Associativity based cluster formation and cluster management in ad hoc networks

Arvind Ramalingam, **Sundarpremkumar Subramani**, and **Karthik Perumalsamy**

School of Computer Science and Engineering,
Anna University,
Chennai,
India.
{arvind_r,spk,pkarthik}@cs.annauniv.edu

Abstract. In this paper we propose "Associativity based cluster formation and management Protocol", a new scheme for cluster formation and cluster management in ad hoc networks. The cluster formation is based on the associativity ticks of the nodes. The metric used for electing the clusterhead is maximal connectivity and spatial and temporal stability. The connectivity of a node in a given cluster is ascertained periodically by means of beacons.

1 Introduction

An ad hoc network is a dynamic multi-hop wireless network established by a group of mobile nodes on a shared wireless channel without any infrastructure. Routing in ad hoc networks has been a topic of research in the past few years. The literature that deals with ad hoc routing identifies two types of networks viz. hierarchical routed networks [1, 2] and flat routed networks[10]. While the former creates a hierarchy among the nodes the latter treats all nodes equally. In this paper we introduce a protocol for the formation and management of clustered ad hoc networks.

In most of the methods proposed in the literature [1, 2] for routing in hierarchical (clustered) networks, much thought has not been given for the selection of the cluster head. For instance [1] selects cluster head based on the node identifier (Least ID clustering algorithm). Though this could reduce initial computational overhead in determining the cluster head, later a lot of network bandwidth is consumed for intra-cluster routing. The cluster head elected by this method need not be centrally located in the cluster. The result is that those nodes that are farther away from the cluster head have to route their packets across more hops than those in the proximity of the cluster head. Given that the nodes are uniformly distributed across the cluster the average number of hops for the cluster head to reach the nodes increases.

The concept of associativity (relative stability of nodes) was used to get long lived routes in routing in [11]. Associativity of the nodes had not been considered while [1] determined the cluster head. As a result the cluster head selected need not be the best in terms of spatial stability. This is not trivial since the movement of cluster head results in dismantling of the entire cluster, which in turn results in additional overhead of cluster formation. We propose using associativity as a metric in the selection of the cluster head so that the temporal stability of the cluster and hence the bandwidth utilisation, which is a crucial factor in ad hoc networks, are increased .

In this paper we address the above-mentioned issues and present the "Associativity based cluster formation and management" protocol.

The remaining part of the paper is organised as follows. Section 2 deals with the cluster formation, section 3 deals with cluster management and section 4 concludes the paper.

2 Cluster formation

The associativity based cluster formation scheme forms clusters based on the spatio-temporal stability of the mobile stations forming the ad hoc network. The main goal of this protocol is to form clusters in such a way that the cluster is stable over a long period of time.

The node elected as the clusterhead(CH) is such that it has maximum associativity as well as satisfies a minimum connectivity requirement.

The CH periodically sends cluster head beacon (CHB) which has the format as shown below and serves to advertise the existence of the cluster.

PACK_TYPE	CH_ID	HOP_COUNT

The CH sets the HOP_COUNT to MAX_HOP_COUNT. Each time a node forwards a CHB it decrements the HOP_COUNT field. A node forwards CHB to all its neighbours only if HOP_COUNT> 0 .

Whenever a node is switched on, it waits for a specified period of time, $T_{CHB\_MAX\_HOP}$, which is larger than the average time for CHB to traverse an entire average sized cluster . If within this period it hears a cluster head's beacon (as explained in section 2.3) it joins the cluster by sending a "register me" packet. Otherwise, it enters the cluster formation phase as described below. Any node that is switched on, starts sending Alive Beacon(AB) irrespective of the cluster proximity(section 3).

The cluster formation involves three phases viz. Neighbour Acquisition, Cluster Control Claim and Cluster Head Assertion. All nodes need not be in the same phase at the same time.

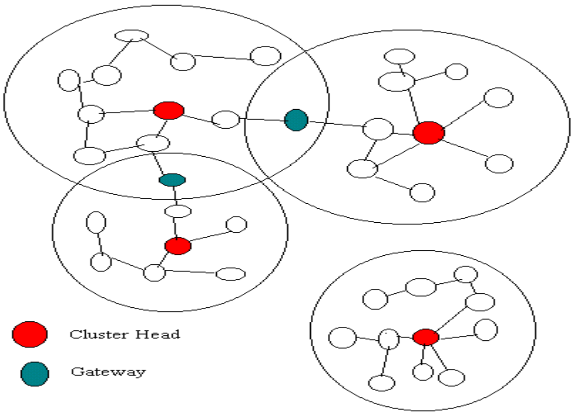

Fig. 1. A clustered ad hoc Network

2.1 Neighbour Acquisition

The node sends a Neighbour Acquisition (NA) packet which has the following format,

PACK_TYPE	SENDER_NODE_ID

The nodes that are able to hear this packet reply with a Neighbour Acknowledge (NACK) packet. The NA packets are not forwarded.The NACK packet format is given below.

PACK_TYPE	SENDER_NODE_ID	NODE_ID

The sender waits for a specified period of time for NACKs to arrive and from the received NACKs determines the number of nodes in its neighbourhood.

2.2 Cluster Control Claim

After the first phase each node waits for a random amount of time and tries to send a Cluster Control Claim(CCC) packet. The node that has the least waiting time succeeds in doing so and is identified as the initiator(INI).The CCC packet format is given below.

ROUTE	INI_ID	HOP_COUNT

When a node receives this packet the node appends its NODE_ID to the ROUTE field (to facilitate the farther nodes to reply back to INI) and decrements the HOP_COUNT field. Each CCC packet is forwarded if HOP_COUNT > 0. It also replies the INI with the tuple

ROUTE	NODE_ID	NN	SUM_AT

in a CCC Reply packet, where ROUTE refers to the entire source route to the INI (recorded in the incoming CCC packet), NN refers to the number of neighbours of the node with identifier NODE_ID and

$$SUM\_AT = \sum_{i=1}^{NN} AT_i$$

is the sum of associativity ticks (Section 3) of the sending node.

After transmitting the CCC packet INI enters the CCC_WAIT_STATE by starting a timer. All CCC Reply packets that arrive before the timer expires are eligible for CCC. Others are discarded. The INI is also a candidate for being the CH.

Only those nodes whose NN value is greater than a threshold say NN_MIN are eligible for becoming the CH. Among these nodes the one with a maximum AT value is selected to be the CH . If there are any ties they are first resolved based on NN. Further ties, if any, are broken by ID.

The INI informs the selected node that the latter has become the CH by sending it a "Take Charge"(TC) packet. This packet is routed to the CH along the route INI knows from the CCC Reply packets. If the INI itself is the CH it enters the next phase.

2.3 Cluster Head Assertion

The CH informs all the nodes in the cluster by sending beacons (CHB). The CHBs contain the CH_ID, a maximum permissible hopcount and a Timestamp. This packet is forwarded to all nodes in the cluster. By knowing the CH_ID field all other nodes in the cluster know the CH.

This marks the end of the cluster formation phase.The cluster formation is explained in algorithm 1.

3 Cluster Management

Each node periodically sends a beacon namely Alive Beacon(AB). AB consists of NODE_ID field and timestamp. Nodes that hear this AB checks their neighbour data table (NDT).If NDT does not contain an entry with AB's NODE_ID it creates a new entry for the sender of AB, sets its AT field to zero and initiates a timer. Whenever it receives AB with the same NODE_ID before the timer expires, it sets AT corresponding to that node to 1 and restarts the timer. If the timer expires before an AB from a node is received the node is deemed to have moved and the CH is informed about this. The corresponding entry in the NDT is removed. ABs are not forwarded.

Each node also hears the periodic CHBs. There are four possible cases:

- If a node hears only its CHB then both the CH and the node are within the cluster.
- If it hears another cluster's CHB along with its CHB, then it enters the CH_ID of the former's CHB in a gateway table (GT). If within a specified period of time (GT_THRESHOLD) it hears both the CHBs once again, it affirms itself as the gateway between these two clusters by sending a "gateway assert"(GA) packet to both the CHs. When a CH receives a GA packet it checks its GT to see if there is any other gateway already available.
  1. If there is no other gateway to that cluster it affirms GA by entering the gateway's NODE_ID in its GT and sends a "gateway registered" packet.
  2. If there is already a gateway entry to that cluster the CH compares the hopcount of the current gateway with that of the GA sender. If the latter has a lesser hopcount, deregister the former and send a "gateway registered" packet to the latter. Otherwise disregard the GA packet. In this case the latter refrains from sending further GA packets for a specified period of time and repeats the process again.
- If it hears another cluster's CHB only it registers with that cluster by sending a "register me" packet. In response to this, CH updates its table of nodes that are present in it's cluster and sends the node a "registered" packet.

Algorithm 1 Cluster formation

```
Require: No CHB is heard
  {Neighbour Acquisition phase begins}
  Send NA packet and start a timer t
  NN ⇐ 0
  while t > 0 do
    if NACK arrives then
      NN⇐NN +1
    end if
  end while
  {Neighbour Acquisition phase ends}
  {Cluster Claim Control phase begins}
  Start a timer rt initialised with a random value
  while rt > 0 do
    if CCC arrives then
      SUM_AT = ∑_i AT_i
      Send SUM_AT, NN in a CCC Reply packet
      Wait for CHB {Cluster formation is over}
    end if
  end while
  if rt = 0 then
    Send CCC packet and start t_CCC timer
    while t_CCC > 0 do
      if CCC Reply arrives then
        Buffer CCC Reply packet
        that have NN > NN_MIN
      end if
    end while
    Find the packet(s) having maximum SUM_AT
    if tie then
      Resolve tie by packet(s) having maximum NN
      if tie then
        Resolve tie by using NODE_ID
      end if
    end if
    if INI is not selected as CH then
      Send TC to CH
    end if
  end if
  {Cluster Claim Control phase ends}
  {Cluster Head Assertion phase begins}
  CH starts sending CHB
  {Cluster Head Assertion phase ends}
```

– If it is unable to hear any CHB two cases can arise.
  1. The **node could have moved** from the cluster. In this case the upstream node(s) (the node which has less hop count distance from the CH) would have detected the mobility as explained in the beginning of this section and they will take necessary actions.
  2. **CH** is considered to have **relatively moved**.In this case the nodes back off for a random time interval and start the cluster formation phase once again.

4 Conclusion

In this paper we have presented "Associativity based cluster formation and management" protocol for ad hoc networks. We first address the need for such a protocol given the plethora of protocols already available. This protocol takes into account the spatio-temporal stability and the optimal location of the cluster head in a cluster. We have elaborated on the mechanism by which the nodes in a cluster interact with each other and select a cluster head.

We have also presented the different issues that may arise in management of a cluster and summed up the necessary action to be taken by the cluster head in this regard. We are presently working on a complete simulation study of this protocol. We believe that this protocol will improve the bandwidth usage of the network and the cluster stability.

5 Acknowledgement

We would like to thank Dr. Arul Siromoney, Assistant Professor, SCSE,Anna University for his valuable guidance and comments.

References

1. M. Gerla and J. Tsai: "Multicluster mobile multimedia radio network," Baltzer Journals, July 12, 1995.
2. P.Krishna, M.Chatterjee, N.H.Vaidya and D.K.Pradhan: "A cluster based approach for routing in Ad-Hoc networks," in proceedings of second USENIX Symposium on Mobile and Location Independent Computing , April 1995.
3. C.K.Toh: "Long-lived Ad-Hoc Routing based on the concept of Associativity," March 1999 IETF Draft. `www.ietf.org/internet-drafts/draft-ietf-manet-longlived-adhoc-routing-00.txt`
4. R. Sivakumar, P. Sinha and V. Bharghavan:
" CEDAR : A Core-Extraction Distributed Ad hoc Routing algorithm," IEEE Journal on selected areas in communication, Vol 17, No. 8, August 1999.
5. R. Sivakumar, B. Das, and V. Bharghavan: "Spine Routing in Ad Hoc Networks," ACM/Baltzer Cluster Computing Journal (special issue on Mobile Computing), vol. 1, no. 2, November 1998.
6. David B. Johnson and Davis A. Maltz: "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," October 1999 IETF Draft.
`www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt`
7. Charles E. Perkins and P. Bhagwat: "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," Comp. Comm. Rev., October 1994.
8. Charles E. Perkins, Elizabeth M. Royer and Samir R. Das: "Ad Hoc On-demand Distance Vector Routing," October 1999 IETF Draft.
`www.ietf.org/internet-drafts/draft-ietf-manet-aodv-04.txt`
9. R. Dube et al.: "Signal Stability based adaptive routing for Ad Hoc Mobile networks," IEEE Pers. Comm., February 1997.
10. Zygmunt J. Haas and Marc R. Pearlman: "A New Routing Protocol for the Reconfigurable Wireless Network," in Proceedings of the 6^th IEEE International Conference on Universal Personal Communications, October 1997.
11. Chai-Keong Toh, "A novel distributed routing protocol to support Ad hoc mobile computing" Proc. 1996 IEEE 15th Annual Int'l. Phoenix Conf. Comp. and Commun., March 1996.