WIANI: Wireless Infrastructure and Ad-Hoc Network Integration

Jiancong Chen[†♮]
[♮]Department of Electrical Engineering
Princeton University
USA
Email: kentchen@princeton.edu

Shejie Li[♯] S.-H. Gary Chan[♯]
[♯]Department of Computer Science
HKUST
Hong Kong
{gchan, lisj}@cs.ust.hk

Jingyi He[♭]
[♭]ASTRI
Shatin, New Territories
Hong Kong
jyhe@astri.org

Abstract— **Wireless networks have been widely deployed in recent years to provide high-speed Internet access to mobile users. In traditional IEEE 802.11 wireless LANs, all users directly connect to an access point (AP) and all packets are forwarded by the AP. As a result, the coverage and capacity of the network is limited. If ad hoc mode is adopted in both the AP and mobile nodes, the one hop connections from AP can be extended to multiple hops. Such arhitecture, termed WIANI (Wireless Infrastructure and Ad-Hoc Network Integration) in this paper, is able to extend the network coverage beyond the coverage of APs. Furthermore, the users may take advantage of the ad hoc connections to forward local data and hence alleviating the traffic load through the AP and increasing the network capacity.**

In this paper, we propose a dynamic load-balancing protocol for WIANI in which all APs and nodes operate in ad hoc mode. Our protocol consists of two parts: the load-balancing zone forming and weighted $x-$hop routing algorithms. Using simulation, we show that our protocol improves system throughput and reduce packet delivery delay.

Index Terms— **Multi-hop wireless LAN, load-balancing routing, infrastructure network, ad hoc network, IEEE 802.11**

I. Introduction

THE penetration of IEEE 802.11-based wireless networks (the so-called "WiFi") into our homes, offices, and cafes has provided users ubiquitous broadband Internet access [1]. These wireless networks make use of "infrastructure" mode, where an access point (AP) offers wireless users in its coverage access to the Internet. Although the 802.11 networks offer high data rate, the transmission range is limited by the one-hop connections in the infrastructure networks (usually less than $100m$ indoor or $250m$ outdoor). Employing ad hoc mode in 802.11 in both APs and mobile nodes can extend traditional wireless local area networks (WLAN) to multiple hops, thus increasing the coverage and reducing the needs of additional infrastructures [2]–[5]. Such architecture is called WIANI (Wireless Infrastructure and Ad-hoc Network Integration) in this paper.

Different from traditional WLAN, nodes in WIANI may connect to an AP through multiple hops or communicate to each other without going through an AP. In Fig. 1 we illustrate the WIANI architecture. The APs are connected by a wired network (such as an Ethernet). The traffic among APs always goes through the wired network. In wireless channel, the APs and nodes communicate using ad hoc mode. WIANI differs from the mobile ad hoc networks, in which there are no APs providing services to the nodes.

There are a number of benefits of the WIANI architecture. We briefly discuss them here.

- *Extended wireless coverage and reduced deployment cost:* In the traditional WLAN, a node must be located within the coverage area of some AP to receive wireless services. WIANI extends the coverage area by ad hoc connections. Such a solution is particularly useful in handling temporary needs. If a transient user population moves into an area with no AP coverage, WIANI can be used to provide immediate wireless services. Obviously deploying more APs to cover the popular area may be a long-term solution. However, covering a large area by single-hop WLAN may require a lot of APs, among which some may not be useful most of the time. Therefore, multi-hop solution is more appropriate to handle transience. By deploying WIANI, fewer APs are required to provide wireless services in the same area. Hence the deployment cost can be reduced.
- *Performance enhancement:* First, with ad hoc support, a node has more opportunities for path selection to route its data through the network. It may use a path through an AP or just use ad hoc nodes to forward its data if the AP is congested. Second, a mobile node with low battery power may choose closer neighbor to forward packets to an AP rather than send directly through a long distance. Hence the node significantly reduces the energy requirements for communication. Meanwhile, communicating with closer nodes also increases the received signal quality and hence allows the 802.11 protocol to operate at the higher data rate.
- *Load-balancing among APs:* In traditional WLAN, the uneven distribution of users may leads to congestion of some APs. While in WIANI, since a node can connect to an AP in a multi-hop way, the traffic load can be better balanced.

To realize such a multi-hop WLAN, there are a number of challenges and issues, such as channel access and assignment, routing and fairness issues, etc. One of the most important issues to be considered is the routing protocol for the network

†Correspondence author: Jiancong Chen

This work has been supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology of the University Grant Council (AoE/E-01/99), and by HKUST post-doctoral fellowship matching grant and high impact area grant (HIA03/04.EG04), Hong Kong.

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

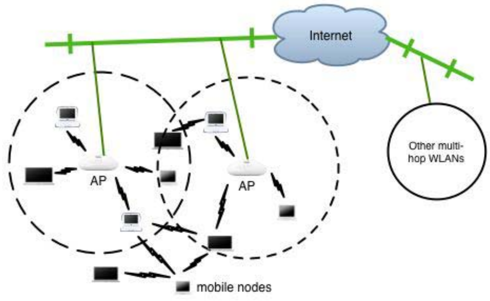


Fig. 1. A unified multi-hop WLAN architecture.

since the connections to an AP are extended to multiple hops and the local ad hoc routing is enabled. Plenty of routing protocols have been proposed for mobile ad hoc networks, such as DSDV, AODV, DSR and so on [6]–[9]. However, these protocols do not take into consideration the advantages of using the APs in the network. Therefore, we propose in this paper a dynamic load-balancing routing protocol for WIANI.

Our proposed protocol consists of two parts: the load-balancing zone forming and weighted $x-$hop routing algorithms. A *zone* for an AP is defined by the nodes that have default route to it. In our zone forming algorithm, the APs broadcast beacons periodically to let the nodes join the zones. In the weighted $x-$hop routing, a dynamic weight is assigned to each node and AP according to its traffic condition. A path with the lowest weight is selected as the best route. The number x limits the traveling hops of a route request (RREQ) and hence reduces the communication overhead.

Our main contribution in this paper is the presentation and performance study of a new WIANI architecture. We simulate WIANI in NS2 to evaluate the performance of our routing algorithm in terms of delay, throughput and served users, etc. Our results show that our load-balancing scheme achieves better system performance comparing to AODV and $k-$hop routing protocol [4].

There has been much work on wireless LAN and ad hoc networks. However, most of them treats the two types of networks separately without considering a unified architecture [10]. Wu *et al.* propose a system named iCAR which integrates cellular system with ad hoc relays [11], [12]. Such integration is also studied in another protocol called Heterogeneous Wireless Network [13]. In them, the cellular system uses a frequency band different from that of the ad hoc networks. It requires two radios in each device, leading to a more expensive solution. Our study differs primarily in that we are considering a unified wireless system in which the bandwidth may be *shared* by all users, and hence the performance improvement with ad hoc relay is not obvious. Lin *et al.* propose a bridging protocol to enable multi-hop connections from a node to an AP [14]. The case for a Multi-hop WLAN is also presented in [5], which modifies the MAC layer to enable multi-hop wireless path establishment. We proposed a mixed-mode WLAN framework to enhance the performance of WLAN and address the hot-spot congestion problem [2], [3]. This work differs by considering a more general system which involves both ad hoc routing and routing to zone APs in network layer.

List et al. propose a routing protocol similar to AODV for $k-$hop networks, in which wireless hops in ad hoc connections are upper bounded by k [4]. Our work differs from it in the following ways: 1) We have a zone forming protocol to maintain topology information. 2) We assign dynamic weights to nodes according to traffic conditions in the nodes for making routing decision. 3) Although we also use TTL to limit the traveling hops of route requests, the number may be different among nodes while the $k-$hop networks predefine the same limit for all nodes.

Load-balancing routing protocols have been studied in different networks including wired (ATM and optical) and wireless (hot-spot, ad hoc and sensor) networks [15]–[17]. However, the previous schemes focus on different systems with different characteristics. Hence it is not straight forward to apply them in other environments.

The rest of this paper is organized as follows. We present in Sect. II the framework of load-balancing zone forming. Weighted $x-$hop routing is presented in Sect. III. Simulation results are presented in Sect. IV. We conclude in Sect. V.

II. Load-Balancing Zone Forming for WIANI

In this section, we describe the detailed protocol of load-balancing zone forming. A *zone* is defined as the set of nodes virtually associated to the same AP through one hop or multiple hops. Here, the association to a AP means the node maintains a default route to the AP. The APs send beacons periodically to gather and maintain the zone information. On receiving a beacon, a node makes decisions on which zone (AP) to join and then responds. In the following, we first describe the unified multi-hop wireless LAN architecture. Then we discuss the detailed load-balancing zone forming algorithm.

A. The Architecture of WIANI

In WIANI, there are a number of APs and nodes as shown in Fig. 1. The APs are connected to the wired network (such as the Internet). In the wireless channel, all the APs and nodes use a unified ad hoc mode to communicate with each other.

In such a network, the APs provide Internet services and inter-network connectivities to the mobile users. The local communications among nodes may be routed either along the ad hoc nodes or through the APs. Therefore, the APs should know that which nodes they need to serve and each node need to maintain a route to an AP. Such routes may change dynamically as nodes move around. To accomplish such a purpose, we deploy an AP-driven beaconing mechanism to form a zone for each AP. A joining cost is assigned to the beacon message. A node receiving a beacon makes joining decision based on the cost and sets default route to the AP with the lowest cost by sending a reply. The AP discovers the nodes of its zone from such replies.

B. Load-Balancing Zone Forming

To balance the traffic load among the APs, we propose a load-balancing zone forming protocol based on AP-driven beaconing. The beaconing process works as follows.

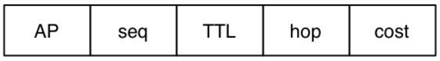


Fig. 2. The BEACON message format.

TABLE I
THE AP_LIST MAINTAINED IN A NODE.

AP_ID	seq	hops	next hop	cost
0	21	2	15	15.93
1	5	4	11	23.11
···	···	···	···	···

Each AP periodically transmits a *beacon* (BEACON) message. The format of the beacon message is shown in Fig. 2. The fields *AP* and *seq* are the unique AP ID and sequence number of a beacon message. Each AP increments its own beacon sequence number before it sends a beacon message. From the two firelds, a node knows whether the message is new or not. The *TTL* (Time-To-Live) field determines how many hops that a message is to be forwarded. The *hop* field records how many hops that a message has traveled. The initial *cost* of a beacon is assigned by the AP according to its traffic load and number of nodes in its zone.

Upon receiving a beacon message, a node processes it according to the flow chart shown in Fig. 3. If the beacon is from a new AP, which is not currently in the AP_List of the node, the node adds the AP to its AP_List. The AP_List of a node maintains the APs that a node can reach. The node only uses the default zone AP to forward data if there is no routing error. If there are errors when a node sends data to its zone AP (such as the links are broken due to mobility), it may use one of the other APs in the list. An example of the AP_List is shown in Table I. The meaning of the fields are as follows:

- *AP_ID* — The unique ID of an AP;
- *seq* — The sequence number of the last beacon from the AP;
- *hops* — The number of hops that a node is away from the AP;
- *next hop* — The next hop to forward packets to the AP;
- *cost* — The cost of joining the zone of the AP.

After the new AP is added, the node computes the cost to join the new zone (of the new AP). When the cost is lower than the current cost by a given threshold T, the node flips a coin to determine whether it will switch to the new zone or not. In this way, the node does not switch its zone whenever the new cost is lower, thus avoiding the instability (oscillation) effects in zone forming.

If the node switches to the new zone, the node sends a *Route Reply (RREP)* message to the AP and updates its default route to the new AP. The RREP message is similar to the RREP in AODV. It contains the routing information to the node. The intermediate nodes who forward the RREP message also process the packet and update their route to the RREP source. After that, if the beacon's *TTL* field is larger than 0, the beacon is again forwarded by the node with an updated *hop* and *cost*. If the *TTL* is zero, the beacon will be discarded.

If the beacon is from the current zone AP, the node checks the sequence number to see if it is a new beacon. If it is new, the node updates its cost to the AP accordingly and replies a RREP. Then the node forwards the beacon with updated *hop* and *cost* if the *TTL* is larger than 0. If the beacon is with an old sequence number, it is discarded.

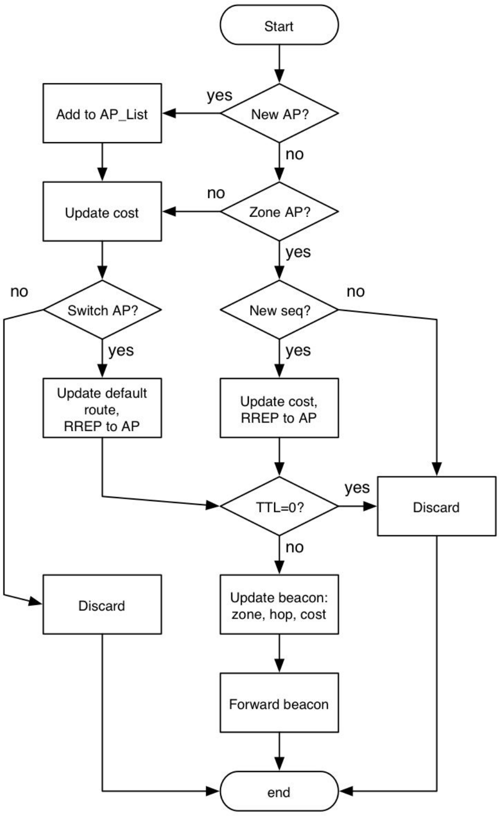


Fig. 3. The flowchart of beacon process in a node.

Upon receiving the RREP message, the zone AP updates the zone topology by maintaining a node list consisting of the nodes in the zone. When a node joins a zone (by sending a RREP to the AP), the AP adds an entry for it in the node list. A default route to each node is updated by the RREP of each beacon. If the AP does not receive the RREP from a node after a certain number of beacons, the node will be removed from the list (i.e., from the zone).

As a node determines which zone it joins according to the cost in each beacon, such cost needs to be properly assigned to achieve load balancing among the zones (APs). The AP assigns an initial cost c_0 to a beacon according to the traffic load (in the past beacon interval) across it and the number of nodes in its zone. When a node i receives a beacon, it updates the cost to the AP in its AP_List. If the node belongs to the zone or it determines to switch to the zone, it adds its own cost c_i to the cost in the beacon and then forwards the beacon. Therefore, the total cost to join a zone is the cost sum of the AP and intermediate nodes, i.e.,

$$C_{joining} = \sum c_i, i \in beacon\ forwarded\ path. \quad (1)$$

III. DYNAMIC WEIGHTED $x-$HOP ROUTING

The support of ad hoc routing enriches the path selections in the network and reduces the potential traffic load through the AP. Hence the multi-hop architecture achieves better load-balancing than traditional WLAN. However, how to select a proper (or optimal) route becomes a more significant issue in a multi-hop WLAN. To address the routing issue, we propose a weighted $x-$hop routing similar to AODV based on our load-balancing zone forming. We use the topology information obtained in zone-forming to reduce the routing overhead as compare to ad hoc routing. In the following, we discuss three essential separate phases in the routing protocol.

A. Beaconing

Our routing protocol attempts to trade off the routing requests by the periodical beaconing. Since the beaconing helps the APs and nodes maintain default routes to others in the network, a source may use the default routes to forward data to the desired destination. The requirement of route requests is hence reduced.

As we describe in Sect. II-B, the beacons not only help a node to select and join a zone, but also allow it to update its default route to its zone AP. To respond the beacon from its zone AP, A node sends a reply message in the form of an AODV-style RREP. It specifies how long the route is valid and is unicast back to the AP. The intermediate nodes also process the message and cache a route with a certain weight to the source node. The weight is the total cost of the intermediate nodes in the cache path. Hence, the AP also has the weights to every node in its zone. In this way, the beaconing mechanism helps the APs and notes to maintain weighted routing information. Depending on different applications, such weight metric can be power, stability or residual bandwidth in a node. We use residual bandwidth as our simulation metric.

B. Route Discovery

There are three types of route discovery in our protocol for local communications, outgoing or incoming traffic. In the first case, the source and destination are in the same network (e.g., a campus network). The source initials a Route Request (RREQ) flood that is limited to x hops, where x is the number of hops that the source is away from the zone AP. A node with cached route to the destination responds a RREP which consists of a weight to the destination. If the destination replies a RREP, the initial weight is set to zero. The zone AP also replies a RREP to the source with an initial weight from the AP to the destination. When the RREP travels back to the source, the weight will be cumulated by the intermediate nodes. If the source receives multiple RREP, it selects the path with lowest weight to forward data.

If the destination is not in the same network, i.e., the traffic is outgoing, the node directly sends data to its zone AP. The AP will relay the traffic to the desired destination. An AP does not respond to the RREQ from a node which is not in its zone.

If an AP receives an RREQ with a destination in its zone, it sends a RREP with the weight information collected by the most recent beaconing. When an AP receives data orienting a node in its zone, it forwards the data using the route maintained by beaconing. If the route for a specific node is expired, the route recovery is performed by the AP.

TABLE II
SIMULATION SETTINGS.

Transmission Range	$250m$
Radio Propagation Model	Two-ray Ground
Simulation Duration	$120s$
Mobility Model	Random way-point
Pause Time	$0s$
Medium Access Protocol	IEEE 802.11 DCF
Link Bit-rate	2 Mb/s
Packet Size	1000 bytes

C. Route Recovery

Although we may use the same route recovery mechanism as in AODV, in our protocol, only APs and source nodes perform route recovery. When an AP receives a packet orienting a node with expired route, it triggers a beacon to refresh its zone topology and a route error (RERR) with the destination address to other APs in the network. Upon receiving a RERR message, if an AP has valid route to the destination, it responds a RREP to the AP sending the RERR. Otherwise, if an AP finds the destination in the next beacon, it sends a RREP to the original AP.

If a source using an ad hoc connection without going through an AP encounters a route error, it uses the path to its zone AP as a backup. On receiving such packets, the AP will forward them to the destination by the default route.

IV. Illustrative Numerical Results

In this section, we present the simulation results of our load-balancing routing (LBR) protocol. We implement our protocol using NS 2 (version 2.27) with CMU wireless extension. The common parameters for all the simulation runs are listed in Table II, which are similar to those in [4]. We compare our results to AODV and the $k-$hop routing protocol (KRP) proposed in [4].

For all experiments, we choose 100 mobile nodes to roam around a rectangular area of $2400m \times 600m$ at four different speeds of 1, 5, 10 and 20 m/s. Four APs are placed $600m$ apart and $300m$ from the top and bottom of the simulation area. Each data point in the results are the average of 5 runs of different traffic files. The generated traffic is comprised of TCP or UDP (constant bit rate of 50 kb/s) communications between mobile sources and sinks. We use the random way-point mobility model with a distribution of $15 : 35 : 35 : 15$ for each $600m \times 600m$ squared areas (for both original positions and motion destinations of the mobile nodes). The mobile nodes move in a velocity from 1 to 20 m/s. The beacon interval in all APs is set to 3 seconds. Our initial set of tests involves a network with 30 active connections (10 inner connections, 10 outgoing ones and 10 incoming ones).

In Fig. 4, we show the distribution of the number of nodes served by the APs in a traditional single-hop wireless LAN versus in a multi-hop wireless LAN with load-balancing zone forming. Clearly, not only the number of nodes served is significantly improved, but the distribution of nodes served by different APs are more uniform. Therefore, Our zone forming scheme achieve substantially better load-balancing.

In the following, we compare our results to AODV and the $k-$hop routing protocol (KRP) proposed in [4] in term of: 1)

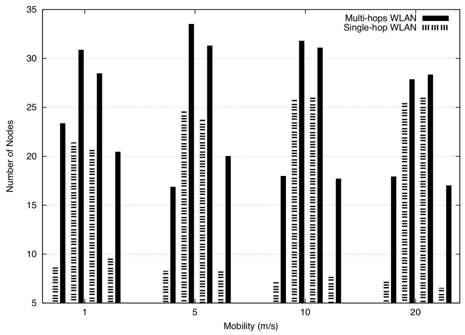
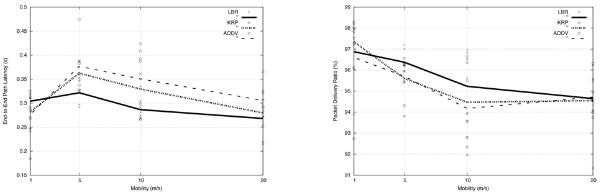


Fig. 4. The number of nodes served comparing to a single-hop WLAN.



(a) End-to-end path latency. (b) Average packet deliver rate.

Fig. 5. Performance comparison among LBR, KRP and AODV.

end-to-end path latency, 2) packet delivery ratio (PDR), and 3) network throughput.

We plot in Fig. 5 (a) the average end-to-end path latency of the active connections versus different mobility velocity. In most cases, LBR achieves shorter end-to-end delay comparing to AODV and KRP.

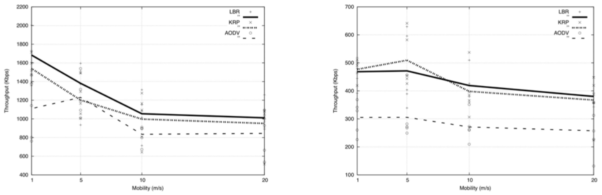
In Fig. 5 (b), we plot the packet delivery ratio defined by the percentage of packets that reach their destination over the total packets generated. Clearly, as mobility velocity increases, the PDR of the three schemes decreases accordingly because link errors are more likely to happen in higher mobility environments. However, our LBR have higher PDR comparing to the other two schemes.

Fig. 6 shows the network throughput using both TCP and UDP traffic flows. In both cases, the network throughput decreases with respect to mobility velocity due to the same reason as in PDR. Our LBR achieves consistently higher throughput when TCP flows are used. When UDP flows are used, LBR and KRP have similar throughput which is higher than AODV.

V. Conclusions

In this paper, we present a dynamic weighted $x-$hop routing algorithm with load-balancing zone-forming for a new unified multi-hop WIANI architecture. Using simulations, we show that with our load-balancing scheme, WIANI achieves better performance in terms of delay, PDR and network throughput comparing to AODV and KRP.

This is an ongoing research work. Our next step is to extend our routing protocol to support multicasting and Quality-of-Service (e.g. support applications with delay or reliability requirements) in such a unified multi-hop WIANI.



(a) TCP flows. (b) UDP flows.

Fig. 6. Network throughput comparison.

References

[1] IEEE Standards Board, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11-1997, Nov 1997.

[2] J.-C. Chen, S.-H. Chan, J.-Y. He, and S.-C. Liew, "Mixed-mode wlan: The integration of ad hoc mode with wireless lan infrastructure," in *Proceedings of IEEE Globecom'03*. San Francisco, CA: IEEE, December 2003. [Online]. Available: http://www.cs.ust.hk/faculty/gchan/papers/g03_M2.pdf

[3] J.-C. Chen, J.-Y. He, and S.-H. Chan, "A framework to relieve wireless hot-spot congestion by means of ad hoc connections," in *Proceedings of IEEE International Conference on Mobile and Wireless Communications Networks (MWCN)*. Singapore: IEEE, October 2003.

[4] W. List and N. Vaidya, "A routing protocol for k-hop networks," in *WCNC 2004*, March 2004. [Online]. Available: http://www.crhc.uiuc.edu/wireless/papers/khop-wcnc.pdf

[5] S. Lee, S. Banerjee, and B. Bhattacharjee, "The case for a multi-hop wireless local area network," in *Infocom 2004*. Hong Kong: IEEE, March 2004.

[6] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," in *IEEE Personal Communications*. IEEE, April 1999, pp. 46–55.

[7] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94*, London, UK, 1994, pp. 234–244.

[8] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90–100.

[9] D. Johnson, D. Maltz, and J. Broch, *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. Addison-Wesley, 2001, ch. 5, pp. 139–172.

[10] R. Draves, J. Padhye, and B. Zill, "Comparison of routing metrics for static multi-hop wireless networks," in *ACM SIGCOMM*. Portland, OR: ACM, August 2004.

[11] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: iCAR," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2105–2115, 2001.

[12] E. Y. amd O Tonguz, S. Mishra, H. Wu, and C. Qiao, "Efficient dynamic load balancing algorithms using iCAR systems: a generalized framework," in *Proceedings of Vehicular Technology Conference, VTC2002*. IEEE, Fall 2002, pp. 586–590.

[13] E. H.-K. Wu, Y.-Z. Huang, and J.-H. Chiang, "Dynamic adaptive routing for heterogeneous wireless network," in *Proceedings of Global Telecommunications Conference, 2001*. IEEE, 2001, pp. 3608–3612.

[14] Y.-D. Lin, Y.-C. Hsu, K.-W. Oyang, T.-C. Tsai, and D.-S. Yang, "Multihop wireless IEEE 802.11 LANs: a prototype implementation," in *Proceedings of IEEE International Conference on Communications, 1999*. IEEE, 1999, pp. 1568 –1572.

[15] L. Du, J. Bigham, and L. Cuthbert, "A bubble oscillation algorithm for distributed geographic load balancing in mobile networks," in *Infocom 2004*. Hong Kong: IEEE, March 2004.

[16] Y. Ganjali and A. Keshavarzian, "Load balancing in ad hoc networks: Single-path routing vs. multi-path routing," in *Infocom 2004*, IEEE. Hong Kong: IEEE, March 2004.

[17] S.-C. Huang and R.-H. Jan, "Energy-aware, load balanced routing schemes for sensor networks," in *International Conference on Parallel and Distributed Systems*. IEEE, July 2004, pp. 419 – 425.