Distance-aware Overlay Routing with AODV in Large Scale Ad Hoc Networks

Ying Liu*, Xiruo Liu*, Wade Trappe*, Radhika Roy†
WINLAB Rutgers University*, Email: {$yingliu, sissi, trappe$} @winlab.rutgers.edu
Army CERDEC†, Email: {$radhika.r.roy$} @us.army.mil

***Abstract*—Overlay networks are a beneficial approach to designing robust and specialized networks on top of the generic IP architecture, and have been applied to the operation of mesh and mobile ad hoc networks. Unfortunately, when routing between entities in the overlay, inefficiencies are incurred due to potential "back tracking" that arises because of the discrepancies between the overlay and underlay topologies. In this paper, we minimize the "back tracking" problem by applying physical contexts shared by the network layer with the overlay so as to efficiently guide application flow. We have devised an intelligent cluster head and path selection algorithm for our overlay routing and compared its performance with the popular Chord protocol and a baseline AODV routing protocol. Simulation results indicate that: 1) the integration between logical and physical routing gives a large improvement in the number of hops for each transmission path; and 2) the selection of a good cluster head has only a moderate increase in transmission time.**

I. Introduction

Mesh and ad hoc networks are a means for mobile users to rapidly establish a network without a dedicated infrastructure. They have been proposed for a variety of scenarios, ranging from tactical networks to supporting urban infrastructure to sensor network deployments. Popular Internet applications, such as Skype and Youtube, will become increasingly prevalent on mesh/ad hoc networks. These popular applications require that communications experience limited delay. To support such applications in areas where there is no wide-area communication coverage (e.g. in tactical environments, or in remote areas removed from cellular coverage), it is necessary to employ peer-to-peer communication methods, such as mesh or mobile ad hoc networking (MANETs). This work aims to reduce the delay associated with packet delivery for peer-to-peer communications in large ad hoc networks, while maintaining reliable communication.

Unfortunately, the basic "physical" routing that is performed at the network layer does not scale well versus the number of nodes in an ad hoc network, and there are significant network-layer routing challenges that should be hidden from applications, e.g. in table-driven proactive ad hoc routing, the communication overhead scales as $O(n^2)$, where n is the number of nodes in the network. Similarly, for on-demand routing, although routing information is transmitted only when needed, these protocols flood the route discovery messages (RREQ) which introduces a heavy load. One approach to work around non-scalability in ad hoc networking involves hybrid routing, such as ZRP [1], which combines IARP proactive and IERP reactive routing. Another solution involves using a hierarchical network structure, where nodes are grouped into several clusters via clustering algorithms like k-Means clustering or BFR (Bradley-Fayyad-Reina) clustering [2]. The cluster head serves as a gateway and thus the clustering technique decreases the network flow between nodes. Routing among cluster heads is carried out at a higher-layer, via an overlay, leading to what will be referred to as "logical" routing in this paper. Logical routing is more application-oriented and usually does not take into consideration the physical routing underneath.

This paper explores the performance of hierarchical routing at the overlay level when used with the popular AODV routing protocol [3]. We adopted a cluster-based architecture and the applications we are concerned with are those involving latency constraints, such as peer-to-peer (P2P) VOIP applications between clients in an ad hoc network. Many of these P2P applications are built using an overlay protocol, such as Chord, which is suitable for P2P file sharing and web searching [4]– applications that do not demand strict time delivery. For applications like VOIP, Chord's performance cannot meet the required QoS delay constraints. This arises because Chord routing is unaware of the relative geographic position of two callers, and the effect is that standard Chord routing at the overlay leads to significant network-layer "back tracking". For example, packets that should be delivered to New York from New Jersey are first routed to California which is geographically far away and then forwarded back to NY state. Further, in ad hoc network settings, Chord-based routing does not take advantage of other cross-layer information available that might be desirable to use when forming routes, such as network topology, device power resources, or even link quality. Our contribution in this paper is to address the "back tracking" problem by building a distance-aware overlay routing protocol. We use physical connectivity to guide our logical routing and implement a distributed distance table (DDT) in a manner similar to the distributed hash table (DHT) in Chord to find the shortest path among cluster heads, where the underlay routing protocol is AODV. In Section II, we give a brief introduction to Chord and AODV, which are closely related to our protocol. In Section III, we examine reasons why it is desirable to take the underlying geographic distance into consideration when routing. After the analysis, we describe our context-aware routing protocol in detail. In Section IV, simulation results are presented, and the paper is concluded in Section V.

978-1-4799-4449-1/14/$31.00 ©2014 IEEE

II. Related Material

In this section, we briefly summarize Chord and the AODV protocol, which serve as the motivation and basis for this work. Chord [5] is a P2P overlay routing protocol based on distributed hash tables (DHTs), and is generally considered one of the four DHT protocols, along with Tapestry [6], CAN [7] and Pastry [8]. In Chord, nodes in a network compose a logical ring, which can contain at most 2^m nodes. m is the length of node identifier. These nodes are uniformly positioned on the logical ring in a random order. Each node has its own unique IDs and "keys". The node ID is mapped to the DHT by the hash function: $n + 2^{k-1} \bmod 2^m$, $k \in [1, m]$, where n is the current node ID. When $k = 1$, we get the ID for the first successor of n, when $k = 2$, we get the second successor, and so on. This hash function makes adding and removing a node in DHT collision free. Each DHT table can contain up to m entries. Each node maintains a separate DHT table when nodes dynamically join and leave the chord circle. During the routing process, the source node first finds whether the target node ID is in its finger table. If not, it finds the closest predecessor to the target node in its DHT and then passes the request to that node. This process occurs recursively until the target node is reached.

The ad hoc on demand distance vector (AODV) routing protocol [9] is one of the more popular mobile ad hoc network routing protocols, along with OLSR [10]. In AODV, when a connection request occurs, the source node broadcasts a route request (RREQ) message, which carries a source identifier, a destination identifier and a destination sequence number (DestSeqNum), to its neighbors. Then the neighbors flood the RREQ to other nodes for route discovery. Any intermediate node that receives the RREQ can either forward the request or send a route reply (RREP) to the source node if its routing table has a valid route entry to the destination. One of the primary advantages of AODV is that the route is built *on demand*, which significantly decreases the network load, while disadvantages for AODV include higher latency for finding routes and overhead associated with sharing stale route entries.

III. Context Aware Routing

A. Why distance aware

Chord does not consider the physical distance between nodes, and rather places all nodes randomly on the Chord ring. Therefore, when performing routing using Chord, packets can be directed to an area that is far from the ideal source-destination path, which results in additional transmissions and extra latency.

Another weakness that results in Chord having a higher failure probability is that Chord routing does not consider the existence of degraded links in the underlay. The lack of a connection to actual network link performance can lead to large failure rates. We now examine the failure probability of delivering a packet using Chord. We first assume every cluster head ID is on the Chord ring so as to exclude from our analysis the failure probability of being unable to find a node ID on

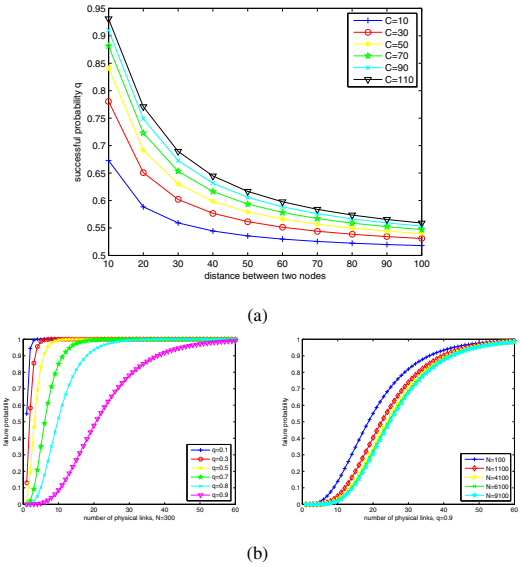

Fig. 1. (a) The probability that a physical link experiences failure increases versus increased node separation distance d. (b) The physical path success is related to the probability of a physical layer link being successful q, as well as to the number of physical links, but shows limited dependence on the number of nodes in the network.

the Chord ring. Therefore, the failure probability of one logical link on the ring is solely caused by the failure probability of underlay's broken links. Assume r is the expected number of physical links that each logical link contains and q is the success rate of each physical link that every logical link contains. Then the probability for one logical link to fail is equivalent to anyone of the composed physical links failing. The expected probability for one logical link to fail is $E(p) = (1 - q^r)$, while the probability a node cannot find a destination node on the logical ring is $p^{\log N}$. N is the number of nodes on the Chord ring. Then the final probability for a node to be unable to deliver a packet to the destination is $Q = (1 - q^r)^{\log N}$. The successful transmission probability q can be represented by the distance between nodes. For simplicity of calculation, we use the bit error rate (BER) for BPSK as a surrogate for packet error rate $1 - q$. To recall, the BER for BPSK is $BER = \frac{1}{2} erfc(\sqrt{\frac{E_b}{N_0}})$, where $\frac{E_b}{N_0} = SINR\frac{B}{R_b}$ and B is the bandwidth of the signal, and R_b is the transmission bitrate. The BER decreases as $\frac{E_b}{N_0}$ increases, and thus, with B, R_b fixed, $\frac{E_b}{N_0}$ is proportional to $SINR$. In our work, we have used the Friis equation to arrive at the SINR estimate for a pair of nodes with a separation d, which yields the successful probability q for one physical link as $q = 1 - BER = 1 - \frac{1}{2} erfc(\sqrt{\frac{C}{d^2}})$. Here, $C = \frac{P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \frac{B}{R_b}}{N_0}$ and incorporates transmit power P_t, and transmitter and receiver gains G_t and G_r. Using these equations, we illustrate the relationship between geographical distance and physical link success in Figure 1.

Figure 1 (a), shows that the success probability of one physical link decreases when the distance between a pair of nodes increases, and thus distance affects the design of the

overlay routing architecture. In Figure 1 (b), the failure probability of a physical path built by Chord increases substantially when the expected number of physical links composing one logical link increases. It is also shown that the high successful probability of one physical link could allow one logical link to accommodate more physical links. For example, to maintain a failure probability below 0.2, for the number of nodes $N = 300$, it is better to maintain the number of physical links between each cluster head below 14 as q is below 0.9.Although the failure probability of Chord, $O(\frac{1}{N})$, decreased with N, N has little effect on the failure probability of a physical path because there is also a relationship involving q and r. For example, to maintain the failure probability below 0.2, for N up to 9000 with success rate $q = 0.9$, the expected number of physical links should be around 15 hops.

B. Overlay Topology Structure

Since we intend for our overlay routing algorithm to work on large-scale networks, we need to employ hierarchies in the overlay topology. A depiction of the network topologies in this work is presented in Figure 2. For a given ad hoc network, in the overlay nodes are grouped into clusters. For cluster formation, we group nodes according to their mutual distance from each other. Each cluster has its own cluster head which is chosen by a cluster head selection algorithm. One possible cluster head selection algorithm is to choose the node with the largest remaining power energy in the cluster as the cluster head, or choose a node as the cluster head who could lead to the minimum total energy consumption [11] [12]. We note algorithm [12] also considered balancing the traffic loads for the cluster head in order to reduce the total amount of energy consumed by the whole cluster. As another example, the overlay could simply choose a cluster head which is closest to being the geographical center of the cluster. In our design, we wanted the overlay routing to minimize the path length of the underlay network being traversed, and thus an ideal cluster head was the one that had the minimum number of hops to every other node in a cluster. The objective function in our case is:

$$CH = argmin_{i \epsilon S} \sum_{j \epsilon S \setminus i} hop(i, j) \quad (1)$$

where S is the collective set containing all the node IDs in the cluster. $hop(i, j)$ is the number of hops from node i to node j. $S \setminus i$ refers to a set which does not contain i.

C. Overlay Routing with AODV: DDT-AODV

Modifying Chord so that it becomes distance-aware is not a simple task. For a new overlay routing protocol to be distance aware, we cannot simply change the Chord DHT table to contain a list of the k of nearest cluster heads without also needing to change the hash function associated with finding closest preceding node. Because the closest preceding function searches the nodes in reverse order and found a node which was farthest to the source. In order to reduce the back and forth, or inefficient paths as mentioned previously, we need to introduce a new process. The new process is to determine

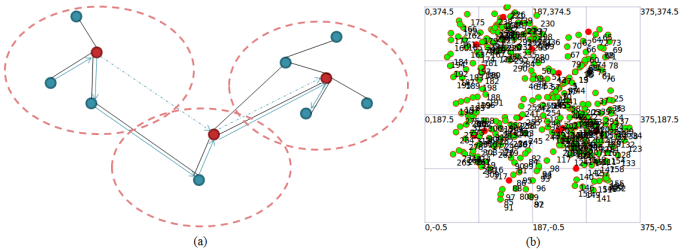

Fig. 2. Overlay and underlay topology: (a) the solid line represents the physical underlay links between nodes, while the dashed line represents the overlay logical link between cluster heads; (b) is a topology used in our simulations, the red circles were cluster heads, the green circles were ordinary nodes.

the precedence of a node based on its distance from the starting point. The precedence ranks its corresponding nodes in the order based on their relative distance to the starting point, with the shortest distance node on top of the list. The optimal path is the node selected based on its shortest distance from the starting point where the route request was first initiated. Therefore, the starting point for calculating the distance changes every time a route request appears. This alteration could result in a huge computational overhead.

In our algorithm, each cluster head stores the k nearest cluster heads and builds an adjacent link list among these cluster heads. An adjacency link list corresponds to one topological graph among the cluster heads. The overlay routing found the shortest path in this topological graph. We note that this topological graph could either be directed or undirected. All that matters are the entries of the DDT table. We could also set the graph to be completely connected or partially connected or sparsely connected. This could be done by tuning the length, k, of the DDT table. Another advantage of tuning the length k is that it also let us make a tradeoff between the number of hops of the logical path, which is related to packet delivery time, and the complexity of finding a shortest path.

In our routing scheme, DDT-AODV, we use AODV for network layer routing. When an application sends a data packet to a node, the cluster head updates geographic positions of other cluster headers through the RREQ and RREP of AODV, which is easily modified to convey location information (e.g. GPS). Each cluster head maintains a distributed distance table (DDT) containing up to k nearest cluster heads. This was done using the RREQ and RREP message of AODV. We modified the RREQ message to carry the position of nodes it encounters as it traverses the network. The cluster head also contains other cluster heads' distance tables. The update of another cluster head's DDT table was also conveyed through the RREQ and RREP of AODV. A second approach is for each cluster head to send their distance tables to a back-up server. The back-up server then takes care of updating the DDT table of the cluster heads and the calculation of the logical path between source and destination heads according to the new DDT. The updating of DDT need only happen when there's a routing request. After figuring out the logical path, the back-up server

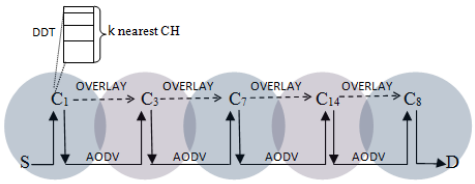

Fig. 3. In context-aware overlay routing each cluster head stores a distributed distance table that contains k nearest cluster heads. Overlay logical routing is taken between cluster heads. AODV routing was used to connect cluster heads on the logical path. The colored circles represent cluster regions, which may overlap.

sends this logical path to the corresponding cluster heads on the logical path.

We adopted the first method since we wanted to avoid having a single point of failure in the network. Our main goal was to compare the length of the path after it was built and the failure probability caused by ignoring the underneath physical link status. When there is a search request, each cluster head updates their distance tables. Only the newly updated entry of the routing table needs to be passed. Then the cluster head calculates the shortest path between the source and destination cluster head and hands over the logical path to the underlay (AODV) for actual network layer routing. The header of the AODV data packet contains the logical path for routing purposes when transmitting. The source node first wraps the data packet and tunnels it to the next cluster head on the logical path. Then the received cluster head unwraps the packet and sets the next cluster head on the logical path as destination and tunnels it out. This process run continuously until data packets reach the destination. AODV finds the real physical path to connect each cluster head on the logical path. Since we assumed each cluster moved very slowly, the re-building of a path did not occur frequently. Figure 3 shows the whole routing mechanism across overlay routing and AODV. Finally, we note that other metrics, such as channel bandwidth, data rate, and node energy could be used to rank the entries of DDT table.

IV. Simulation Results

Our simulation results were obtained using the popular network simulator, NS-3. We used networks consisting of 320 nodes. These nodes were placed in a square area with length 500 units. Several topologies were produced, each composed of 16 clusters and each cluster contained 20 nodes. We required that, within each cluster, the average hops between node and cluster head should be larger than 4, yet no larger than 15. We also set the radio range for each node to be 20 units so that two nodes were not connected when the distance between them was more than 20 units. Under these requirements, we produced 15 topologies. We compared our context-aware protocol with Chord using these 15 topologies. Since our goal was to prove the hop efficiency of the routing protocol, we fixed the cluster heads for each topology at the start of the simulation and kept the cluster heads fixed during the simulation. The ideal cluster head has the minimum average number of hops to every other node within the cluster since the upper layer application, such as VOIP, require low latency delivery of voice packets. We also wanted to see the effect of a "bad cluster head" on increasing routing hops, so "bad" cluster heads were randomly chosen within a cluster.

For the network layer, AODV was implemented. The length of the DDT table was set to be 5. We compared our DDT context-aware routing with AODV routing without clustering and Chord-AODV with min-hop and a random cluster head (CH). Each simulation was run 100 times. Figure 4 shows the average number of hops a packet went through for each protocol. The simulation demonstrated that our distance-aware DDT routing gave much fewer average hops than Chord for cases where Chord-based routing used mini-hop and random cluster head selection. Even with a bad cluster head, DDT-AODV performed almost same as Chord with good cluster head. Compared to Chord, DDT-AODV with minimum hop cluster head selection had performance closest to the ideal of using the baseline AODV. To present clearly, Figure 5 shows the additional cost of DDT-AODV and Chord-AODV compared to baseline AODV, which is calculated as:

$$\frac{average\_hop - baseline\_AODV\_hop}{baseline\_AODV\_hop} \quad (2)$$

Figure 5 implies that DDT-AODV with minimum hop cluster head results in the least additional hops when compared with Chord-AODV. Although Chord with mini-hop cluster head selection sometimes performed better than DDT-AODV, its performance was unstable and inconsistent across topologies. Chord had a larger variance compared to DDT-AODV with good cluster head selection. This big variance was caused by Chord protocol positioned node randomly on the Chord ring without the consideration of the geographic position of nodes. Therefore, some extra hops occurred in the network path. This extra number of hops had a large variance because of the underlying randomness. This large variance reveals the general behavior associated with "back tracking" in Chord-selected paths. We also conclude that choosing a good cluster head according to application requirements affects the performance of routing protocols. A good cluster head was one which improved the QoS of the applications running on top of it. For voice applications, the simulation results indicated that cluster head which considered actual physical distance was a wise choice. Although DDT-AODV does not perform the best for all cases, its average behavior is better than Chord-based methods and this translates into it having desirable delay properties, making it suitable for multimedia applications.

Figure 6 shows the path failure probability for different routing schemes for the 15 topologies. In Figure 6, DDT-AODV with minimum hop cluster head had a low failure probability compared to Chord. Its failure probability was always below 0.2, showing consistency across topologies. However, Chord failure probability was frequently large. The reason for Chord's large variance and inconsistency was the "back tracking" caused by nodes randomly positioned on

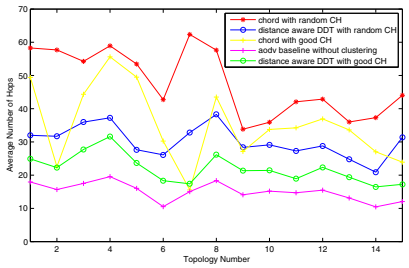

Fig. 4. Average hops of DDT-AODV, Chord-AODV and baseline AODV for 15 topologies

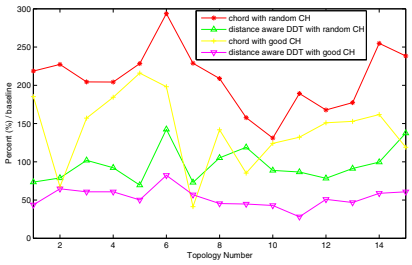

Fig. 5. Additional cost of DDT-AODV and Chord-AODV for 15 topologies

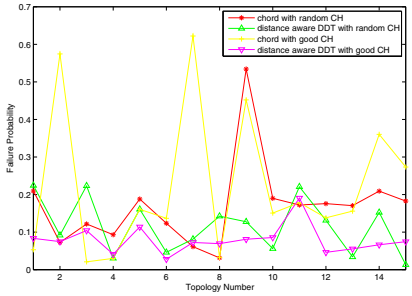

Fig. 6. Failure probability of DDT-AODV and Chord-AODV for 15 topologies

Chord ring without considering the underlying broken links. The logical path it chose could not successfully deliver the packet to the destination when the AODV underlay could not find any network path connected to the destination. The longer path, as expected, has an increased failure probability since the failure probability of the whole path depends on the number of physical links.

When comparing the effects of cluster heads, choosing a good cluster heads was more important for DDT-AODV than Chord. The failure probability of DDT-AODV with minimum hop cluster heads was always under that of DDT-AODV with random head selection, while Chord's performance was almost the same regardless of cluster head selection.

V. Conclusion

We analyzed the weakness of Chord-based overlay routing in ad hoc networks. We pointed out that Chord routing usually experiences a "back tracking" problem since it randomly distributes nodes on the Chord ring without consideration of the physical/network position of nodes. We proposed a distance-aware overlay routing scheme that uses a lower layer AODV routing protocol to guide network-level packet delivery. Our protocol, DDT-AODV, involves cluster heads updating each other with their physical position and ranking the closest cluster heads in their DDT table. When there is a route request, logical routing is first calculated among cluster heads then passed to the lower tier nodes. Lower tier nodes use AODV to find the network path between themselves and next hop cluster heads. NS-3 simulations showed that: 1) DDT-AODV out-performed Chord, both with mini-hop and random cluster heads, in terms of additional cost; 2) DDT-AODV with good cluster head selection had dramatically reduced failure probability. DDT-AODV always showed consistency in performance across different topologies: the differences in terms of average hops for 15 topologies were small compared to that of Chord-AODV.

References

[1] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (zrp) for ad hoc networks," *draft-ietf-manet-zone-zrp-04. txt*, 2002.

[2] U. M. Fayyad, C. Reina, and P. S. Bradley, "Initialization of iterative refinement clustering algorithms." in *KDD*, 1998, pp. 194–198.

[3] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications, IEEE*, vol. 6, no. 2, pp. 46–55, 1999.

[4] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, "Minerva: Collaborative p2p search," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 1263–1266.

[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4. ACM, 2001, pp. 149–160.

[6] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 1, pp. 41–53, 2004.

[7] S. Ratnasamy12, P. Francis, M. Handley, R. Karp12, and S. Shenker, "A scalable content-addressable network," 2001.

[8] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Middleware 2001*. Springer, 2001, pp. 329–350.

[9] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*. IEEE, 1999, pp. 90–100.

[10] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, L. Viennot *et al.*, "Optimized link state routing protocol (olsr)," 2003.

[11] M. C. M. Thein and T. Thein, "An energy efficient cluster-head selection for wireless sensor networks," in *Intelligent systems, modelling and simulation (ISMS), 2010 international conference on*. IEEE, 2010, pp. 287–291.

[12] A. Thonklin and W. Suntiamorntut, "Load balanced and energy efficient cluster head election in wireless sensor networks," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*. IEEE, 2011, pp. 421–424.