International Journal of Future Generation Communication and Networking
Vol.7, No.5 (2014), pp. 137-150
http://dx.doi.org/10.14257/ijfgcn.2014.7.5.12

Adaptive Distributed Gateway Discovery with Swarm Intelligence in Hybrid Wireless Networks

Gongjing Zhang and Jingwei Hu

College of Information Engineering, Qingdao University, Qingdao 266071, China
zhang8887327@163.com, 328704547@qq.com

Abstract

Hybrid Wireless Networks are emerging as a promising new technology, benefiting from both fixed and ad-hoc networks technologies whilst alleviating some critical problems in these networks. Gateway discovery is one of the major components in order to realize a HWN，and it has to be optimized in order to release wireless resources for the data transmissions. In this sense, we apply biologically inspired metaphor to the Gateway Discovery problem in these networks, and present a new approach-Adaptive Distributed gateway Discovery with Swarm Intelligence (ADDSI). Simulations have been conducted to demonstrate the performance of the proposed approach and to compare it with certain existing schemes in terms of packet delivery ratio, average end-to-end delay and normalized routing overhead.

Keywords: Hybrid Wireless Network; gateway discovery; swarm intelligence

1. Introduction

An ever-growing demand for coverage extension, higher data-rates and improved connectivity has motivated the interconnection of mobile ad-hoc networks to fixed IP networks to form a Hybrid Wireless Network (HWN). Such networks not only preserve the benefits of conventional infrastructure-based networks, where the service infrastructure is constituted of fixed entities (gateway), but also incorporate the flexibility of ad-hoc networks, where wireless transmission through mobile stations in multi-hops is allowed [1]. HWNs seem attractive in opening new business opportunity for network operators and service providers, enabling commercial services provision with broad coverage on the one hand and seamless mobility for mobile clients with improved overall QoS on the other hand.

Gateway (GW) facilitates the Mobile Nodes (MN) to connect with the core IP network. MN, which is not in the direct coverage of GW, discovers the route to GW in multi-hop fashion i.e. passing through other MNs in order to connect to the infrastructure network. Each set of nodes covered by a GW forms an ad-hoc network. We frequently refer to GW as a means to connect to the Internet, although the GW is flexible enough to offer a wide range of other services as well. For the mobile nodes to connect to the Internet, a gateway discovery mechanism is required for the nodes to discover the route towards the gateway [2]. As compared with infrastructure-based networks, where the gateway is always at a single-hop, the problem of gateway discovery in hybrid networks is challenging and node mobility, scalability of ad hoc networks etc. make it even more complex [2]. Swarm intelligence refers to complex behaviors that arise from very simple individual behaviors and interactions, which is often observed in nature, especially among social insects such as ants. Although each individual (an ant) has little intelligence and simply follows basic rules using local information obtained from the environment, such as ant's pheromone trail laying and following behaviors, globally optimized behaviors, such as finding a shortest path, emerge

ISSN: 2233-7857 IJFGCN

when they work collectively as a group. In this paper, we apply this biologically inspired metaphor to the Gateway Discovery problem in Hybrid Wireless Networks (HWNs).

The main technical contributions of our work can be summarized as follows:

a) We advocate a totally distributed approach-ADDSI (Adaptive Distributed gateway Discovery with Swarm Intelligence) in HWNs.

b) We propose a novel mobility prediction model to realistically estimate the mobility strength of a concerned node. The proposed estimation model is integrated into ADDSI, and the node's mobility prediction value is used as the route selection criterion. With the help of this new mechanism, ADDSI discovers available stable routes, establishes more efficient multicast routes and is better adapted to network topology changes.

c) After the forwarding set initialization, each data source periodically deploys a small packet that behaves like an ant to opportunistically explore different paths to the GW. This exploration mechanism enables the new scheme to discover new forwarding set that yields lower total forwarding costs under some limitations i.e. mobility threshold.

d) The ADDSI algorithm is inspired by a hypothesis that the gateway advertisements should only be targeted to those nodes which are looking for the gateway (such as active data sources); and other nodes should not be hampered with the periodic gateway advertisements.

e) We evaluate ADDSI using the NS-2 simulator and the experimental results prove that our new scheme is more effective. In particular, ADDSI is practical to enhance the dependability of communication, improve packet delivery ratio and reduce the normalized control overhead.

The remainder of this paper is organized as follows. In Section 2, we present the state of the art achievements in gateway discovery mechanisms. After that we propose our novel scheme called Adaptive Distributed gateway Discovery with Swarm Intelligence (ADDSI) in Section 3. Furthermore, simulation-based performance evaluation of ADDSI scheme is presented in Section 4. Finally, we conclude the paper in Section 5.

2. Motivation and Related Works

The interconnection of MANETs and the Internet is supported by an Internet Gateway that provides the necessary configuration parameters to the mobile nodes. Gateways are generally the specialized nodes, which lie in the ad-hoc network and also have connectivity with the fixed network, such as Internet. With the help of gateway discovery mechanisms, nodes in the mobile ad-hoc network discover the route towards the gateway for different reasons, for instance, to communicate with a node in the fixed network or to access the Internet or Intranet.

In the existing literature, different proposals for gateway discovery have used either reactive or proactive mechanisms [2]. In proactive approaches, the gateway periodically sends the Gateway Advertisements (GWADV) messages, which are flooded throughout the entire ad-hoc network. The GWADV is employed to update the route to the Internet Gateway. The interval of emission of these messages is set to a fixed value (*T*). When a stored route is used, the node can detect that it is no longer valid. Alternatively, in reactive approaches, the nodes which require connectivity to the gateway broadcast the Gateway Solicitation (GWSOL) messages. These solicitation messages are responded by the gateways.

In the recent studies, according to the global connectivity support, the hybrid gateway discovery could be considered as the combination of the two previous schemes. This algorithm follows a proactive scheme in an area close to the Gateway. The area can be easily delineated with the time to live (*TTL*) parameter present in the IP header of GWADV messages. Thus, the nodes that are more distant to the Gateway than *TTL* hops should acquire the route to the Gateway on demand (by broadcasting a GWADV) when needed. Some works focus on how to tune the *TTL* parameter conveniently. In [3], the gateway is initially discovered reactively, furthermore, for route maintenance, the scope of the gateway advertisements i.e. *TTL*, is adapted according to the active sources in the network. By simply looking at the IP header of the data packets, the gateway keeps track of the number of hops at which each of its active sources is located. The gateway adaptively selects the *TTL* value so that all the active sources receive the periodic advertisements from the gateway. This approach is referred to as Maximum Source Coverage (MSC). MSC may suffer from suboptimal performance in certain networking scenarios such as when a small number of sources are at a large number of hops from the gateway. Another hybrid approach is presented in [4], in which some intermediate nodes are allowed to answer GWADV messages based on local information.

In previous works, the Internet Gateway is in charge of the previous optimizations as it decides the value of the *TTL* in the GWADV messages. In [5], the authors present a novel strategy in which nodes make a decision about the convenience of retransmitting the GWADV message. Specifically, the work named adaptive distributed gateway discovery (ADD) proposes that just the mobile nodes which are retransmitting data packets to the Internet will be allowed to forward the GWADV messages. Decentralizing the decision about the retransmission of GWADV messages has demonstrated to be effective as more variables could be taken into account [5]. This scheme is the first step towards the distributed adaptation of gateway discovery in HWNs, which causes significant performance gain, as validated by simulation results.

The proposed scheme [6] reduces the flooding processes related to gateway discovery. To do so, the algorithm dynamically adjusts the interval of emission of the gateway advertisement messages to the need for updated routes to the gateway. The tuning is supported by the analysis of the spatial distribution of nodes in the MANET and by the impact that the relative position of nodes has on the route lifetimes.

When a node in MANET wants to obtain a globally routable address for Internet connection, it has to first find a route to a gateway. Under some network conditions, performance of Internet connectivity will be degraded by obvious unidirectional links and broadcast storm due to the blind rebroadcast of gateway discovery packets. An effective hybrid gateway discovery algorithm for MANETs is proposed to remove unidirectional links from global route computation simultaneously and enhance Internet connectivity, in paper [7]. The routing protocol called ad hoc on-demand distance vector (AODV) has been modified to set up support hybrid network. Results using NS2 simulation show this hybrid Internet connectivity approach can provide better performance than others due to avoiding unidirectional links with little overhead.

In [8], the authors propose a novel technique by which the messages that the gateway periodically generates are exclusively forwarded in the areas where the links are expected to remain stable. By this, nodes are able to learn stable routes, and in turn, the messages generated from the breakages of used links are avoided.

In our work, we also follow a decentralized scheme. The decision of retransmitting the GWADV message is locally made in each MANET node. In particular, we propose to incorporate the node's mobility factor to decide on the suitability of forwarding a GWADV message.

3. A New Scheme: Adaptive Distributed Gateway Discovery with Swarm Intelligence (ADDSI) for HWNS

In this paper, we present a novel adaptive gateway discovery scheme called ADDSI (Adaptive Distributed gateway Discovery with Swarm Intelligence), in which the adaptation is done in a fully distributed manner.

3.1. Overview of ADDSI

Our proposed scheme establishes multicast connectivity between the GW and the multiple active data sources. In ADDSI algorithm, initially, the potential data source node reactively discovers the gateway. The GW announces its existence to the multiple active data sources by flooding the network with a gateway advertisement (GWADV), in which the node's mobility prediction value (*mpv*) is considered. The new mechanism follows the rule, the smaller the mobility prediction value is, the more stable the node becomes. Three additional fields, *required mobility prediction value* (*rmpv*), *mobility prediction value* (*mpv*) and *path mobility prediction value* (*path_mpv*), are appended into gateway advertisement by the intermediate nodes while the message travels through the network. We use *rmpv* to represent for the mobility prediction requirement of data packet transmission, which is determined by the GW and remains unchanged during the flood. The field *path_mpv* denotes for the continued product of intermediate nodes' mobility prediction values on this path that the GWADV has passed. It is initialized to 0 by the GW and varies with the transmission of the packet.

When an intermediate node *j* receives a GWADV from a neighbor node *k*,

Step 1: it creates a route entry to node *k* with *path mobility prediction value* as $path\_mpv_{jk}$ ($path\_mpv_{jk}$=max[mpv_j, mpv_k]), if there is no route to *k* in its local routing table.

Step 2: it checks whether one copy of the same GWADV has been received. If a copy of the same GWADV has been received and the later copy has no less *HopCounter* and no smaller *path_mpv* (That is, the later path is, (*a*) farther and bigger mobility or (*b*) farther and equal mobility or (*c*) equal hops and equal mobility or (*d*) equal hops but bigger mobility than the existing paths), the GWADV will be discarded and the procedure ends; otherwise, go to step 3.

Step 3: If node *k* is not the GW, node *j* creates a reverse route to the GW using the previous hop (node *k*) of the GWADV as the next hop. The *path mobility prediction value* of the reverse route entry is set to max[*path_mpv*, $path\_mpv_{jk}$] when $path\_mpv_{jk}$ is known, otherwise the value is max[*path_mpv*, *rmpv*].

Step 4: If a valid route to any of the data source node is available and Sequence Number of the route is greater than the *DestSequenceNo* in the GWADV, node *j* generates an *RREP* to node *k*.

Step 5: Otherwise, node *j* modifies the *path_mpv* of the GWADV using max[*path_mpv*, $path\_mpv_{jk}$] when $path\_mpv_{jk}$ is known, or max[*path_mpv*, *rmpv*] when $path\_mpv_{jk}$ is unknown. Then node *j* increases *HopCounter* by one and propagates the GWADV to all its neighbors.

Each active data source then relies on this advertisement to reactively establish initial connectivity by sending a Join Request back to the GW via the reverse path. The nodes on the reverse paths to the GW will be requested by active sources to serve as *forwarding* or *intermediate* nodes (***forwarding set***), on which gateway advertisement packets are accepted and forward from the GW to the active sources. In addition to the support of gateways, ADDSI also involves the participation of *forwarding set* which is used as a relay to forward the data packets from the sources toward the gateway and utilized in a multicast fashion.

Once the initial *forwarding set* is formed, each active data source attempts to learn a better connection to the GW via using the swarm intelligence mechanism. Once the route towards the gateway is known, the source node starts sending the data packets.

3.2. Mobility Prediction Model

To make data forwarding more effective under mobility [8], while maintaining good efficiency when the network is static, we incorporate a mobility-adaptive mechanism into ADDSI. With this mechanism, each node *i* keeps track of the normalized link failure frequency, denoted by *nlff*(*i*), which reflects the dynamic condition of the area surrounding *i* in terms of the number of link failures per neighbor per second. A calculation of *nlff*(*i*) is performed every *NLFF_TIME_WINDOW* time period as follows:

$$current\ nlff(i) = \frac{f}{NLFF\_TIME\_WINDOW \times |ntab(i)|} \tag{1}$$

$$nlff(i) = [current\ nlff(i) + nlff(i)]/2 \tag{2}$$

Where *f* is the number of link failures detected during the last *NLFF_TIME_WINDOW* time period. Initially *nlff*(*i*) is set to zero.

3.3. ADDSI Description

3.3.1. Local Data Structures: Each node in the network with a unique ID *i* maintains a list of neighboring nodes, *ntab*(*i*), obtained via a neighbor discovery protocol such as periodic hello messaging. The node cost associated with *i* is represented by *cost*(*i*), as any node in a single hop away from *i*, the *cost*(*i*)=1.

Join table: maintains a list of nodes that have requested to flow toward to the gateway via this node. The join table of node *i* is denoted by *Join*(*i*). This table is updated when *i* hears a Join Request packet intended to itself. Each entry in *Join*(*i*) is of the form <*r*>, where *r* is a requesting node's ID that it has sent along with its Join Request. The join table is initially empty for each node. The node *i* becomes a forwarding node as long as *Join*(*i*)≠*Φ*. When a neighbor *j* is removed from *ntab*(*i*) due to a link failure, *i* will remove all the corresponding entries <*j*> from all join tables.

Pheromone table: maps neighboring nodes to pheromone intensities. The pheromone intensity associated to the link (*i*, *j*) seen by *i* is denoted by *τ*(*i*, *j*), where 0<*τ*(*i*, *j*)<1. This table is initially empty. If a neighbor *j* is removed from *ntab*(*i*), all entries *τ*(*i*, *j*) are removed as well.

Best cost table: keeps track of how close the node *i* thinks it is to forwarding nodes in terms of path costs. The cost of the best path to any forwarding node that *i* has seen so far is represented by *bestCost*(*i*). This best cost information is used to determine whether a Backward Ant (described in subsection 3.3.3) has returned from a good path or a bad path. This table is also initially empty.

3.3.2. Forwarding Set Initialization: The Gateway Advertisement contains the Gateway ID (in case of multiple gateways), a sequence number, *rmpv*, *mpv*, *path_mpv* and a cost which is initially set to zero. Upon receiving this Gateway Advertisement, each node *i* discards the message if it has seen an advertisement from the same node with the same sequence number before. Each active data source or forward node then uses this *nlff* to determine the stability of its surrounding area. If its *nlff* is lower than a threshold *NLFF_THRESHOLD*, it will consider its area stable and join the set by sending Join Requests toward its best next hop. For example, as long as an active data source or a current forwarding node *i* keeps hearing Gateway Advertisement from the gateway, it periodically broadcasts a Join Request packet to its neighbors with *TTL=1*. The Join Request packet contains an entry <*k*>, where *k* is defined as:

$$k = \arg\max_{n \in ntab(i)} \sum \frac{\tau(i,n)}{1 + bestCost(i)} \tag{3}$$

The above formula implies that an active source node that is willing to connect the GW should send a request to a neighbor whose goodness was recently confirmed by Backward Ants (*i.e.*, having high pheromone intensity), moreover, the formula potentially meets the limitation of mobility threshold and yields the lowest joining cost. At this moment, however, no actual ant packets are involved and each node has only one entry in its best cost table. Therefore, the initial forwarding set generally consists of all the nodes that are on the paths on which the Gateway Advertisement are forwarded to the active data sources. Figures 1(*a*), 1(*b*), and 1(*c*) illustrate the forwarding set initialization process.

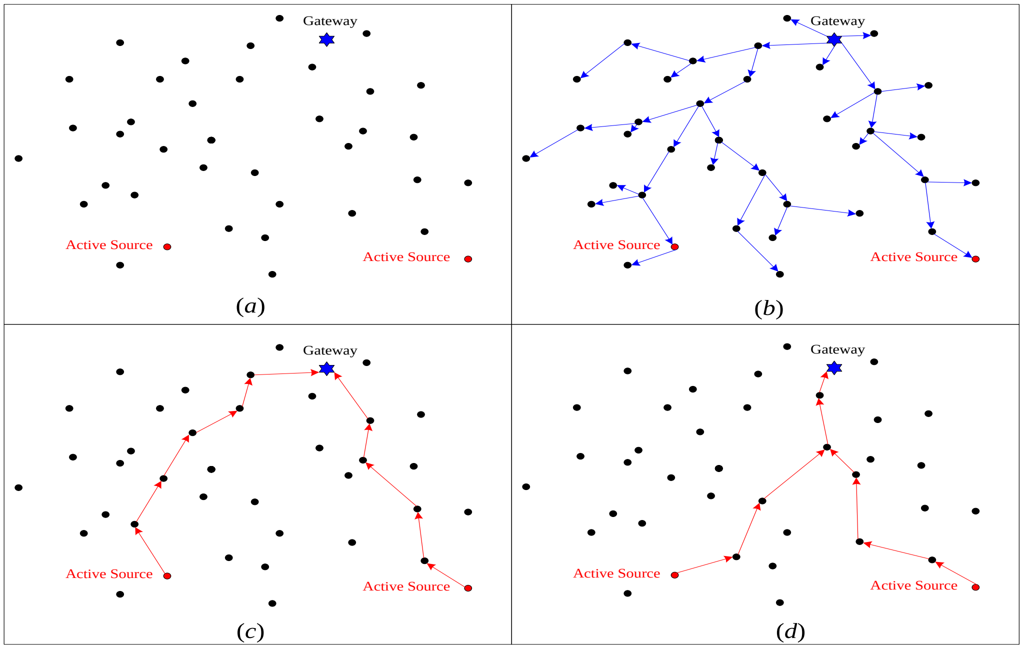


Figure 1. Forwarding Set Initialization and Evolution Process

If *nlff* exceeds the threshold, however, it will add another entry for the second best next hop into its Join Requests. All subsequent Forward Ants will be redirect to other paths, while the majority of them take the next hop whose pheromone intensity was the second highest before the link failure. Since all the neighbors are ranked by their goodness in terms of pheromone intensities, the second best next hop is easily determined. Formally, if *k* is the best next hop for *i* to join the set, as defined in (3), then the second best next hop *k*' is defined as:

$$k' = \arg\max_{n \in ntab(i),\ n \neq k} \sum \frac{\tau(i,n)}{1 + bestCost(i)} \tag{4}$$

Algorithm 1 Procedure ***UpdatePheromoneAndCost***(*next*; *cost*; *detFlag*) executed by node *i*

```
Parameters:
next←neighbor ID indicating which pheromone table entry to be updated
cost←cost of joining the GW at a forwarding node via next node
detFlag←flag indicating whether this update is deterministic
Begin:
 if $\tau(i, j)$ is not defined then
  $\tau(i, j) \leftarrow 0$
 end if
 if detFlag=TRUE then
  bestCost(i)←cost
  $\tau(i, next) \leftarrow \tau(i, next) + \frac{1}{2(1 + \cos t)}$
 else
  if bestCost(i) is not defined OR cost<bestCost(i) then
   bestCost(i)←cost
   $\tau(i, next) \leftarrow 1$ /* set intensity to max */
  else
   $\tau(i, next) \leftarrow \tau(i, next) + \frac{1}{1 + \cos t}$
  end if
 end if
 $\tau(i, next) \leftarrow \min\{\tau(i, next), 1)\}$/* pheromone intensity is at most one */
End
```

Each node *i* remains to serve as a forwarding node for active data sources as long as *Join*(*i*) is not empty.

3.3.3. Forwarding set evolution: Once the initial forwarding set is formed, each active data source attempts to learn a better connection to the GW, in order to minimize the overall cost of the forwarding set and the normalized routing load, by deploying a Forward Ant every *ANT_INTERVAL* time period. A Forward Ant packet deployed by a member *i* for finding a better path join the GW, contains the following fields:

f**:** forwarding flag indicating whether this ant is a Forward Ant or a Backward Ant (since they share the same structure). Since *i* is deploying a Forward Ant, this flag is set to *TRUE*.

exLimit**:** the number of times the ant is allowed to randomly pick a next hop that is not the current best one in order to prevent it from aimlessly traversing the network. This field is initially set to *EXPLORE_LIMIT* and is reduced by one every time the ant decides to pick a random node.

d**:** deterministic flag indicating whether the ant should always follow the current best path in order to obtain the actual current cost for the best cost table. The reason for using deterministic ants is that costs in the best cost table may no longer reflect the actual costs and *path_mpv* may no longer reflect the chosen route's actual mobility due to node mobility, dynamics of nodes' costs, or dynamics of the forwarding set. If this flag is set, the *exLimit* field is always ignored. Every other ant deployed by each member is deterministic.

cost**:** the total cost of the nodes this ant has visited, is initially set to zero.

costLimit**:** the cost limit of the path that the ant is allowed to traverse after leaving its originator. This field is used in conjunction with the cost field to prevent the ant from traversing forward after the accumulated cost exceeds the limit. By this way, the ant can stop proceeding once it is certain that it will not find anything better than what its originator currently has. This cost limit is ignored if the ant is deterministic since its goal is not to find a better cost, but to find the actual current best cost.

visitedNodes**:** the set of nodes visited by the ant, initially set to $\{i\}$.

A node deploying a Forward Ant invokes the procedure *Release Forward Ant* to find the next hop that the ant will travel to. If the ant is deterministic or is not allowed to explore any more, the process of choosing a next hop is similar to formula (3) and (4) except that it excludes all the nodes in the *visitedNodes* field. Otherwise, a probability function is built for all the neighbors by giving higher probabilities to neighbors that have higher pheromone intensities and potentially give lower costs to connect to an existing forwarding node, while giving zero probability to all the nodes that have been visited before. Once a next hop is chosen, its ID is appended to the end of *visitedNodes* and the ant is broadcast.

When a node *j* receives a Forward Ant, it checks if its ID matches the ID at the end of the ant's *visitedNodes* field. If not, the ant is discarded. Otherwise, *j* knows that this ant is intended to itself and accepts it. First, *j* checks if it is currently a forwarding node. If so, *j* realizes that the member who deployed the ant is eligible to set up the link to GW via *j* itself. This ant is then turned into a Backward Ant by resetting its *f* flag. Its cost is reset to zero in order to start computing the total cost on the way back. The last entry of its *visitedNodes* is removed in order to send this ant back to the previous hop. If the condition is not satisfied to convert the ant to a Backward Ant, *j* increases the ant's cost field by its own cost *cost*(*j*). It then invokes the procedure *Release Forward Ant* to release the ant to a next hop if the updated cost does not exceed the limit or the ant is deterministic.

When a node *k* hears a Backward Ant from *j*, it invokes the procedure *UpdatePheromoneAndCost*, described in Algorithm 1, which updates the entries in *k*'s pheromone and best cost tables in accordance with *j*. If the ant is deterministic, the cost that it carries back is the actual cost of the path its originator is currently using to join the tree. Therefore, the best cost is updated to this value. If the ant is not deterministic, however, the best cost is updated only when it is higher than the returned cost, which means that the ant has found a better path to join the tree from this node. The pheromone intensity on this link is also updated to the maximum in order to encourage subsequent Forward Ants to use the same link, as well as to redirect join request to this link instead. If the ant comes back with a higher cost, a pheromone amount of 1/(1+cost) is added instead. In case of deterministic ant, the added amount is reduced by half since this link already has the highest pheromone intensity as it has just been chosen by a deterministic Forward Ant.

Note that we have mentioned this procedure before when we explained how a node uses it while processing a gateway advertisement. This is because a gateway advertisement more or less serves as a deterministic Backward Ant returning from the gateway.

After updating the pheromone and the best cost tables, *k* checks if the Backward Ant was intended to itself by examining the last entry in the *visitedNodes* field. If its ID matches, it adds its cost into the cost field, removes the last entry from *visitedNodes*, and rebroadcasts as long as there is at least one entry left in *visitedNodes*.

Similar to pheromone evaporation of biological ants, each node *i* updates all the entries $\tau(i, j)$ in its pheromone table by reducing their values by *DECAYING_FACTOR* at every *DECAY_INTERVAL* time period:

$$\tau(i, j) \leftarrow \tau(i, j) \times DECAYING\_FACTOR \qquad (5)$$

where $0 < DECAYING\_FACTOR < 1$.

By means of probabilistic selection of next hops, the majority of the Forward Ants will choose paths with high pheromone intensity, while some of them may explore totally different new paths. If a Backward Ant comes back with a better cost on a new branch, the pheromone amount on that branch will be increased significantly. As a result, a change in multicast connectivity (i.e., forwarding set) is triggered due to the periodic broadcast of Join Request packets with *TTL*=1, as illustrated in Figure 1(*d*).

3.3.4. Handling mobility: In ADDSI, mobility and other network dynamics are handled inherently rather than as exceptions. With the pheromone laying/following behavior of Backward Ants and Forward Ants, each path comprising the forwarding set keeps being reinforced as long as no link on the path is broken. However, network dynamics can cause optimal connectivity to change from time to time even though the current connectivity may still be valid. With the randomness of Forward Ants to explore new paths, the forwarding set should be able to evolve into a configuration that is more efficient for the new topology. When a link is currently used by a forwarding node to send Join Requests breaks, the pheromone table entries corresponding to that link are also removed. However, in case that Forward Ants fail to find a new path, Gateway Advertisement flooded periodically will eventually restore the connectivity. Connectivity of the forwarding set may still be fragile if the network is sparse and members are far apart from each other, especially with the presence of mobility.

3.4. Maintaining Connectivity

To maintain connectivity, the GW floods gateway advertisement periodically. This new algorithm is inspired by a hypothesis that the gateway advertisements should only be targeted to those nodes, which are looking for the gateway (such as active data sources); and other nodes should not be hampered with the periodic gateway advertisements. During the periodic gateway advertisement, initially the gateway sends the GWADV message with *TTL*=1 [5].

Then, on the reception of the GWADV message, each node verifies whether it is a *forwarding node*. If the node finds that it is already relaying data of an active source towards the gateway, it learns that it is a *forwarding node*. In case if it is a *forwarding node*, it forwards the GWADV message further with *TTL=1*, otherwise it does nothing. Following this procedure, all the *forwarding nodes* broadcast the GWADV message. This distributed gateway discovery adaptation ensures that all the nodes (*active sources*) which are actively sending data towards the gateway and are actually willing to maintain the route towards the gateway receive the GWADV message traversing hop-by-hop through the *forwarding nodes*. Other nodes in the network, which are not interested in knowing the route towards the gateway, are not hindered with GWADV messages; therefore, the control overhead is significantly decreased.

4. Implementation and Performance Evaluation

To perform the comparison, it was necessary to extend the simulator with the global connectivity support and the evaluated gateway discovery procedures. The global connectivity can be used with different routing protocols. In our implementation, ad hoc on demand distance vector routing (AODV) was chosen because it is the most commonly used algorithm in the related literature. In this section, we evaluate our proposed ADDSI scheme. For this performance evaluation, we conduct extensive simulation under a variety of multi-hop hybrid networking scenarios.

4.1. Simulation Methodology

It is difficult to capture the details of gateway discovery schemes in an analytical model, since the real-time behavior of the multi-hop hybrid network cannot be accurately quantified. For that reasons, we evaluate and analyze the performance of ADDSI in the 802.11-based hybrid network, snapshot ns-2.34 by using the ns-2 simulator [10, 11]. The simulated scenario performed in a rectangular area of 1200×800 m^2. The radio channel capacity for each mobile node is 2 Mbps, using the IEEE 802.11b DCF MAC layer and a communication range of 250 m. There is one gateway located at the center of the rectangular to provide Internet connection service. For our simulated scenarios, we use a random waypoint model. In the simulated scenarios, some mobile nodes are randomly selected as active sources, which communicate with some fixed nodes located across the Internet. The traffic under consideration is Constant Bit Rate (CBR) traffic. Each of these randomly selected CBR sources starts sending data at a uniformly distributed time within the first 10 seconds of the simulation and lasts until the end of simulations (i.e. 900 secs). Each of the sources generates 512 bytes data packets at a rate of 5 packets per second (20 Kb/s). The swarm intelligence parameter values used in our simulation are HELLO_INTERVAL 1 sec, ANNOUNCE_INTERVAL 10 sec, ANT_INTERVAL 2 sec, EXPLORE_LIMIT 3, DECAY_INTERVAL 1 sec, DECAYING_FACTOR 0.1, and NLFF_THRESHOLD 0.1. Note that NLFF_THRESHOLD is used only when the mobility-adaptive mechanism is enabled. Each scenario is simulated 10 times, and the results are obtained as the mean values over these 10 runs.

4.2. Performance metrics and evaluation model

The gateway discovery procedures are evaluated according to the following parameters:

Packet delivery ratio: It represents the ratio between the delivery data packets and the data packets originated by the MANET active data sources. All the data packets originated in these scenarios have a generic host as destination in the Internet which is accessed through any Gateway.

Average end-to-end latency: It corresponds to the mean time that a data packet needs to reach the final destination from the source that originated it. As our objective is evaluating the MANET performance, we consider that the delay from the Gateway to the host in the Internet is negligible.

Normalized routing overhead (Normalized load): It is defined as the total number of control packets (e.g. GWADV) divided by the total number of received data packets. For this computation, every time a control packet is retransmitted, it is computed as an additional control packet. This overhead is of interest in the MANET performance as it may seriously impact on the battery consumption of the nodes.

The performance evaluation of reactive, proactive, hybrid and adaptive gateway discovery schemes has been extensively simulated and evaluated in the past. In this paper, the analyzed procedures to be compared are:

Maximum source coverage (MSC): The gateway adaptively selects the *TTL* value so that all the active sources receive the periodic advertisements from the gateway [3]. In this approach, gateways use *TTL*=2 for GWADV.

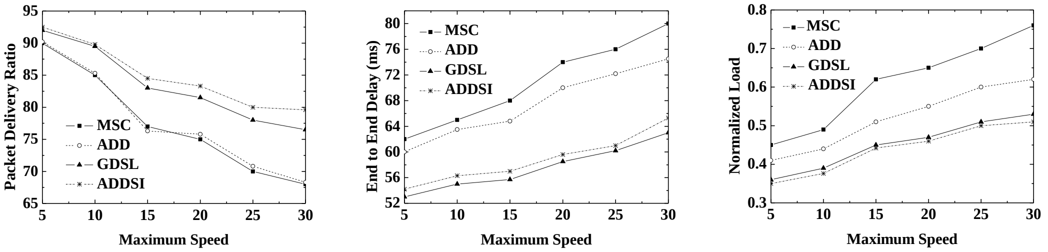
Adaptive distributed gateway discovery (ADD): Nodes retransmit the GWADV message only if they have active connections with the Internet in this scheme [5].

Gateway discovery based on stable links (GDSL): It is the scheme proposed in this paper [8] to prioritize the use of stable links.

4.3. Simulation results

4.3.1. Test of variation of node maximum speed: The first simulation scenario measures the performance of the schemes as the maximum speed of nodes varies from 5 m/s to 30 m/s. 40. Mobile nodes are randomly distributed in the simulated scenario, out of which 4 nodes are randomly selected as active data sources. The performance results for the first simulation scenario are shown in Figure 2.

We show in Figure 2(a) the packet delivery ratio as a function of node mobility speed. As shown in Figure 2(a), the delivery ratios of MSC and ADD decline remarkably, and the packet delivery ratio of GDSL and ADDSI decrease more slowly compared with the above two algorithms. The differences become more apparent at higher speeds. The reason is that: MSC and ADD are significantly affected by the mobility speed. ADDSI has the highest delivery ratio (over 80%) regardless of speed because our new algorithm utilizes a simple mobility prediction model, and it can choose a more stable route for data transmission. Most data are delivered to the GW without being dropped, which can enhance the probability of successful delivery. This latter result validates our simple mobility prediction model and swarm intelligence mechanisms introduced in ADDSI are very useful for overall performance optimization in high mobility multi-hop hybrid networking scenarios.



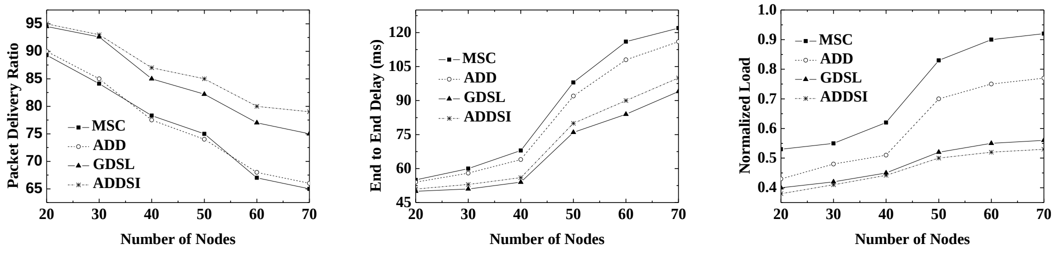
(a) (b) (c)

Figure 2. The Performance of Maximum Speed of Nodes from 5 m/s to 30 m/s

Figure 2(b) illustrates that the average end-to-end latency in these mechanisms rises with the increase of the maximum speed. As shown in this figure, our approach experiences a slightly lower delay compared with MSC and ADD, while a higher delay than GDSL.

In Figure 2(c), we can intuitively see that in terms of normalized load, ADDSI consistently generates less overhead and outperforms the MSC, ADD and GDSL gateway discovery mechanisms. This is mainly attributed to the distributed adaptation mechanism of ADDSI which permits to inform only the active sources about the route towards the gateway periodically. MSC is inherently based on the hybrid gateway advertisement principle in case of high mobility; the nodes quickly learn the updated routes towards the gateway at the expense of high overhead. Whereas, ADDSI broadcasts the gateway advertisement only towards the active sources, thereby the learning of new routes is comparatively slower in case of high link breaks. Moreover, with the help of mobility prediction mechanism, our new scheme is better adopted with node mobility.

4.3.2. Test of variation of node numbers: The second simulation scenario assesses the performance of GW discovery schemes in variation of node numbers. The mobility model is configured at pause-time of 2 seconds and maximum node speed of 15 m/s. The number of nodes in the given area is uniformly increased from 20 to 70 nodes. Moreover, ten percent of the given total simulated nodes are randomly selected as active data sources. The simulation results reported in Figure 3 show considerable overall gain in terms of packet delivery ratio, average end-to-end latency and normalized routing overhead for ADDSI scheme.



(a) (b) (c)

Figure 3. The Performance of the Number of Nodes from 20 to 70

ADDSI has the highest packet delivery ratio, as it can be seen from the Figure 3(a). The mobility mechanism used during distributed gateway discovery helps ADDSI to outperform in terms of successful packet delivery, in case of significant mobility and network density scenarios. Moreover, just after 40 nodes, there is a noticeable decrement of packet delivery ratio in any mechanism. The latter result occurred due to mutual signal influence.

As shown in figure 3(b), ADDSI scheme offers higher delay as compared with GDSL, while lower delay than the other two simulated schemes. Due to mutual signal influence, the end-to-end delay increases dramatically starting from 40 mobile nodes in the scenario.

We can notice in figure 3(c) that, for all schemes, with the increase of node number, the normalized control overhead gradually increases. Our proposed scheme outperforms other schemes when the number of nodes increases in item of normalized load. This can be explained that our two strategies used to reduce control overhead work well. On the one hand, each node receives a copy of control message when gateway broadcasts GWADV message or active node floods GWSOL message. While GWADV messages in ADDSI are only transmitted to active node through a subnet, the overhead produced in this procedure is slightly affected by the number of nodes. On the other hand, the selection of the robust links can decrease the probability of breakage of the path during data transmission, and it also contributes to the performance improvement of our proposed scheme.

5. Conclusion

The convergence of ad hoc and infrastructure network is attractive in real-world scenarios due to its practicality and usefulness. In Hybrid Wireless Networks (HWNs), we aim at the integration and coexistence of two wireless paradigms *i.e.*, infrastructure-based and ad-hoc, in order to offer higher flexibility and improved performance. Mobile ad hoc networks (MANETs) connected to the Internet require a gateway to operate. As MANET nodes move freely, the gateway discovery procedure has to be optimized in order to effectively release wireless resources for the data transmissions.

In this paper, we exposed the motivation and importance of deploying HWNs, while discussing the existing mechanisms of gateway discovery in the hybrid networks ranging from reactive, proactive and hybrid discovery to adaptive discovery schemes. Furthermore, we apply biologically inspired metaphor to the Gateway Discovery problem in Hybrid Wireless Networks, and present the new mechanism-Adaptive Distributed gateway Discovery with Swarm Intelligence (ADDSI). Being intelligent and capable to cope up with the ever changing conditions of the network, our distributed adaptive gateway discovery schemes outperform the existing gateway discovery mechanisms and provide an excellent platform decreasing gateway re-discovery and route maintenance overhead, as validated by the simulations. The results show that ADDSI is able to offer a higher packet delivery ratio comparable to the other schemes.

Acknowledgements

The research work was supported by Qingdao science and technology development project (13-1-4-151-jch).

References

[1] E. Nordström, P. Gunningberg and C. Tschudin, “Robust and flexible Internet connectivity for mobile ad hoc networks”, Ad Hoc Networks, vol. 9, no.1, **(2011),** pp. 1-15.

[2] P. M. Ruiz, F. J. Ros and A. Gomez-Skarmeta, “Internet Connectivity for Mobile Ad Hoc Networks: Solutions and Challenges”, IEEE Communications Magazine, vol. 43, no. 10, **(2005),** pp. 118-125.

[3] P. M. Ruiz and A. Gomez-Skarmeta, “Adaptive Gateway Discovery Mechanisms to Enhance Internet Connectivity for Mobile Ad Hoc Networks”, Ad Hoc and Sensor Wireless Networks, vol. 1, no. 1, (**2005)**, pp. 159-177.

[4] F. Ros and P. Ruiz, “Low overhead and scalable proxied adaptive gateway discovery for mobile ad hoc networks”, Proc. Third IEEE Int. Conf. Mobile Ad-hoc and Sensor Systems, Valencia, Spain, October **(2006)**, pp. 9-12.

[5] U. Javaid, F. Rasheed, D. E. Meddour and T. Ahmed, “Adaptive distributed gateway discovery in hybrid wireless networks”, Proc. IEEE Wireless Communications and Networking Conf., April, Las Vegas, USA, **(2008)**, pp. 2735-2740.

[6] A. J. Yuste, A. Triviño, F. D. Trujillo and E. Casilari, “Improved scheme for adaptive gateway discovery in hybrid MANET”, Pro. IEEE 30th International Conference on Distributed Computing Systems, Genoa, Italy, June **(2010)**, pp. 270-275.

[7] L. Zhuang, Y. Liu and K. Liu, “Hybrid gateway discovery mechanism in mobile ad hoc for internet connectivity”, Proc. Int. Conf. Wireless Communications and Mobile Computing, **(2009)** June, Leipzig, Germany, pp. 143-147.

[8] A. J. Yuste, A. Triviño, E. Casilari and F. D. Trujillo, “Adaptive gateway discovery for mobile ad hoc networks based on the characterization of the link lifetime”, IET Communications, vol. 5, no.15, **(2011)**, pp. 2241-2249.

[9] H. Xia, Z. P. Jia, Z. Y. Zhang and E. Sha, “A link stability prediction-based multicast routing protocol in mobile Ad Hoc networks”, Chinese Journal of Computers, vol. 36, no. 5, **(2013)**, pp. 926-936.

[10] [Online].http://www.isi.edu/nsnam/ns/, accessed, September **(2009)**.

[11] V. Mhatre, “Enhanced wireless mesh networking for NS-2 simulator”, Computer Communication Rev., vol. 37, no. 3, **(2007)**, pp. 69-72.

Author

Gongjing Zhang received his Master degree in Computer Software and Theory from Qingdao University (China).Now work in Qingdao University as an associate professor and a Master instructor. His recent research interests include computer network, information security and artificial intelligence.