formed that spans four nodes – m_2,a,n_2,b – and hence, it is a multi-hop loop. The probability of such multi-hop loops forming is, however, very low as compared to single-hop loops.

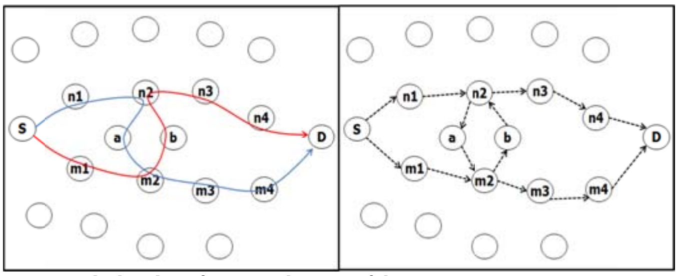

Fig. 2. Multi-hop loop formation by ants of the same generation.

Solution:

If the loop formed in this fashion is single-hop, i.e., it involves only two nodes, it can be prevented as follows. Before forwarding a BANT to the next hop, each node checks in its routing table whether it has a path to the involved destination (the node from which the BANT has arrived) that passes through the next hop of the BANT. If such a path exists, the node does not forward the BANT further. This prevents single-hop loops from forming. However, multi-hop loops can still form.

B. Loop formation by forward ants of different generations (route-repair and proactive forward ants)

1) Single-hop loop

Consider again a network in which a source node S has already established some paths to a destination D. Suppose three such paths are

P_1: $n_1, n_2, n_3, n_4, n_5, n_6, n_7 \ldots\ldots n_k, D$

P_2: $m_1, m_2, m_3, n_4, n_5, n_6, n_7 \ldots\ldots n_k, D$

P_3: $m_1, m_2, m_3, m_4, m_5, m_6, m_7 \ldots\ldots m_p, D$

Note that P_1 and P_2 merge at node n_4. Now, suppose that a proactive FANT is travelling along P_1 and it happens to be broadcast at n_4. Since m_3 is a neighbor of n_4, it picks up the broadcast and forwards the FANT to the next hop m_4 (along P_3). The proactive FANT travels along the rest of P_3 and reaches the destination D. Now, if the corresponding BANT successfully arrives back at source S, a loop is formed between m_3 and n_4.

2) Multi-hop loop

Consider the slightly different paths shown below for multi-hop loop formation.

P_1: $n_1, n_2, n_3, n_4, n_5, n_6, n_7 \ldots n_k, D$

P_2: $m_1, m_2, m_3, u, n_4, n_5, n_6, n_7 \ldots n_k, D$

P_3: $m_1, m_2, m_3, m_4, m_5, m_6, m_7 \ldots m_p, D$

Again, suppose that while data packets are in transit along P_1, the link between n_4 and n_5 breaks. Node n_4 attempts to locally repair the path and therefore it broadcasts a route repair FANT. Suppose that a node v receives this broadcast and rebroadcasts it further. This rebroadcast is received by m_3. Now, from here on the FANT can travel along the rest of P_3 all the way to the destination. In this case the loop formed spans the nodes m_3, u, n_4, v.

Such loops are formed in the case of proactive ants if, for example, a proactive FANT, having travelled along P_1, is broadcast at n_4 and travels to the destination just like the route repair ant.

Solution:
Single-hop loops thus formed can be prevented. Before forwarding a route-repair (or proactive) FANT, each node checks whether it (the node) has a path to the involved destination passing through the node from which the ant arrived; if yes, then the node drops the ant.

C. *Loop formation (by route-repair forward ant): when a broadcasted forward ant is received by an upstream node, including the source node, that has multiple paths to the involved destination.*

1) Single-hop loop

A single-hop loop may form when the upstream node, with multiple paths to the destination, is a neighbor of the node of which the downstream link to the destination breaks.

Suppose we have the following paths from a source S to a destination D.

P_1: $n_1, n_2, n_3, n_4, n_5, n_6, n_7 \ldots n_k, D$

P_2: $m_1, m_2, m_3, m_4, m_5, m_6, m_7 \ldots m_p, D$

P_3: $m_1, m_2, r_3, r_4, r_5, r_6, r_7 \ldots r_i, D$

Notice that there are two paths from node m_2 to destination D (P_2 and P_3 span the same nodes up to m_2 and then they branch). Now, suppose that some link, downstream from m_2, along P_2 breaks; say, the link between m_3 and m_4 breaks during an ongoing data session. Node m_3 broadcasts a route repair FANT. Since m_2 is a neighbor of m_3, it also receives the broadcast and forwards it to r_3. Now, the FANT can travel along the rest of P_3 all the way to the destination D. When the corresponding BANT reaches m_3, a loop is formed between m2 and m3.

2) Multi hop loop

Such loops could also have formed, although with much smaller probability, even if m_2 and m_3 had not been neighbors. A loop would also have formed if the source S had received this ant and forwarded it along the path P_1. When a link breaks near the source node and the source node has multiple paths to the involved destination, the probability of such loops forming is high.

Solution:
Single-hop loops and loops that involve the source node of data packets can be eliminated. To eliminate single-hop loops, a node that receives a broadcast FANT first checks whether it has a path to the involved destination passing through the node that broadcast the ant (the source node of the ant). If such a path exists, the node drops the ant.

All such loops, single- or multi-hop, that involve the source of data packets can also be eliminated by including its address in the route repair FANT. However, loops that do not involve the source node of data packets and span multiple hops cannot be prevented, although, as mentioned earlier, the probability of such loops forming is very small.

IV. Performance Evaluation

AntHocNet was implemented with and without the incorporation of the proposed loop prevention solutions. Here, we call AntHocNet with loop prevention LP-AntHocNet. In this section we compare the performance of the two protocols.

A. *Simulation Environment*

Simulations were carried out using ns-2. A total of 50 nodes were deployed in an area of 1500x300 m2. For each scenario, five simulations were run and then the results were averaged. Each simulation was run for 600 s. The traffic type was CBR. At the MAC layer, IEEE 802.11 DCF was used. The propagation model used was two-ray ground. Transmission range was maintained at 250 m.

B. *Simulatin Results*

Figure 3 shows a comparison of the delivery ratio of AntHocNet and LP-AntHocNet. As the pause time increases, the delivery ratio for both AntHocNet and LP-AntHocNet increases. However, LP-AntHocNet performs much better than AntHocNet for all pause times. The average delivery ratio for LP-AntHocNet is greater than 90% for all sample intervals, whereas that of the pure AntHocNet slightly rises above 80% for higher pause times.

Figure 4 shows a comparison of the average end-to-end delay. As the pause time increases, the end-to-end delay for both protocols drops. For smaller pause times, LP-AntHocNet performs considerably better than AntHocNet.

In Figure 5, the normalized overhead is compared and again LP-AntHocNet outperforms AntHocNet for all pause times. For smaller pause times, the performance of LP-AntHocNet is again considerably better than that of AntHocNet.

Figure 6 shows the frequency of occurrence of the three types of loops for different pause times. Type-A loops are caused by reactive and route-repair ants and the figure shows that the frequency of such loops is almost negligible. Type-B loops are mainly caused by proactive ants (and occasionally by route repair ants). The figure shows that variation in pause time has little impact on the frequency of these loops. This behavior can be explained by the fact that such loops are caused mainly by proactive ants which are launched by data sources at regular interval irrespective of the pause time. Type-C loops are caused by route repair ants. The higher the pause time, the smaller the number of route breakages and therefore the smaller the number of route repair ants launched, and vice versa. Therefore, as the pause time increases, the frequency of type-C loops decreases. Overall, the total frequency decreases as the pause time increases. It is worth mentioning here that performance degradation depends not only on the frequency of loop formation but also on the duration for which a loop remains intact, and on the goodness of the path along which the loop occurs.

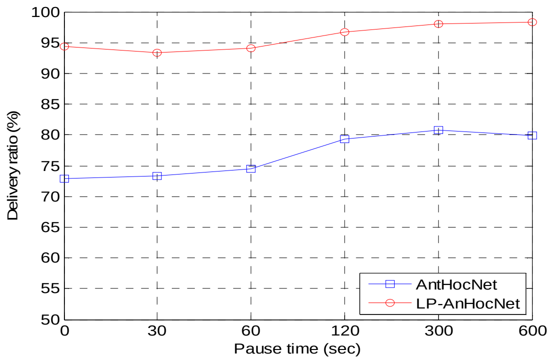

Fig. 3. Comparison of the packet delivery ratio.

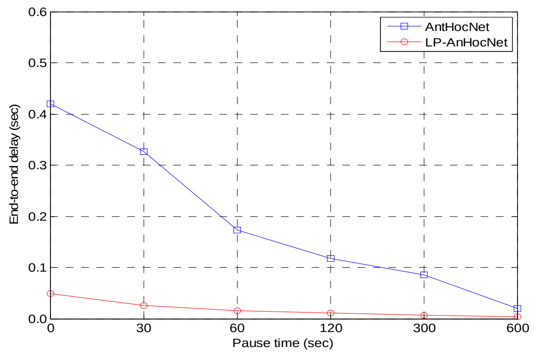

Fig. 4. Comparison of the average end-to-end delay.

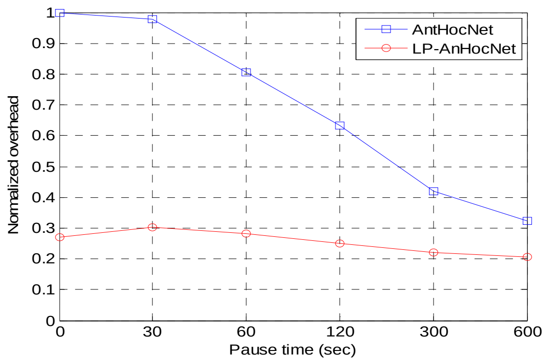

Fig. 5. Comparison of the overhead.

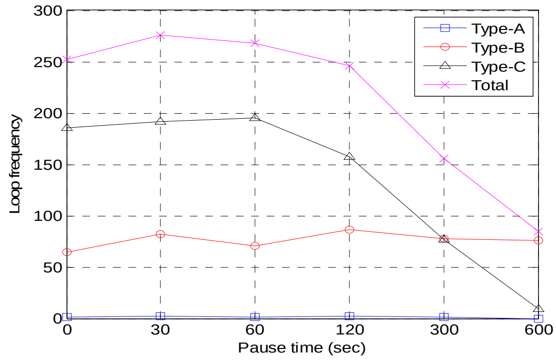

Fig. 6. Loop formation frequency.

V. CONCLUSIONS

In this paper, the looping issues that occur in AntHocNet were identified and solutions for preventing the formation of such loops were proposed. Simulation results indicated that looping is a serious problem in AntHocNet and degrades the performance significantly. The results showed that the performance can be greatly improved when the loop prevention is implemented.

VI. ACKNOWLEDGEMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2013R1A1A4A01012534), and the IT R&D program of MSIP/IITP [10041145, Self-Organized Software platform (SoSp) for Welfare Devices].

VII. REFERENCES

[1] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, and D. Turgut, "Routing protocols in ad hoc networks: A survey," *Computer Networks*, vol. 55, pp. 3032-3080, 2011.

[2] F. Ducatelle, G. Di Caro, and L. Gambardella, "Principles and applications of swarm intelligence for adaptive routing in telecommunications networks," *Swarm Intelligence*, vol. 4, pp. 173-198, 2010/09/01 2010.

[3] R. H. Jacobsen, Q. Zhang, and T. S. Toftegaard, "Bioinspired Principles for Large-Scale Networked Sensor Systems: An Overview," *Sensors* (Basel, Switzerland), vol. 11, pp. 4137-4151, 2011.

[4] G. Di Caro, Ducatelle, F. and Gambardella, L. M., "AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks," *Eur. Trans. Telecomm.*, vol. 16, 2005.

[5] G. D. Caro, F. Ducatelle, and L. M. Gambardella, "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks," Dalle Molle Institute for Artificial Intelligence Galleria, Manno, Switzerland IDSIA-27-04-2004, 2004.