Journal of Computer Science 2 (7): 583-588, 2006
ISSN 1549-3636
© 2006 Science Publications

Distributed Clusterhead Architecture for Mobile Ad Hoc Networks

[1]Emad Qaddoura, [2]Wesam AlMobaideen and [3]Ahmad Omari
[1]Faculty of Computer Science and Information Technology, Applied Science University, Jordan
[2]King Abdullah II School for Information Technology, The University of Jordan, Jordan
[3]Faculty of Computer Science and Information Technology, Applied Science University, Jordan

Abstract: Mobile Ad Hoc Network (MANET) consists of dynamic topology as nodes in the network are mobile, and connected via wireless links. The nodes are grouped in clusters for the purpose of routing. Nodes within the cluster communicate directly. However, nodes communicate outside the cluster through a centralized node that is called a clusterhead. An elected clusterhead is assigned for communication with all other clusters. The centralized clusterhead can become a bottleneck and possibly cause a lower throughput for the clustering system. We propose a mechanism in which communication outside the cluster is distributed through separate clusterheads. We prove that the overall averaged throughput for the clustering system increases considerably, about doubles for our distributed number of clusterheads. We also show that the increase in overall averaged throughput is reduced by the increase in the overall averaged error rate ratio resulting from the added clusterheads. Additionally, the routing reliability to outside the cluster is increased since routes to different clusters use distinct clusterheads.

Key words: Routing Protocols, Mobile Ad Hoc Networks, Distributed Routing Protocols, Clusterhead

INTRODUCTION

A wireless ad hoc network consists of mobile nodes that move freely and communicate with each other using wireless links. A MANET system does not require a predetermined infrastructure and nodes act as hosts and as routers for other nodes. Nodes are grouped into distinct or overlapping clusters. Clustering provides a hierarchical MANET system which assists in making the routing scalable. Some of the nodes are elected to be part of the backbone for the MANET system [2][4][16][19-21]. These nodes are called clusterheads or gateways. Clusterheads are elected according to several techniques. The clusterhead allows for minimizing routing details overhead from other nodes within the cluster. Overlapping clusters might have nodes that are common among them which are called gateways [14]. MANET requires efficient routing algorithm in order to reduce the amount of signaling introduced due to maintaining valid routes[5][10], and therefore enhance the overall performance of the MANET system. As the clusterhead is the central node of routing for packets destined outside the cluster in the distinct clustering configuration, the clusterhead computing machine pays a penalty of unfair resource utilization such as battery [1], CPU, and memory. Several studies have proposed a clusterhead election in order to distribute the load among multiple hosts in the cluster [1][4][8][17]. Our approach can use any of them to elect a clusterhead. This paper is organized as follows: Section 0 discusses the related work, section 0 discusses the approach taken for this research, section 0 outlines system model used, section 0 discusses the numerical results obtained, and finally section 0 discusses the conclusions and proposes future work.

RELATED WORK

Several mechanisms of clusterhead election exist with an objective to provide stable and efficient routing in the MANET system [5][13][20-21]. Some mechanisms favor not changing the clusterhead to reduce the signaling overhead involved in the process [7], which also makes the elected node usage of its own resources higher. Other mechanism assigns the clusterhead based on the node id as in the Linked Cluster Algorithm, LCA, which selects as the clusterhead the node with the highest ID [4]. However, this selection process burdens the node due to its ID. Other mechanisms favor allowing some type of fair share of clusterhead responsibility by changing the clusterhead based on an assigned ID to the clusterhead [2], where all nodes have a chance to be a clusterhead for a clusterhead duration

Corresponding Author: Emad Qaddoura Faculty of Computer Science and Information Technology, Applied Science University, Amman-Jordan.

budget. This mechanism keeps the clusterhead load within one node for the clusterhead duration budget, while it provides a balance of responsibilities for nodes within the cluster. A node with a high mobility rate may not get the chance to become a clusterhead if its mobility rate is higher than the duration of clusterhead rotation. Other clusterhead election mechanisms consider relative node mobility to ensure routing path availability [4][14][18], however, causing an added signaling overload and causing the elected clusterhead to pay the higher resource utilization penalty.

We can conclude from the existing research that several tradeoffs exist for the elected clusterhead and the other cluster nodes. Firstly, the clusterhead has to bear higher resource utilization such as power, which may deplete its battery sooner than other nodes in the cluster. In addition, possibly causing more delay for its own application routing due to the competition with the routing for other nodes. Secondly, despite fair share responsibility of clusterhead role, it is possible that heavy burst of traffic takes place causing some clusterheads to use maximum resources, while others encounter low traffic bursts resulting in minimum resource use. Thirdly, the fair share or load balancing technique [2], might result in a clusterhead that will not provide the optimal path for routing, or yet a link breakage. However, nodes which are not elected as clusterheads don’t pay a routing penalty and has its resources dedicated for its own usage. Therefore, there is no one common clusterhead election mechanism that is best for MANET systems, without some tradeoffs.

The Zone Routing Protocol, ZRP, in [12], provides a hybrid approach between proactive routing which produces added routing control messages in the network due to keeping up to date routes, and reactive routing which adds delays due to path discovery and floods the network for route determination [12]. ZRP divides the network into overlapping zones, while clustering can have distinct, non overlapping clusters. In ZRP, Proactive routing is used within the zone, and reactive routing is used outside the zone, instead of using one type of routing for the whole network. The authors in [12] selected such an approach to minimize the flooding of the reactive routing in the whole network. In addition, the authors in [11-12] claim this hybrid approach is suited for large networks, enhances the system efficiency, but adds more complexity. Each node has a routing zone within a radius of n hops. All nodes with exactly n hops are called peripheral nodes, and the ones with less than n are called interior nodes. This process is repeated for all nodes in the network. A lookup in the node’s routing table helps in deciding if the destination node is within the zone resulting in proactive routing. Otherwise, the destination is outside the zone, and reactive routing is used which triggers a routing request. As a result of a routing response, one of the peripheral nodes will be used as an exit route from the zone to the destination. While, if clustering is applied, the same elected clusterhead is used for routing outside the cluster without triggering any route discovery to the destination.

As discussed above, the main focus of the previous work focuses on an election of one clusterhead for a cluster. Even though this minimizes the overall signaling overhead in the cluster, but it mainly can make the central clusterhead a bottleneck.

APPROACH

The base of our research is the distinct clustering approach where one clusterhead is elected. We enhanced the architecture to use multiple clusterheads, one for each neighboring cluster. Our contribution distributes the load of the clusterhead amongst multiple clusterheads in the same cluster. The proposed mechanism does not mandate a specific clusterhead election process. Any of the prior work can be used to select the clusterheads for a cluster. By distributing the load, a single clusterhead does not have to bear all the added responsibility of being the central point for routing in a cluster. Therefore, we believe this approach provides a more fair solution of sharing inter-cluster routing responsibilities for a cluster. In addition, other mechanism can be applied to switch the responsibility of a clusterhead to another node, such as in [2]. In the case of one clusterhead per cluster, a link breakage caused by the failure of the clusterhead isolates all cluster nodes from communicating to/from outside the cluster. However, our approach reduces the link breakage to be only in the direction towards a path where the failed clusterhead forwards the data. Therefore, the reliability of routing in the MANET system is increased. Our architecture does not state the routing protocol as in [12], but leaves the decision as done in the original designs of clustering [2][4][16][19-21].

SYSTEM MODEL

We have used glomosim [22] simulator, running IEEE 802.11 to prove our contribution. Our MANET system consists of 6 distinct non-overlapping clusters with a physical terrain of 2000 meters by 2000 meters as shown in Fig. 1. For the same cluster, we ran simulation experiments with one clusterhead, and compared its performance results with tests using multiple clusterheads. During the simulation, we maintained the same clusterheads in both cases (single, multiple clusterheads), since changing the clusterhead was irrelevant to what we are proving. Our traffic type has Constant Bit Rate, (CBR), and File Transfer Protocol, (FTP), traffic. The same traffic load was run for both cases (single, multiple clusterheads). The selected traffic load was chosen based on tests that allowed sufficient utilization of the channel.

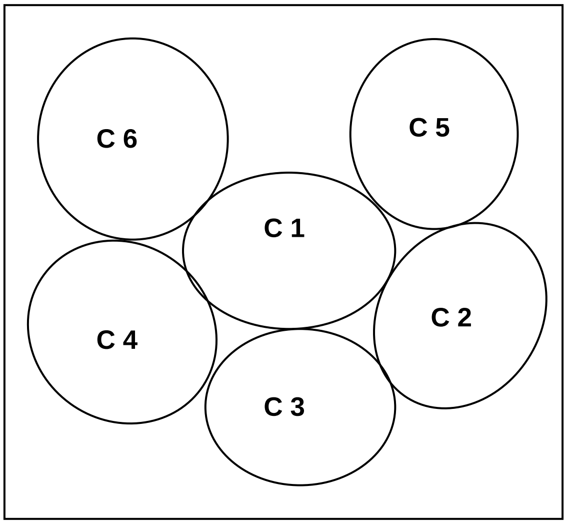

Fig. 1: Simulation Clustering Setup

Cluster 3 is the one which we had as a cluster with one clusterhead and with multiple clusterheads. The remaining clusters 1-2, and 4-6, operated with one clusterhead. Our traffic included FTP traffic generated between nodes in cluster 3 and nodes in all other clusters in the MANET system. The FTP sessions where established in both directions. In addition, CBR traffic was generated in both directions between nodes in cluster 3, and clusters 5, and 6. In order to focus on the objective of distributing the clusterhead load, we setup static routes in our MANET system. Routing from cluster 3 to cluster 5 was done via the intermediate cluster 2, and vice versa. In addition, routing from cluster 3 to cluster 6 was done via the intermediate cluster 4, and vice versa. For example, in the case of one clusterhead in cluster 3, to send a packet from a node in cluster 3 to a node in cluster 5, the packet is routed via the single clusterhead in cluster 3, then through the clusterhead in cluster 2, and finally via the clusterhead in cluster 5. However, routing from cluster 3 to clusters 2, and 4, was done directly from the clusterhead in cluster 3 to the clusterheads in clusters 2 and 4.

In the case when using multiple clusterheads in cluster 3, a different clusterhead was used for routing to each of the neighboring clusters. For example, routing from cluster 3 to cluster 1, uses a clusterhead that is different when routing from cluster 3 to clusters 2 and 5, which is also a different clusterhead when routing to clusters 4, and 6. Therefore, since there are 3 neighboring clusters to cluster 3, the system allowed for the use of 3 clusterheads, one for routing to/from each neighboring cluster.

NUMERICAL RESULTS

Our simulation provides cumulative averaged results for the whole MANET system. We focused on the cumulative averaged throughput and response time. We believe analyzing the cumulative results provides less bias results towards a special node in the system where the distribution may provide a higher advantage. Our reference to all performance numbers, (response time, throughput, channel error ratio), indicates the results of the “overall averaged cumulative”. We may refer to it without the phrase “overall averaged cumulative” to simplify the text.

Fig. 2 shows the percentage of increase in cumulative averaged throughput when running multiple clusterheads over using one clusterhead in cluster 3. In all cases, the throughput increased for the multiple clusterheads case. For the small simulation time of 1000S and with the traffic load used, the increase was only about 18% since the system was lightly loaded as a result of a short simulation time. Therefore, one clusterhead operated well since the channel was not well utilized. Our peak results show that at 7000S of simulation time, we reached a maximum throughput improvement as this case indicates the channel utilization was at its optimal condition. Therefore, for the longer simulation times, beyond what we concluded as optimal, the throughput decreased due to the added traffic on the channel as we concluded from the channel error statistics.

The rest of our results, focus on the optimal case of 7000S to prove the advantage of distributing the load to multiple clusterheads.

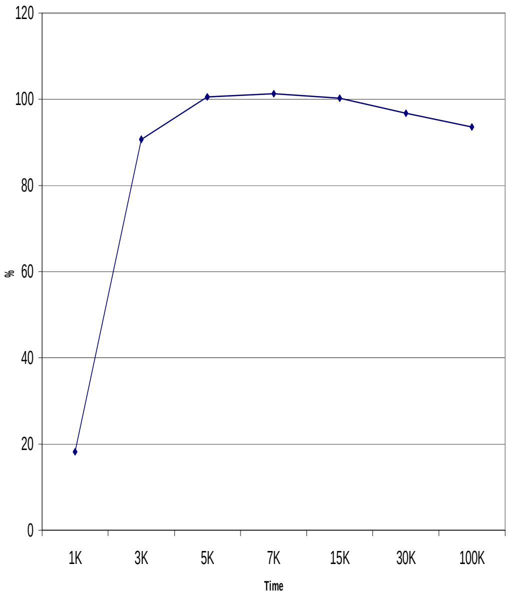

Fig. 2 : Simulation Time (sec) VS Throughput Percentage Improvement

Our results for the 7000S simulation time for the traffic type above, that we have gained about 101% improvement in throughput. Our results are explained by the simple queuing theory model:

$\rho = \lambda / \mu$ ………. (1) where,
ρ is the utilization rate or traffic intensity
λ is the traffic arrival rate
μ is the service rate

(1), indicates that ρ increases if the λ increases while μ remains at the same rate. In addition, the overall averaged cumulative response time, (is the round trip time which includes processing, queuing, and transmission time from the time of sending the message until receiving its response by the source), increases if a constant service rate is maintained, while the traffic arrival rate increases. Our simulation showed that the response time remained constant when using one single clusterhead, and multiple clusterheads of about 0.5. The traffic rate in the system increased as indicated by the throughput increase due to the multiple clusterheads, while maintaining the same response time. Normally, if the arrival rate increases while maintaining the same service rate, then the response time should increase accordingly. Therefore, we can conclude that, by maintaining the same response time, the added traffic rate due to an increase in service rate results in constant system utilization. In addition, there was no impact to the overall system capacity, and the MANET system is still capable of handling more traffic, which is proven by the higher throughput when using multiple clusterheads.

In our topology, we increased the number of clusterheads to 3. However, our throughput is about doubled as shown in **Fig. 2**. We should expect by the distribution of work to 3 clusterheads, and by having the same averaged delay for the MANET system, a 3 fold increase in throughput since the service rate has tripled. However, we only gained double the throughput. In addition, as the traffic arrival rate increased due to having the 3 clusterheads, the service rate also increased, resulting in the same utilization rate for the MANET system. Additionally, we have also noticed that the cumulative increase in overall averaged channel error rate has doubled when increasing the number of clusterheads to 3. Therefore, we conclude that our throughput did not increase to 3 fold, as a result of doubling the channel error rate due to the added traffic rate by having multiple clusterheads. Hence, we derive:

Throughput rate for one clusterhead:
σ … (2)
Expected throughput rate for 3 clusterheads:
$3 * \sigma$ ... (3)

(3) is concluded, since the service rate has increased by 3 fold, while maintaining the same arrival rate, then the utilization will decrease by 3 fold. However, the throughput should still increase by 3 fold.

However, we conclude that the ratio of increase in channel error rate affects the actual throughput as follows:

Actual throughput rate for 3 clusterheads:
$\gamma = (3 * \sigma) - (\lambda)$ ... (4)
Where, **λ** is overall cumulative channel error ratio increase, and $\lambda = \sigma$, meaning that a 1 ratio of channel error rate increase equals 1 ratio of throughput decrease.

Therefore, we conclude that as long as the system utilization and the averaged response time remain constant, then the number of clusterheads increase should improve throughput by a factor equals to the number of added clusterheads, minus the rate of increase for the channel error rate for the system. This indicates that the increase in the arrival traffic rate did not increase the utilization of the system since the response time remained constant.

We ran additional test to validate the traffic rate at our selected simulation time of 7000S. The tests were run with one clusterhead and multiple clusterheads for cluster 3. The throughput results are presented in Fig. 3. The results show the percentage of increase in the averaged cumulative throughput for running multiple clusterheads over one clusterhead. Our selected rate is labeled as “med” for medium. We ran traffic rates at ½ of the medium which we called low traffic rate. In addition, we ran traffic rate at double of the medium which was called High traffic rate. Finally, we ran much lower traffic rate than the low traffic rate which we called very low rate traffic. We have noticed, as shown in Fig. 3, the percentage of throughput improvement for the very low was only at about 49%. This is attributed to the low channel utilization by the low traffic rate. At the high traffic rate we have shown a reduced improvement in throughput due to a higher channel error rate resulting from the traffic overload in the MANET system. This traffic overload was created by the higher arrival rate due to the added sessions. However, at ½ of our selected traffic rate, we obtained about the same level of throughput improvement as our optimal selected rate. We conclude that at these rates we obtained system stability with the offered traffic and service rates. Therefore, the results shown in Fig. 3 validate the selected traffic for our results above.

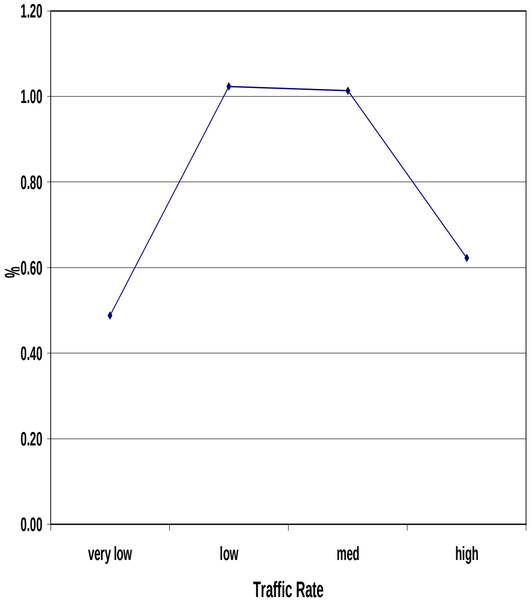

Fig. 3 : Traffic Rate vs % Throughput Improvement for 7000S

CONCLUSIONS AND FUTURE WORK

Our contribution proves that one clusterhead per cluster does not provide for a maximized throughput of the MANET system due to the added responsibility for the one clusterhead. Using multiple clusterheads per cluster distributes the load among multiple nodes which enables simultaneous and shared responsibility of inter cluster routing among multiple nodes. In addition, routing reliability is increased since a failure of one clusterhead does not break all routing to outside the cluster. It is an interesting finding to realize that the increase in throughput due to the added clusterheads is proportional to the number of clusterheads, and is indirectly proportional to the ratio of channel error rate. Depending on the topology and traffic pattern, if all clusterheads are simultaneously used to route traffic, the rate of throughput increase is the multiplier of the original throughput when using one clusterhead by the total number of clusterheads used minus the increase in channel error rate.

It is suggested to do further research when having all clusters employing multiple clusterheads, one per neighboring clusters. It is expected that the throughput will increase at a very high rate, and to compare this new approach with ZRP[11], for system efficiency. Another work is suggested to dynamically elect the clusterhead and provide some type of load balanced approach.

REFERENCES

1. A. Amis, P. Prakash, D. Huynh, and T. Vuong, , Max-Min D-Cluster Formation in Wireless Ad Hoc Networks, Infocom 2000, The Conference on Computer Communications, Volume 1, pp 32-41., Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies.
2. A. Amis, and R. Prakash, March 2000. Load-Balancing Clusters in Wireless Ad Hoc Networks. In Proceedings of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00), pp 25.
3. D. J. Baker, A. Ephremides., and A. J. Flynn, 1984. The Design and Simulation of a Mobile Radio Network with Distributed Control. *IEEE Journal on Selected Areas in Communications*, COM-2(1), pp 226–237.
4. D. J. Baker, and A. Ephremides, November 1981. The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm. *IEEE Transactions on Communications*, COM-29(11): pp 1694–1701.
5. J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, October 1998. A performance comparison of multi-hop wireless ad hoc routing protocols. In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking.
6. P.Y. Chen, and A.L. Liestman, April 2003. Zonal Algorithm for Clustering An Hoc Networks, International Journal of Foundations of Computer Science, in a special issue dedicated to Wireless Networks and Mobile Computing, Vol. 14, No. 2, pp 305-322.
7. C. Chiang, H. Wu, W. Liu,, and M. Gerla, Apr. 1997. Routing in clustered multihop, mobile wireless networks with fading channel. IEEE SICON, pp. 197-211.
8. B. Das, and V. Bharghavan, 1997. Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets. In *Proceedings of ICC.*
9. S. Das, Castaneda, Yan, J., and R. Sengupta, October 1998. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In Proceedings of 7th Annual ICCCN.
10. M. Denko, 2003The Use of Mobile Agents for Clustering in Mobile Ad Hoc Networks, In Proceedings of ACM International Conference, Vol. 47, pp 241 – 247.
11. Z. J. Haas, and M. R. Perlman, The performance of query control schemes for the zone routing protocol. In Proceedings of ACM Sigcomm’98, Vol. 28, No. 4, pp 167 – 177, October 1998.

12. Z. J. Haas, and M. R. Pearlman, 2000. “The zone routing protocol: A hybrid framework for routing in *ad hoc* networks,” in *Ad Hoc Networks*,C. E. Perkins, Ed., Chapter 7, Addison-Wesley.
13. I. Er, and W. Seah, July 2005. Clustering Overhead and Convergence Time Analysis of the Mobility-Based Multi-Hop Clustering Algorithm for Mobile Ad Hoc Networks, ICPADS, The 11th International Conference on Parallel and Distributed Systems.
14. I. Er, and W. Seah, March 2004. Mobility-based d-Hop Clustering Algorithm for Mobile Ad Hoc Networks. In Proceedings of WCNC’04,
15. X. Hong, M. Gerlo, Y. Yi, K. Xu, and T. J. Kwon, Scalable Ad Hoc Routing in Large, Dense Wireless Networks Using Clustering and Landmarks, In Proceedings of the IEEE International Conference on Communications. (ICC'02), Apr. 2002, vol. 25, no. 1, 3179 -- 3185.
16. M. Gerla, and J. Tsai, 1995. Multicluster, Mobile, Multimedia Radio Network. *ACM Journal on Wireless Networks*, Vol. 1, No. 3, pp 255-265,
17. L. Kleinrock, and J. Silvester, January 1987. Spatial Reuse in Multihop Packet Radio Networks. *Proceedings of the IEEE*, 75(1), pp156–167,
18. A. McDonald, T. F. Znati, Aug. 1999. A mobility-based framework for adaptive clustering in wireless ad hoc networks. IEEE JSAC, 17(8):1466-1486,
19. F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic , 2003. Connectivity Based *k*-Hop Clustering in Wireless Networks, Telecommunication Systems Journal, Vol. 22, No 1-4, pp 205-220.
20. S. Sivavakeesar, and G. Pavlou, Stable Clustering Through Mobility Prediction for Large-Scale Multihop Intelligent Ad Hoc Networks, In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'04), Georgia, USA, Mar. 2004, vol. 3, 1488 -- 1493.
21. Y. Wang, W. Wang, X. Li, 2005. Distributed Low-Cost Backbone Formation for Wireless Ad Hoc Networks, In the Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, pp 2 – 13,
22. Web site for glomosim simulator, http://pcl.cs.ucla.edu/projects/glomosim/