Virtual structure routing in ad hoc networks

Fabrice Theoleyre, Fabrice Valois

To cite this version:

Fabrice Theoleyre, Fabrice Valois. Virtual structure routing in ad hoc networks. International Conference on Communications. ICC, May 2005, Seoul, South Korea. pp.3078-3082, 10.1109/ICC.2005.1494962 . hal-00371394

HAL Id: hal-00371394

https://hal.archives-ouvertes.fr/hal-00371394

Submitted on 27 Mar 2009

Virtual Structure Routing in Ad hoc Networks

Fabrice Theoleyre, Fabrice Valois
CITI - INRIA ARES, INSA Lyon
21, Avenue Jean Capelle, 69621 Villeurbanne Cedex, France
Email: {fabrice.theoleyre, fabrice.valois}@insa-lyon.fr

***Abstract*— Routing protocols are the main issue of ad hoc networks. Because flat propositions (reactive, proactive) are not sufficient and suffer of lack of performance, new solutions should be investigated and proposed. In the other hand, virtual topologies propose to structure the network and to give a hierarchy between the *strongest* and the *weakest* nodes. We propose a new routing protocol, *Virtual Structure Routing* (VSR), based on a virtual topology including both a backbone and clusters. The backbone is used to collect control traffic and to reduce overhead for route discovering. VSR uses clusters to define a route as a list of clusters Id. This cluster topology is more stable than the physical topology. Hence, routes are more robust. VSR combines the assets of both flat approaches: intra-cluster routing is proactive while inter-cluster routing is reactive. Finally, routes are computed dynamically and a mechanism of route repair is proposed.**

I. Introduction

MANet could be defined as *spontaneous wireless networks*: all nodes are mobile, use wireless communications to exchange information and no specialized device exists. Hence, all nodes must collaborate to structure the network without the help of wired or wireless routers. Each node is both a client and a router. *Hybrid Networks* are such ad hoc networks connected to the Internet via a dedicated device: the *Access Point* (AP). Hybrid networks could be very useful to improve classical wireless networks: the radio coverage area is extended, the deployment is easier and the cost is reduced. Major applications for hybrid networks could be military communications or wireless networks for universities. The main objective in these networks is to route efficiently information among the nodes and the Internet. The key challenge is to structure the nodes in order to create an efficient collaboration to manage the network and to route packets. Routing protocols are mainly separated in *proactive* and *reactive* protocols. The proactive approach maintains continuously routes toward each node in the network: the delays are minimal but the overhead is important. The reactive approach discovers routes *on demand*: only useful routes are known, reducing the overhead. However, both approaches must use floodings causing the *broadcast storm problem* [1]: broadcasts are redundant and unreliable. Finally, such protocols don't take into account the natural heterogeneity of MANet.

We propose to create a new efficient routing solution for MANet. In [2], we have proposed a self-organization based on the construction and maintenance of a virtual dynamic topology constituted by both a backbone and clusters. This backbone consists in electing strongest nodes to act as routers and network managers, the root of the backbone being an AP. The backbone forms a connected structure and optimizes floodings: only backbone members forward the control packets, reducing the global load, and allowing other nodes to spare their energy. Moreover, we propose the creation of clusters on this backbone. Some of the strongest backbone members are elected as clusterheads, and form connected services areas with their own clients. We cut of the network in homogeneous zones. This solution is totally parameterizable: the maximum distance from a node to the backbone (k$_{cds}$) and the cluster radius (k$_{cluster}$) are both parameters. An extended solution of this virtual structure [3] provides more robustness and reliability according to multipath topology and multiple AP.

In this study, we propose to use this virtual topology in order to provide a new routing protocol which combines the assets of both proactive and reactive approaches: each node knows proactively a route toward its k$_{cds}$-neighbors, and toward each member of its cluster; other routes are discovered on demand. The latency is reduced for short distances, and the global overhead is optimized. Moreover, this proposition is hierarchical to reflect the network heterogeneity.

Next, we will expose related works about routing solutions in ad hoc and hybrid networks. Section III describes our routing solution. Section IV gives simulation results about the overheads, delays and delivery ratios. Finally, section V gives conclusions and provides perspectives.

II. Related Works

A. Flat routing

There exist several routing solutions in MANet. In the proactive approach, a node knows a route toward each node. If it knows the whole topology, it can compute dynamically optimal routes. Such a solution could be interesting when each node communicates with several destinations changing along the time. However, this knowledge has a cost: the overhead for topology packets is important. OLSR [4] tries to minimize the amount of control packets in selecting privileged nodes to forward control packets; nevertheless the overhead remains important. In a reactive solution, a node doesn't know initially any route. It initiates a route discovering only when it must send a data packet. In AODV [5], the source floods a `Route Request` to discover a route to the destination. The destination and each node knowing a route are allowed to reply with a `Route Reply`, forwarded in unicast toward the source. Hence, this approach doesn't create overhead for useless information. However, the latency for route discovering can be high, and the floodings can create an important overhead and problems of unreliability.

B. Hybrid routing

In ZRP [6], each node maintains the topology knowledge of its zone and proceeds in a reactive way for inter-zone routing. The node's zone is the set of its p-neighbors: the zone being different for each node. When S initiates a route discovering, it will send a `Route Request` to the nodes which are border of its zone, i.e. the nodes exactly p hops far. These border nodes forward the request until it reaches the destination. However, the zones are overlapping: there exists in consequence much redundancy in this flooding, potentially worse than blind flooding (if a node forwards several times the same packet for different but redundant or already covered border nodes). Some improvements trying to limit this flooding are presented in [7]. However, when the density increases, the number of border nodes increases, having an important impact on overhead. [7] proposes to select a limited set of border nodes for each `Route Request`, but according to a knowledge of the 2p+1-neighborhood, creating also an important overhead. Finally, the zones being not defined hierarchically, ZRP doesn't reflect the network heterogeneity. A more sophisticated structure could be perhaps more efficient.

C. Hierarchical routing

Hierarchy in MANet is also studied. In CBRP [8], clusters are constructed: each client must be neighbor of its clusterhead. Periodically, each node sends `hello` packets with the list of 1-neighbors and the list of clusters directly adjacent for which they are gateways. Hence, each node knows the exact 2-topology and each clusterhead knows the identity of all adjacent clusters. The set of clusterheads and gateways is connected.

When a node wants to send a data packet, its clusterhead acts as proxy for the route discovering: the clusterhead forwards a `Route Request` in unicast to its neighbors not yet covered and, through the gateways, toward adjacent clusters. The destination can send a `Route Reply`, along the inverse route. Each intermediary clusterhead will try to compute locally a route optimization according to its local topology knowledge before forwarding the `Route Reply`. However, the route being a list of intermediary nodes (and not of clusters), when a node moves, even inside its cluster, the route is repaired if possible, else it is broken. Moreover, the `Route Request` flooding could be suboptimal since the topology constituted by the clusterheads and the gateways is redundant and forms many loops. A more efficient backbone could be interesting.

III. Proposition

Our proposition focus on the problem of routing in MANet. Our solution must take into account the key challenges of MANet: network dynamicity because of nodes mobility, network heterogeneity, low radio bandwidth and the scalability according to the number of nodes and the amount of traffic. Our goal is to organize the network to set up an efficient routing protocol. Thus, our solution is based on a self-organized topology. A backbone helps us to optimize floodings and to create a hierarchy among the nodes whereas clusters create stable routes inside the network. As a routing protocol, our proposition provides routes. For us a route is not a list of intermediary nodes but a list of clusters. Clusters topology is more stable than nodes topology. Next, we introduce the virtual structure and then we will detail our routing solution separated into intra-cluster and inter-cluster routing.

A. Virtual Structure

We use the virtual structure detailed in [2]. This solution constructs first a backbone in selecting the strongest nodes to act as backbone members, the *dominators*. These nodes are selected according to a stability weight representing their aptitude to act as network managers [1]. This backbone construction is based on the k_{cds}-neighborhood knowledge. Some nodes are elected as *dominators* to form a connected structure where each *normal* node (or *dominatee*) is at most k_{cds} hops far from its *dominator*. This backbone represents a k_{cds}-CDS [9]. The backbone constitutes a tree of dominators where the leaves are the dominatees. Each dominator maintains the identity of its parent (except the AP) and the identity of the dominators for which it is a parent: they constitute its children. Based on this backbone, $k_{cluster}$-clusters are constructed: each node must be at most $k_{cluster}$ hops far from its clusterhead. To reduce the overhead, the *dominatees* don't take part to the election: they choose the clusterhead of their dominator. The *dominators* proceed to an election to select the most suitable clusterheads.

MANet are volatile environments. Hence, we have proposed a maintenance protocol to maintain the efficiency of the virtual structure. To maintain the backbone connectivity, `AP-hello` are periodically sent by the AP. The `AP-hello` are only forwarded by the dominators, limiting the overhead. `AP-hello` may include later the parameters for the Internet Access. If a *dominator* missed several `AP-hello`, it must try a reconnection to the backbone. In `hello` packets, each dominator provides the identity of its clusterhead and its distance. With this information, each dominator can choose a clusterhead and verify its validity. Procedures are proposed to merge clusters and to reconnect the backbone and clusters.

B. Intra-Cluster Routing

We assume that the nodes in both ad hoc and hybrid networks will mainly communicate with neighbors, and sometimes with farther nodes. For example, on a campus, a person will communicate mainly with colleagues, and will sometime request information to remote servers or nodes. Hence, it could be interesting to maintain a proactive knowledge of the local zone. Moreover, there exist many changes in the physical topology: the clusters create a high level simplified view where less changes occur. A node can move inside its cluster, it

[1] [2] provides a full information about weight and topology behavior.

occurs no change in the cluster topology. A routing solution on this topology optimizes the stability of routes. Each node must already know its k$_{cds}$-neighbors. To enlarge this zone, each node maintains the topology of its cluster. In parameterizing k$_{cluster}$, we can regulate the induced overhead.

k$_{cds}$=2 optimizes the backbone robustness: it is stable and the proportion of *dominators* is small (about 30% with a degree of 10 neighbors). Because of the radio links properties, the nodes must distinguish bidirectional and unidirectional links: the `hello` packets must contain the list of 1-neighbors. Thus, to know the 2-neighborhood, each node sends one single `hello` packet. Each node forwards this packet if it comes from a bidirectional link, and if the source has the same clusterhead, the initial *TTL* of the packet being set to $2 \cdot k_{cluster}$. Since the `hello` packets contain the 1-neighbors of the source, each node can reconstruct the exact topology of its cluster and its k$_{cds}$-neighborhood, and can execute for example the Dijkstra algorithm to find optimal local routes. Routing tables being coherent in a cluster, the route could be coherently computed hop by hop.

C. Inter-Cluster Routing

The inter-cluster routing solution is reactive: a node discovers a route *on demand* when it must send a data packet to an unknown destination. This route discovering is optimized thanks to the virtual backbone: the `Route Requests` reach a limited number of nodes. More, routes use the virtual topology constituted by the clusters: routes are more stable, and dynamically adaptable.

1) Cluster Topology Discovering: The route contains the addresses of the intermediary clusters toward the destination. Hence, each node must know routes inside its cluster and a route for each adjacent cluster. The local routes are known with the proactive intra-cluster routing protocol. However, we must propose a mechanism to discover the cluster adjacency and to detect gateways toward other clusters. We choose to integrate this process to the neighborhood discovering. All the clusterhead identities listened by a node are integrated in its `hello` packets: such a node can play the role of gateway. We assume the increase in length of the packet is acceptable. In radio communications, the number of packets has a more important impact on the performances than the packet length.

2) Routes Discovering: When a source S wants to send a data packet to D, one of the following possibilities can occur:

1) D is at most k$_{cds}$ hops far, or D and S are in the same cluster: D is in the neighborhood table of S. So, S has already a route toward D, it executes the intra-cluster routing as described above;
2) D is in the routing table of S: S has a route of clusters for D, it executes the inter-cluster routing to reach D;
3) D is unknown: S initiates a route discovering.

To discover routes, `Route Requests` are used and their flooding are optimized thanks to the backbone. A dominatee lets its dominator act as proxy for the route discovering. The first dominator adds its clusterhead in the cluster list of the `Route Request`. Then, it initiates a backbone flooding. A dominator receiving such a packet must forward it if D is not in its neighborhood table and if the packet has never been seen before. Before relaying the `Route Request` in multicast, a dominator adds its clusterhead in the cluster route of the packet if the clusterhead of the previous dominator is different from its. If a dominator has D in its neighborhood-table, it acts as proxy for D and sends a `Route Reply`. If necessary, it adds in the cluster list its clusterhead and the clusterhead of D. Then, the `Route Reply` is sent to S with the unicast inter-cluster routing protocol. Thus, `Route Request` are only forwarded by dominators, sparing many useless transmissions whereas, in blind flooding (like in AODV), all the nodes must forward the `Route Requests`.

3) Routing: For `Route Replies` and `Data Packets`, the inter-cluster routing is processed. Before relaying a `Route Reply`, a node may add the route toward the source in its routing table. Hence, we reduce the amount of generated `Route Requests`. Each packet requiring the inter-cluster routing contains a route of clusters. If the final destination is in the neighborhood table, the packet is directly sent with the intra-cluster routing. Else, the relaying node N_1 searches, in the cluster route, the first known cluster C, the nearest of the destination:

1) A 1-neighbor, N_2, has the clusterhead C: the next hop is N_2;
2) A 1-neighbor, N_2, is gateway for C: the next hop is N_2;
3) A neighbor of the same cluster of N_1 is gateway for C: N_1 chooses the nearest gateway and executes the intra-cluster routing to join this gateway. N_1 and the next hop N_2 are in the same cluster, they have the same view of the cluster topology and its gateways. Hence, N_2 will take a routing decision coherent with N_1.

This procedure can't create loops in the routing: when a packet is forwarded, it is one hop nearer of the destination. It could be possible to have incoherences in neighborhood tables [10], due to convergence delays. To avoid loops, the packet is simply dropped if it was already seen. The route is dynamically computed: it could consequently be different from the cluster route present in the packet. If a node changes the route for a `Route Reply`, the packet is updated on the fly, so that the destination receives a valid route. For `Data Packets`, this update is not required since the source won't be prevented. Finally, the route length is not optimal, but we try to reduce it in choosing to forward the packet to the cluster nearest of the destination.

This inter-cluster routing is robust: we compute dynamically the route, along the way. If the cluster route remains valid, the `Data Packets` and `Route Replies` reach the destination even if many individual nodes move. Because the cluster topology is more stable than the physical topology, our solution is more stable according to the networks changes.

D. Route Repairs

We assume that the data delivery ratio is one of the major performance criterion to optimize in radio networks where packets losses are frequent. Hence, we propose to

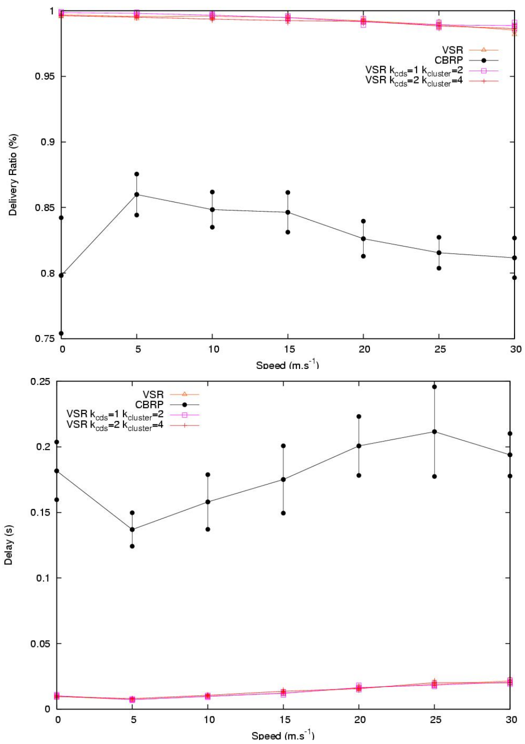

Fig. 1. End-to-end delay and delivery ratio according to mobility

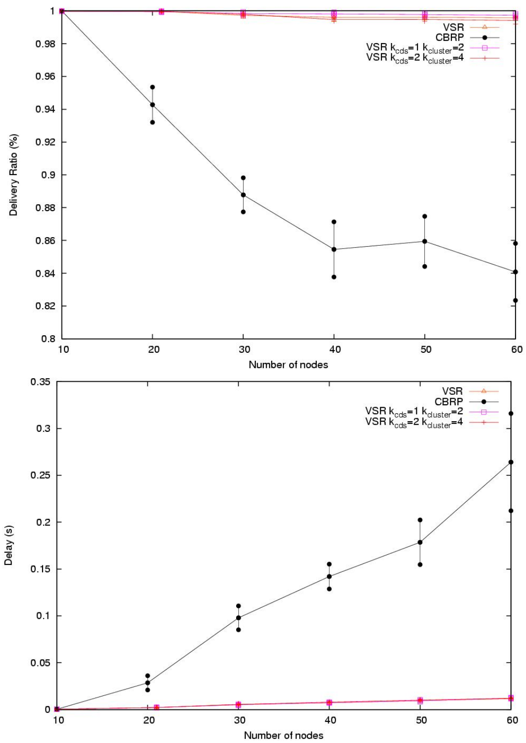

Fig. 2. Delay and delivery ratio according to the number of nodes

implement a simple mechanism of acknowledgments for the `Data Packets`. When a `Data Packet` is sent, if no acknowledgment was received after $timeout_{ack}$, the packet is retransmitted (with a maximum of nb_{retry}). If the node is in promiscuous mode, and the next hop is not the final destination, no explicit acknowledgment is required, it is sufficient to hear the next hop forwarding the packet. Only the final destination must send an active acknowledgment. This mechanism presents a very limited overhead, and has not an important impact on energy: the reception is not the principal energy consumption factor [11]. We propose, like CBRP, a mechanism of route repairs. This process is not frequent since the route is dynamically computed with local and fresh information. When a node tried nb_{retry} retransmissions through N, but no passive or active acknowledgment was received, it will initiate a route repair: it executes the same routing procedure as normally, but forbidding N as the next hop. This route repair limits the impact of neighborhood table convergence delays, improving the delivery ratio but disfavoring naturally the end-to-end delay of `Data Packets`.

IV. Performance Evaluation

Our proposition is simulated using Opnet Modeler 8.1, with the implementation of the 802.11b radio model provided by Opnet with a radio range of 300m. Nodes trajectories are simulated with the random waypoint mobility model on a square area, the surface depending on the number of nodes. The default parameters are 40 nodes, a speed of 5 m.s^{-1}, a degree of 10 and 3 simultaneous data flows. A data flow is constituted by 8 packets of 128 octets, sent every 0.25 s. The inter-flow time follows an exponential distribution with an average value of 2s. The virtual structure is parameterized according to k$_{cds}$=2 and $k_{cluster}$=3. For comparison, we also simulate VSR with k$_{cds}$=1/$k_{cluster}$=2 and k$_{cds}$=2/$k_{cluster}$=4. For a more generic purpose, we suppose that the promiscuous mode is not available, so an unicast acknowledgment must be sent by each node receiving a `Data Packet`. We set $timeout_{ack}$=0.2s, the end-to-end delay will hence reflect the robustness of the routes. We set $timeout_{RREQ}$=0.5s, as in [8]. We have a maximum of 1 route repair per node. To evaluate our solution (VSR), we investigate the behavior of our routing protocol according to the mobility, the number of nodes and the data load. The performance evaluation parameters are mainly (1) the end-to-end delay between the source and the destination (2) the data delivery ratio (3) the route length. We compare VSR to CBRP since it is the most efficient cluster routing protocol to our knowledge.

a) Mobility: We study the impact of mobility on the performances of both VSR and CBRP. The delivery ratio of VSR remains almost constant when mobility increases: the routes are robust, the local knowledge allows to update routes on the fly without delays. CBRP has a smaller delivery ratio and is more impacted by mobility. Because the mechanism of retransmissions, CBRP must wait a timeout before retransmitting. Hence, the delay increases when the network dynamicity increases. Oppositely, VSR combining proactive and reactive presents very good delays (about 10 times lower than CBRP). Then, we investigate the robustness of the protocol. First, at 5m.s^{-1}, CBRP has on average 0.30 retransmissions per packet per hop, and VSR 0.06. VSR creating stable routes needs less retransmissions. Secondly, VSR has an average route length of 2.13 hops whereas CBRP has a route length of 2.55 hops. The

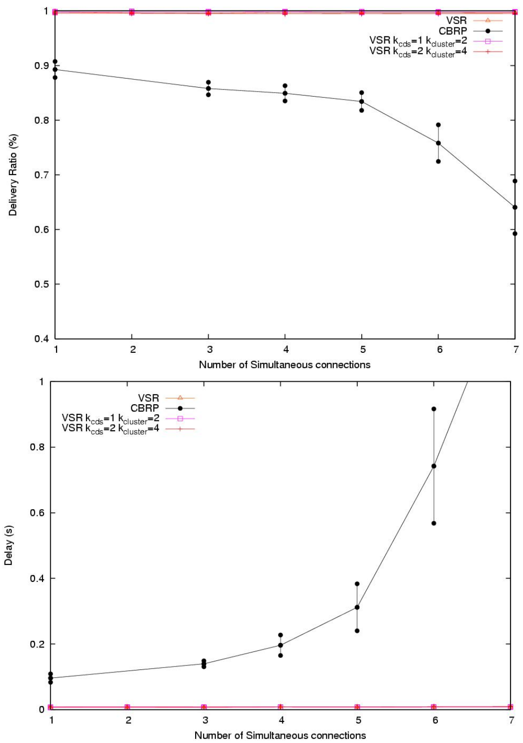

Fig. 3. Delivery ratio and delay according to the data load

virtual structure combined to a dynamical route computation allows to maintain almost optimal routes. Moreover, CBRP increases the route length when it initiates a local repair.

b) Number of nodes: We study the behavior of VSR when the number of nodes in the network increases, maintaining a constant degree. When the number of nodes increases, the network radius increases and performances decrease. However, VSR seems very scalable according to the network cardinality compared to CBRP. With CBRP, the impact of the `Route Request` floodings is more important: more nodes need to be flooded, and the redundancy in the flooding creates a broadcast storm. This creates `Data Packets` retransmissions, and the performances fall, increasing the delay.

c) Load: VSR is scalable according to the data load. The delivery ratio and the delay are constant when the number of simultaneous connections increases. The performances of CBRP fall when the number of communications exceed 4. The `Route Request` floodings create collisions, many `Route Reply` are lost, creating more `Route Requests`, increasing the delays. The `Data Packets` experience collisions, increasing the delays and reducing the delivery ratios.

d) Overhead: VSR creates an important overhead due to the virtual structure and the local proactive routing, but the `Route Request` and `Route Reply` have a very low impact. `hellos` could be lower if we implement a proactive protocol like OLSR. Oppositely, CBRP has an important overhead for route discovering, increasing with the data load and the number of nodes. The advantage of VSR over CBRP is higher when the network load increases. Overheads in packets sent per second per node are with 5 simultaneous communications:

	hellos	structure	RREQ	RREP	acks	Total
VSR	3.59	0.20	0.011	0.006	1.00	4.8
CBRP	0.52	0	2.95	0.36	2.11	5.94

V. Conclusion

We propose a routing solution organizing the hybrid network, using a virtual backbone to collect control traffic, and clusters to create services areas, particularly efficient for hierarchical routing. Inside clusters and in k$_{cds}$-neighborhood, the routing solution is proactive, optimizing latency. Inter-cluster routing is reactive, to limit overheads of topology packets exchanges. Moreover, we optimize the route discovering process in using a virtual backbone, avoiding other nodes to receive and forward these control packets. Furthermore, routing is more stable with the hierarchical organization of the network, routes using the cluster topology rather than the volatile physical radio topology. Finally, routes are computed dynamically for both intra and inter-cluster routing. Routes are computed on the fly, using the local proactive knowledge, using the freshest information. These key properties explain the good performances on the delay and the delivery ratio of our solution. VSR creates a trade off between proactive and reactive protocols, combining their assets. Next step of this study will be the implementation of new functions like address assignment for auto-configuration, a sleeping mode for backbone clients, QoS routing, and a multicast routing solution using the backbone to distribute data and clusterheads to manage the inscriptions. In the future, our virtual dynamic structure will constitute a framework for easier services deployment.

References

[1] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *MobiCom*. Seattle, USA: ACM, August 1999.

[2] F. Theoleyre and F. Valois, “A virtual structure for mobility management in hybrid networks,” in *Wireless Communications and Networking Conference (WCNC)*, ser. 1, vol. 5. Atlanta, USA: IEEE, March 2004, pp. 1035–1040.

[3] ——, “Robustness and reliability for virtual topologies in wireless multihops access networks,” in *Mediterranean Ad Hoc Networking Workshop*, Bodrum, Turkey, June 2004, pp. 81–92.

[4] T. Clausen and P. Jacquet, “Optimized link state routing protocol (olsr),” IETF, RFC 3626, October 2003.

[5] C. E. Perkins, E. M. Belding Royer, and S. R. Das, “Ad hoc on-demand distance vector (aodv) routing,” IETF, RFC 3561, July 2003.

[6] M. R. Pearlman and Z. J. Haas, “Determining the optimal configuration of the zone routing protocol,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395–1414, June 1999.

[7] Z. J. Haas and M. R. Pearlman, “The performance of query control schemes for the zone routing protocol,” in *SIGCOMM’98*. Vancouver, Canada: ACM, September 1998.

[8] M. Jiang, J. Li, and Y. C. Tay, “Cluster based routing protocol (cbrp),” draft-ietf-manet-cbrp-spec-01.txt, IETF, Internet draft version 01, July 1999.

[9] S. Butenko, X. Cheng, D.-Z. Du, and P. M. Pardalos, “On the construction of virtual backbone for ad hoc wireless networks,” in *Cooperative Control: Models, Applications and Algorithms*, ser. Cooperative Systems. Kluwer Academic Publishers, January 2003, vol. 1, ch. 3, pp. 43–54.

[10] J. Wu and W. Lou, “Extended multipoint relays to determine connected dominating sets in manets,” in *SECON*. Santa Clara, USA: IEEE, October 2004.

[11] L. Feeney and M. Nilson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in *INFOCOM*. Anchorage, USA: IEEE, April 2001.