Integration of Cluster based routing and Mobile Service Discovery Protocol for MANETs: A Novel Approach

M. Shukla, Monika, Rohit Kishore, Rohit Kumar

Abstract--Service discovery technologies enable services to advertise their existence in a dynamic way, and can be discovered, configured and used by other devices with minimum manual efforts. In order to achieve efficient service discovery in large scale Mobile Ad hoc Networks (MANETs) we have proposed a scheme that integrates service discovery with cluster based routing by piggybacking service information into routing layer messages. Thus, service discovery does not generate additional messages and a node requesting a service, in addition to discovering that service; it is simultaneously informed of the route to the service provider. The proposed Service Discovery Protocol uses a dynamic clustering algorithm to group nodes in a MANET into clusters, and utilizes Distributed Hash Tables (DHTs) called Service Table to efficiently cache service information in a peer-to-peer manner. This paper proposes a scheme that greatly enhances service discovery in terms of efficiency, minimizes the traffic and energy consumption.

Index Terms--Service Discovery, Dynamic Clustering, Service Table, Cluster based Routing

I. INTRODUCTION

A\n
An ad hoc network is a wireless network made up of mobile terminals, connected together using wireless links. An ad hoc network consists of mobile nodes with wireless devices and routing functions, communicating with each other without any fixed infrastructures. In mobile ad hoc networks, the source node can communicate with the destination node through some intermediate nodes which forward data packets. However, since the battery and bandwidth resources of each node are tolerably limited, they must be efficiently utilized to provide the assured communication in mobile ad hoc networks. In addition, along with the increased network size and the higher node mobility, the ad hoc network encounters scalability issues. The proposed work aims to achieve efficient service discovery that considers the characteristics of ad hoc networks.

A critical challenge in designing MANETs is service discovery. In this paper, we define the problem of service discovery as how a user can efficiently search and locate the service information he desires in a MANET. Service information refers to any information that is used to describe, locate and access a given service. An efficient service discovery mechanism should have low network resource consumption, fast response time, and high query success rate.

The service request is broadcasted by the users who want to discover a service. Response messages are sent using routing protocol packets. In the case of a reactive routing, sending answer messages will trigger the routing protocol if the terminal does not already know the route towards the service requester. Thus, it is necessary to minimize the number of messages exchanged and to avoid the activation of the routing protocol. In order to achieve reduced message transfer, the nodes in MANET can be grouped into clusters.

Hence the Cluster based routing protocol is elected to integrate with service discovery. The autonomous clustering divides a network into sub networks called clusters and manages it hierarchically. A cluster head which manages a cluster must surely exist in each cluster. Each cluster consists of a cluster head which manages the cluster, and only the neighboring nodes called cluster member. If a cluster member is adjacent to two or more cluster heads, it changes the role in the cluster and serves as the gateway between two clusters.

However, clustering has some problems as follows. Firstly, there is the possibility that every time cluster heads change, clusters tend to be reconstructed frequently. This is because, when a node moves out of the communication range of the cluster head, the node either becomes a new cluster head or a cluster member of another cluster head. As a result, if nodes move around in the network at much higher speed, the clusters to which they belong would be frequently reconfigured. Secondly, the overhead of each cluster head depends on the density of nodes in the network. Here, the density denotes how many cluster members are included in a cluster. In case that the density of nodes in each cluster becomes much higher, the cluster head may have much overhead to manage all cluster members in the cluster. In addition, in case that the density of nodes becomes much lower, there are many small size clusters in the network. As a result, it is difficult to construct and maintain an effective hierarchical structure. The proposed cluster based routing protocol considers all these factors into account by adapting a dynamic clustering algorithm.

This work aims at several objectives: a) To group the nodes of MANET into dynamic clusters that adapts to the changing infrastructure of MANETs. b) To implement service discovery in the routing layer by piggybacking the
service information into the routing protocol control messages. c) To enable the devices to acquire and update both service and routing information simultaneously.

The rest of the paper is organized as follows: In section 2 we provide the essential background on service discovery protocols. Section 3 describes the motivation of this proposal. In section 4 we present our approach of routing layer based service discovery, and finally section 5 concludes the paper and proposes our future research directions.

II. RELATED RESEARCHES

Many service discovery mechanisms are already widely used in today’s Internet. One example is the search service provided by Google. With the proliferation of the Internet in recent years, the number of web pages has grown tremendously. In order to let users find the information they want, search engines like Google have created large databases to cache the information on WebPages throughout the Internet, and allow users to use key words to search for the information over the Internet. The above mechanisms are mainly designed for use over the Internet. They rely on central servers for information storage and dissemination, and thus are not suitable for the dynamically changing MANET environment. The ubiquitous personal computers and laptops and the availability of Personal Area Network (PAN) technologies motivated the design of several service discovery protocols.

Examples are Universal Plug and Play (UPnP), Jini, Service Location Protocol (SLP), and Bluetooth Service Discovery Protocol (SDP). However, these protocols aim to perform service discovery among hardware devices, and do not fit our purposes here. Peer-to-peer (P2P) networks have been increasingly popular. Besides being a growing way of file sharing over the Internet, P2P networking technologies also provide a promising means for service information sharing and dissemination over MANETs. Early P2P networks like Napster relied on centralized index servers. In Napster, some servers were responsible for storing all the node information in the P2P network.

In [6], Christopher N. and George proposed to integrate service discovery messages into Zone Routing Protocol. Dividing a large MANET into several smaller regions can effectively reduce the message overhead caused by service discovery messages. Hence the cluster based routing protocol is well suited to integrate with Service discovery protocol.

In Ad-Hoc On-Demand Distance Vector routing protocol, like all reactive protocols the topology information is only transmitted by nodes on-demand. In [4], R.Oda, T.Ohta, and Y.Kakuda described about Ad hoc On-demand Distance Vector Routing Protocol and also discussed about the improvement over AODV routing protocol.

The set of applications for MANETs is diverse, ranging from small, static networks that are constrained by power sources, to large-scale, mobile, highly dynamic networks. It is unlikely that a single service discovery protocol will be optimal for all scenarios. However, any protocol must efficiently handle several inherent characteristics of MANETs such as dynamic topology, Variable capacity wireless links, Physical security, Power constrained operation, etc. Such service discovery protocol should reduce message traffic, bandwidth requirement, etc.

The proposed scheme in this paper has three key unique features. First, it can dynamically group nodes in a MANET into clusters for the purpose of caching service information. Dividing a large MANET into several smaller regions can effectively reduce the message overhead caused by service discovery messages. Second, it utilizes Distributed Hash Tables called Service Table to efficiently cache service information in clusters. The use of DHT not only further reduces the message overhead, but also provides a definite guide which tells nodes where to search and cache their service information. Third, it implements service discovery in the routing layer by piggybacking the service information into the routing protocol control messages, thus enabling the devices to acquire both service and routing information simultaneously. The combination of these two mechanisms (service discovery and routing) leads to high scalability and robustness making the proposed protocol suitable for ad hoc networks.

III. MOTIVATION

The development of Mobile Ad Hoc Networks (MANETs) aims to support the proliferation of mobile devices and it leads to the emergence of pervasive computing and gives rise to the challenges of the service discovery techniques, because MANET allows these devices to communicate dynamically without fixed infrastructure and centralized administration. However, since the battery and bandwidth resources of each node in MANET are tolerably limited, they must be efficiently utilized to provide the assured communication and scalability in mobile ad hoc networks. In this paper, we present a dynamic service discovery method that considers the properties of nodes in Ad hoc networks and reduces the messages exchanged in discovery process. Automatic service discovery will play essential role in future network scenarios as they require minimum manual effort. The proposed work aims to achieve efficient service discovery that considers these limitations of ad hoc networks. This proposal aims to reduce the number of control messages in service discovery by encapsulating the service discovery messages into routing reply messages.

IV. PROPOSED SCHEME

We assume that in a given MANET, a subset of all the nodes is capable of providing one or more services; others may search and use any services which are not restricted to the services in the MANET. Services can be files, environmental information or hardware devices such as printers. The goal of a service discovery protocol is to efficiently help users find the services they want in a given
MANET. The proposal comprises three major parts, which are dynamic cluster formation, routing management and service discovery and caching. The following subsections will explain how they work.

A. Cluster Formation

(i) Outline of Clusters

A cluster is a subset of nodes and consists of one cluster head and two or more cluster members. Node ID of the cluster head becomes cluster ID of the cluster. A cluster head and other cluster members should connect with each other with the same cluster ID. The cluster is managed by the number of members. The number of nodes in each cluster is always adjusted between the lower bound L and the upper bound U which are set before hand, respectively. Moreover, a node which is adjacent to other node in a different cluster ID is called a gateway node. A gateway node performs the communication and the information gathering with the neighbor cluster.

(ii) Cluster head

A node which manages the cluster is called cluster head. The cluster head constructs the cluster head-based tree in the cluster to collect member’s information efficiently. The cluster head manages the cluster member list and the neighboring cluster list from information on the cluster members and the neighboring clusters.

(iii) Construction of Clusters

We explain the mechanism that the cluster head manages the clusters. Every node has a unique identifier (UID), which can be generated through applying a specific hash function to a given string. In the process of forming dynamic clusters, every node can be in one of the following states: Unknown, Member, or Head. The radius of the cluster is \( CSIZE \) hops from the cluster head. Each node also has a countdown timer whose maximum value is \( \text{TIMERMAX} \) seconds. Two kinds of control packets that are called MEP (Member Packet) and MAP (Member Ack Packet) are used to construct the clusters. The cluster head constructs the cluster head-base tree using these two control packets and manages the cluster. The cluster head periodically broadcasts MEP with its cluster ID within the cluster to construct the cluster. The node that received MEP sets the node that transmitted MEP as the upstream node or the parent node, and forwards MEP to any nodes except for the parent node. When the node rebroadcasts MEP, the node sets the response timer. Here, the response timer is set to \( t1 - n \times t2(\geq 0, t1 > t2) \) in each node. \( t1 \) and \( t2 \) are some positive numbers, and \( n \) is a hop number of the route from the cluster head to the each node in the cluster head-based tree. When cluster ID included in MEP the node received is different from the node’s cluster ID, the node that received MEP becomes a gateway. And, the gateway acquires information of the neighboring cluster from MEP. When the response timer becomes timeout, the node sends MAP back to its parent node. In this time, the node adds its own node ID to MAP. If the node that forwards MAP is a gateway, its own node ID and the neighboring cluster information are added to MAP. Thanks to the response time, it is possible for each node to aggregate MAP transmitted from the downstream nodes or the child nodes, and the increase of the control packet can be suppressed. This mechanism is repeated until the cluster head receives the MAP. The cluster head constructs the cluster head base-tree by the above operation and collects cluster member’s information and the neighboring cluster information.

Algorithm 1: Upon MEP Message Reception

begin
  for each entry e in the MEP Message do
    add neighbor entry(e, e.state);
    if e.state = Head and e.hop < CSIZE then
      me.state = Unknown
      me.state = Member;
    else if me.state = Head then
      e.id < me.id then
        start timer();
    end
    else if e.state = Unknown then
      if me.state = Unknown and e.id < me.id then
        restart timer();
    end
  end
end

Algorithm 2: When Response Timer Expires

The cluster head-based tree is used for management of the cluster and intra-cluster routing. Algorithm 1 shows the procedure that a node uses to handle incoming MEP. Upon hearing a MEP, a node will store the UID and state of each node the MEP carries in its Neighbor Table. It then checks the state of each entry, and performs different procedures depending on the state of itself. When an Unknown node learns that there is a Head node within CSIZE hops, it changes its state to Member to join the cluster represented by this cluster head. If a Head node finds that there is another Head node with smaller UID within CSIZE hops, it starts its countdown timer. Finally, if an Unknown node u1 knows the existence of another Unknown node u2 within CSIZE hops and u1 has a larger UID than u2, u1 will restart its countdown timer from TIMER MAX. Algorithm 2 shows the procedure a node executes when its countdown timer expires. Similarly, nodes in different states have different things to do. When the timer of an Unknown node expires, it changes its state to Head; this is because it has local minimal UID. A Member node checks whether it is still within CSIZE hops of any cluster heads; if not, it changes its state back to Unknown. A Head node checks whether the other Head node that caused its timer to start still exists within CSIZE hops; if so, it changes its state to Member and becomes a cluster member. This operation can avoid Head nodes to change their states when other Head nodes move in their coverage temporarily, and thus stabilizes the cluster structure.
begin
if me.state = Unknown then
    me.state = Head;
else if me.state = Member then
    if no Head exists within CSIZE hops then
        me.state = Unknown;
    else if the Head with smaller UID still exists
    within CSIZE hops then
        me.state = Member;
end
end

(iv) Cluster Reconfiguration

The cluster head can acquire the number of cluster members by referring to the cluster member list. In case that the number of cluster members becomes beyond or below the specified cluster size, the cluster head is merged or divided. The mechanism of the merger and the division of the cluster is shown as follows. When the number of nodes in the cluster is less than the lower bound \( L \), the cluster head checks the sizes of all the neighboring clusters, and then merges the cluster with one of the neighboring clusters. When it is larger than the upper bound \( U \), the cluster head divides the cluster to two clusters. Division and merger mechanisms for the autonomous clustering scheme are shown in [10] in detail. In either case, though information on neighboring clusters of merged or divided clusters is updated, influence of merger and division of clusters is restricted. Maintenance of clusters is thus locally performed.

B. Routing Management

MANET Routing protocol has to consider the characteristics of MANET like: Dynamic Topologies, Bandwidth-constrained, variable capacity links Energy-constrained, Limited Physical security Scalability. Based on route discovery time, MANET routing protocols fall into two general categories: (i) Proactive routing protocols and (ii) Reactive routing protocols. Proactive MANET protocols maintain routing information independently of need for communication and they update messages send throughout the network periodically or when network topology changes.

Reactive MANET protocols only find a route to the destination node when there is a need to send data. The major goal is to achieve a distributed, efficient, scalable protocol. The major design decision is to use clustering approach to In this paper we have proposed to integrate the reactive routing protocol with discovery protocol. In reactive: Ad-Hoc On-demand Distance Vector Routing (AODV) the sender tries to find destination by broadcasting a Route Request Packet (RREQ). So as shown in Fig 1 the Source S “Flooods” all cluster heads with Route Request Packets (RREQ) to discover destination D.

minimize on-demand route discovery traffic.

![Fig. 1: Route Discovery in AODV](To be merged with Service Discovery)

Each node maintains route cache and uses destination sequence number for each route entry. Reply for this request is sent back to the source in the form of Route Reply (RREP). This is shown in Fig 2

Two types of packets are used for route maintenance: Route Error Packet and Acknowledgement. If transmission error is detected at data link layer, Route Error Packet is generated and sends to the original sender of the packet. The node removes the hop is error from its route cache when a Route Error packet is received. ACKs are used to verify the correction of the route links.

C. Service Discovery and Caching

Every Cluster Head node treats all the Member nodes within CSIZE hops as its cluster member, and uses their UIDs to form a key space. We assume that there exist a function which takes a service information or description string as input, and outputs an integer key that will be used in the service discovery process. In the MSDP, the SHA1 hash function is used to generate a 128-bit integer from a given service information string. We also use Consistent Hashing to map the integer generated by the SHA1 function to one of the keys in the UID key space. In consistent hashing, the keys in the key space are sorted in increasing order and form a circular linked-list, i.e. the last element of the sorted list links to the first element. The size of the key space is \( n \), i.e. all the keys are in the range \( [0, n-1] \). The hash function takes an integer \( i \) as input, and finds the “nearest” key \( k \) to this integer in the key space using modulo operations. In consistent hashing, \( k \) is the smallest key that is no less than \( (i \mod n) \); if \( (i \mod n) \) is greater than the largest key in the key space, then \( k \) is the smallest key in the key space. When a user wants to
find a specific service, he packs the service description along with the 128-bit ID used to represent the description in a Service Request packet, and sends the packet to the head of the cluster he belongs to.

When a Head receives the Service Request, it uses consistent hashing to map the 128-bit ID in the packet to one of its Member nodes, and relays the Service Request to the mapped node. Finally, when the Member node receives the Service Request, it checks its own service database; if the requested service is found, it sends back a Service Reply containing the service information. Otherwise, it sends back a Service Reply with no service information, indicating that no service information is found and it is a cache miss. If the user receives a Service Reply which indicates that no service information is found in the local cluster, it sends out another Service Request which, through the help of the local cluster head, is then broadcasted to all Head nodes. Other Head nodes that receive this Service Request then query all the Member nodes they have for the requested service information. Eventually, the user will receive a Service Reply if the requested service does exist in the MANET. After assuring the availability of the requested service, the user sends out a Service Publish packet containing the service information to his cluster head. Similar to Service Requests, the Head uses consistent hashing to map the service information to a Member node and relays the packet to that Member. The Member that receives the Service Publish then caches the service information in its database. Later when other users in the cluster wish to access the same service, their requests can be fulfilled locally and wide area broadcasting can be avoided. Caching service information can effectively reduce the message overhead occurs during service discovery process. In the MSDP, the service information items are cached in soft states. When a service information has not been accessed for EXPIRE PERIOD seconds, it will be removed from the cache database. A popular service will be updated frequently, and can thus stay in the cache database for a long time.

There are two advantages of using soft state to store service information. First, service providers do not have to periodically update their service information. Service information updates are automatically done by users that access this service. Thus less network resources are consumed. Second, frequently used services will stay in the cache database for a long period of time and services that are not in popular demands will be removed quickly. This leads to more efficient use of node storage space.

In the proposed scheme Service discovery process has to be merged with route discovery process. Service Request has to be encapsulated into Route Request. Similarly, Service Reply packets have to be piggybacked with Route Reply. As the service discovery is integrated with routing process the number of control messages exchanged for these processes are reduced. Hence the traffic is reduced. Consequently the bandwidth requirement is also reduced. Overall performance of MANET gets improved.

D. Benefits

In the proposed scheme, each cluster head periodically distributes the information of the inter cluster routing with MEP to all gateways. Each gateway which received the MEP can acquire the next hop cluster information to forward a data packet to the destination cluster and the gateway information containing the cluster IDs to which each gateway in the cluster is neighbor in order to forward it to the next hop cluster. The proposed efficient on demand hierarchical routing scheme can reduce the number of data packets and control packets which each cluster head handles. As a result, the communication efficiency can be improved. The proposed scheme can reduce the overhead of the cluster head and the number of hops of data packets because the service discovery and routing processes are merged together. Therefore, we can say that the proposed scheme is more effective than the previous scheme. The routing layer based service discovery scheme has more number of benefits including less energy consumption, less network resources are consumed, service providers do not have to periodically update their service information, frequently used services will stay in the cache database for a long period of time and services that are not in popular demands will be removed quickly.

V. CONCLUSION

Our work is a cross-layer design that combines network routing protocol with service discovery protocol. The proposed service discovery combines dynamic cluster formation and routing process, distributed hash tables, and can efficiently organize the MANET to provide service information caching. Service discovery messages can be resolved locally in clusters, and thus reduces the message overhead caused by the service discovery process. The proposed scheme has steady performance in terms of service information discovery and caching, and is able to minimize the network resources consumption.

In future, we plan to obtain simulation results of this proposal. Also we plan to extend our proposed scheme to work under other environment and on different hardware technologies.

VI. REFERENCES


