Mixed-Mode WLAN: The Integration of Ad Hoc Mode with Wireless LAN Infrastructure

Jiancong Chen[†♮] S.-H. Gary Chan[♮] Jingyi He[♯]
[♮]Department of Computer Science
[♯]Department of Electrical & Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
Email: {kentchen, gchan}@cs.ust.hk, eehjy@ust.hk
Fax: +852 2358-1477

Soung-Chang Liew[♭]
[♭]Department of Information Engineering
The Chinese University of Hong Kong
Shatin, New Territories
Hong Kong

Abstract— **In the traditional IEEE 802.11 wireless LAN using infrastructure mode, all users share the same channel and all packets are forwarded by an access point (AP). As a result, as the number of users in the cell increases, the throughput for each user degrades substantially. If there are users communicating with each other within the cell (as in conferencing or file exchange applications), such throughput degradation could be relieved by making these users communicate through ad hoc connections without going through the AP. The advantages are multi-fold. First, the traffic load at the AP is reduced, hence relieving the contention. Second, ad hoc connections are single-hop, hence improving the channel efficiency. Moreover, ad hoc connections could use different channels, hence multiplying the system bandwidth. In this paper, we propose to integrate the infrastructure mode and the ad hoc mode in a wireless network so as to achieve these advantages. We present a framework for such mixed-mode wireless LAN (termed M^2-WLAN). In such a network, a node can dynamically switch between the infrastructure mode and the ad hoc mode according to the instruction of the AP, and hence the switching is transparent to the users. Using simulations, we show that M^2-WLAN can indeed improve system throughput substantially without user's manual configuration.**

Index Terms— **Mixed-mode wireless LAN, ad hoc network, infrastructure network, multi-channel, IEEE 802.11**

I. INTRODUCTION

IEEE 802.11-based wireless network (the so-called "WiFi") has become ubiquitous in our homes, offices, and even cafes to provide broadband Internet access to users [1]. Most wireless networks installed today make use of the "infrastructure" mode, where an access point (AP) provides an interface to wireless users in its coverage so that they can access the Internet. An IEEE 802.11b WLAN AP provides a bandwidth up to 11 Mbps, which is shared by all users in the network. When the number of users increases in the cell, the contention for the channel leads to a degradation in system throughput. In a wireless LAN, there is no direct connection between any two hosts, since the AP always serves as one of the intermediate nodes. For communications among the hosts within the same cell (e.g., the case of conferencing or accessing a file nearby), using the AP is not effective because not only the traffic load in the cell is increased, but also the throughput would not be so high as if they were connected by a direct wireless connection. A better way would be setting up ad hoc connections among the hosts. For example, for those who are holding a conference in a room, they can form a small wireless ad hoc network within their meeting room to share documents without going through the AP. Moreover, the small ad hoc network can use a different channel from that of the AP. Hence, there are less contention and collisions in the WLAN channel, increasing the system throughput for both ad hoc and WLAN users.

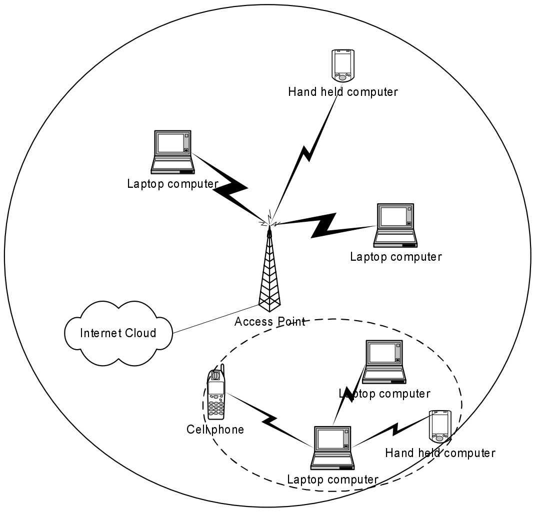


Fig. 1. A mixed-mode wireless LAN (M^2-WLAN) integrating ad hoc and infrastructure modes.

Indeed, IEEE 802.11 standard specifies an "ad hoc" mode, which allows users to spontaneously form a wireless network without the need of access point and cables. In ad hoc mode, users communicate directly with each other in a peer-to-peer manner. Certainly, for those users accessing data outside the cell, WLAN infrastructure mode is the only choice.

As we can see from above, if the infrastructure and ad hoc modes can be integrated to run simultaneously in a network the system performance can be improved. However, the ad hoc mode is not designed to work simultaneously with the infrastructure mode. In this paper, we consider the integration of these two modes to form a so-called "mixed-mode wireless LAN (M^2-WLAN)." We illustrate the system in Fig. 1. In the network, users can not only communicate with an AP, but also set up ad hoc connections under the administration of the AP. A host (or node) in an M^2-WLAN can dynamically switch between infrastructure mode and ad hoc mode depending on the traffic condition in the cell. Localized communication groups (e.g., the four users in Fig. 1), may set up a small ad hoc network to communicate with each other until they need to switch back to infrastructure mode.

There are several strengths of this system:

This work was supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council in Hong Kong (AoE/E-01/99), and by the Competitive Earmarked Research Grant of Research Grant Council in Hong Kong (HKUST6199/02E).

†Correspondence author: Jiancong Chen

1) Better bandwidth utilization — The bandwidth for WLAN infrastructure is better used for those outgoing traffic instead of intra-cell communication;
2) High throughput performance — As we will see in this paper, M^2-WLAN has much better performance (in terms of throughput) than a pure wireless LAN since it can utilize multiple wireless channels;
3) Transparency — The mode switching operations is transparent to users.

The M^2-WLAN should be useful for companies looking for fully wireless solutions for office area, where many users need to exchange data among themselves as well as to get access to the Internet. If they were to share the same AP using infrastructure mode, the performance of the network would be unsatisfactory. To improve the performance, multiple access points may be used to separate users onto non-overlapping channels to reduce access contention. Such an approach, however, is not cost-effective.

An important issue in the M^2-WLAN is the channel assignment for different ad hoc networks, as an inappropriate assignment may lead to collision and hence low performance. As for the nodes, the main issues are when to use infrastructure mode and ad hoc mode and how to switch from one mode to the other without affecting other nodes or ongoing communications. In this paper, we propose a framework to address these issues. In this framework, the AP manages the modes of the nodes in the network. Initially, all the nodes operate in the infrastructure mode. If the traffic load is high, the AP requests some local groups to switch to ad hoc mode. When such mode can be switched, the sources and destinations form an ad hoc network, otherwise, the nodes remain in the infrastructure mode.

We implement the framework in NS2 to evaluate the performance of M^2-WLAN in terms of total system throughput, per-flow throughput, etc. Our simulation results show that M^2-WLAN can substantially increase the system throughput as compared to traditional wireless LANs.

There are many research works on wireless LAN and ad hoc networks. The performance of WLAN is extensively studied and an IETF has been established to provide QoS in 802.11 based networks [2], [3]. Regarding network integration of different architectures, several previous works have been proposed in wireless area. Lin *et al.* present a prototype of multihop wireless LAN and a bridging protocol such that mobile stations can get access to the Internet through access points probably in multiple wireless hops [4]. Wu *et al.* propose a wireless system architecture named iCAR, which integrates cellular and ad hoc relaying systems to address the congestion problem due to unbalanced traffic in a cellular system and provides inter-operability for heterogeneous networks [5], [6]. Such integration is also studied in another protocol named Heterogeneous Wireless Network [7]. To the best of our knowledge, however, no research work has been proposed to integrate infrastructure mode and ad hoc mode to operate together to enhance network performance. Our study differs primarily in that we are considering a unified wireless system in one frequency band, in which the bandwidth is shared by all the communications, and the performance improvement by introducing ad hoc connections is not as obvious and deserves a separate study.

The rest of this paper is organized as follows. We present in Sect. II a framework of building a mixed-mode WLAN. Simulation results illustrating the performance of the framework is presented in Sect. III. We conclude in Sect. IV.

II. A FRAMEWORK FOR MIXED-MODE WIRELESS LAN

In this section, we first look at the difference between infrastructure mode and ad hoc mode in the IEEE 802.11 standard, and then present a framework for the mixed-mode WLAN.

A. Infrastructure Mode versus Ad Hoc Mode

Infrastructure mode (I-Mode) is distinguished from ad hoc mode (A-Mode) by several parameters in the frames, and by the behavior of the transmissions.

In a MAC frame header, *ToDS* and *FromDS* bits indicate whether the frame is transmitted into a Distribution System (DS) or from a DS. When both bits are zero, the communication is in an ad-hoc network. *BSSID* identifies the Basic Service Set (BSS) network into which the frame was transmitted. The BSSID is used in 802.11 to distinguish co-located networks on the same channels. In an infrastructure network, the BSSID is the MAC address of the wireless interface of the access point serving the BSS. In an ad hoc network, the BSSID is a randomly-chosen string. Normally, a node joins a network by choosing a single BSSID, and filters all received frames by the BSSID; i.e., it only receives frames which carry the BSSID used by it. Some frames sent to the AP also include an *Association ID*.

When a node runs in I-Mode, the frames are sent to the AP. Such frames must carry the AP's MAC address as BSSID, and set the ToDS to one and FromDS zero. In contrast, when a node runs in A-Mode, the frames sent to a peer should carry the ad hoc network's BSSID, with ToDS and FromDS set to zero. In this paper, we assume that a node in ad hoc mode can receive frames with correct BSSID sent by an AP.

In our framework, nodes switch between infrastructure mode and ad hoc mode dynamically. Clearly, the mode switching process only needs to change some setting parameters accordingly. Therefore, mode switching can be performed without latency. In the following, we discuss the proposed framework in detail.

B. A Framework for Mixed-Mode Wireless LAN

As mentioned before, the main challenges of building a mixed-mode WLAN are mode switching control and channel management in the network. Hence the framework consists of three components: mode switching algorithms (MSA) in both access point and nodes, a traffic monitoring module (TMM) and a channel management module (CMM) in AP.

The key idea of the framework is as follows. Initially, nodes operate in infrastructure mode (I-Mode). If the number of users is large and traffic load becomes heavy, the AP will drive some local communications to use ad hoc mode (A-Mode). When a communication session can be performed in A-Mode, the nodes involved switch to A-Mode and form an ad hoc network with the given channel. Usually, there are multiple channels available in a wireless network. For example, there are fourteen channels defined in the *DSSS PHY frequency channel plan* in the standard. In this case, the AP has to manage the use of such channels so that they are efficiently utilized. On the other hand, if the source cannot directly reach the destination desired, it still sends data through AP.

Algorithm 1 and 2 are the mode switching algorithms as used by the AP and its users, respectively. The corresponding flowcharts are shown in Fig. 2 and 3. We explain the operations as follows.

When a node joins a network, it has to first communicate with the AP to get authorization and association. It updates the I-Mode channel field in the status table as the channel that the AP uses. The status table only maintains the additional information needed by the framework, and consists of five fields as shown in Table I: *mode, bssid, I-Mode channel, A-Mode channel* and *alive-timer*. The "mode" field specifies the current operating mode of a node. The two channel fields store the channel that a node uses in the corresponding mode. The "alive-timer" stores the time interval that a node reports the existence of the ad hoc network to which it belongs.

When the AP receives a communication request (CR, the node needs to send data or data are sent to the node), it needs to determine

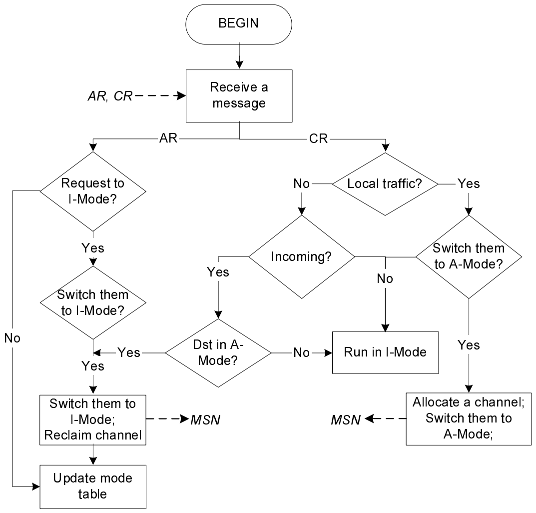


Fig. 2. The flowcharts of mode switching algorithms in AP.

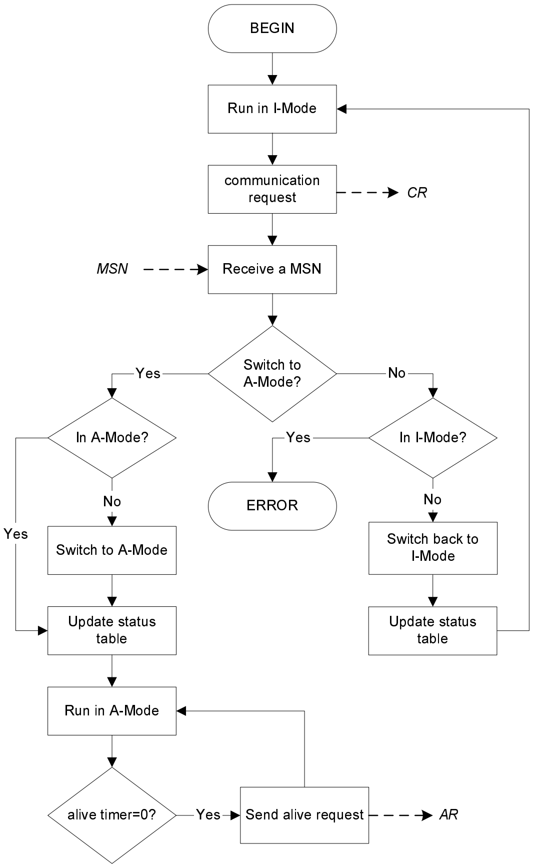


Fig. 3. The flowcharts of mode switching algorithms in nodes.

TABLE I

STATUS TABLE IN A MOBILE NODE.

bssid	mode	I-Mode channel	A-Mode channel	alive-timer
CSD3	A	0	1	10.0

TABLE II

FRAME FORMAT OF THE CONTROL MESSAGES.

AR:	node_id	purpose		
MSN:	bssid	mode	channel	alive-timer

which mode the communication should use. If the communication is non-local, the participants must run in I-Mode since they have to use the AP. If the traffic is local and the AP decides to switch the mode of the nodes, it sends a Mode Switching Notification (MSN) to control the node's access mode. The frame format of the MSN is shown in Table II. It is similar to the status table in a node. When the AP switches a node to A-Mode, it allocates a channel and assigns bssid and alive-timer in the MSN. Upon receiving a MSN, a node switches its mode and updates its status table accordingly.

A node works in I-Mode until the AP switches it to A-Mode. Once a node is in A-Mode, it has the right to determine whether it requests to switch back or not, if the does not send data to them. A node in the A-Mode has to send alive request (AR) to AP periodically so that the AP knows about its existence. Otherwise, the AP will assume that the ad hoc communication is over and reclaim the assigned channel. If the A-Mode nodes have very bad performance (nodes move out of transmission range, high loss rate, etc), they may request to switch back to I-Mode in the AR. The AR frame contains the information that a node need to report to the AP. Its format is shown in Table II. The "node_id" identifies the node who sends the request. The "purpose" field specifies the request purpose. If the frame is a regular alive notification to AP, it is set to zero. If the node requests to switch the mode, the reason is set to "out-of-range" or "low-performance" as shown in Algorithm 2.

As discussed in the MSA of the nodes, a node in A-Mode sends AR periodically. Upon receiving such an AR, the AP responds to the request and updates its mode table accordingly. The mode table in an AP contains the information of all nodes in the network. Each node has an entry in the table. The entry format is shown in Table III. It basically consists of the same information as the status table in a node. In addition, traffic information is also maintained by traffic monitoring module. If the node requests to switch back to I-Mode with the reason of "out-of-range", the AP will turn the nodes in the communication session back to I-Mode. If the reason is "low-performance", the AP accepts it only if traffic load is light.

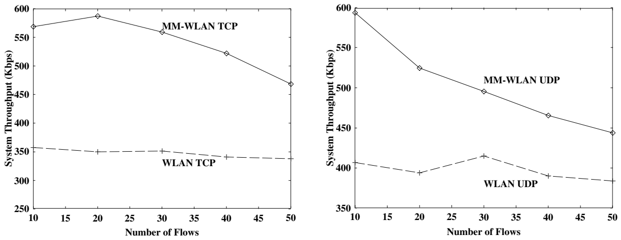
The access point not only controls the modes of the nodes, but also manages the channel usage and monitors the traffic condition in the network. The channel management module keeps track of the usage of each channel. Algorithm 3 shows the basic idea of CMM the in the AP. It assigns a channel when it gets a request from the MSA and reclaims channels when they are released. If one end of a new communication is already in A-Mode and the AP decides that the communication will run in A-Mode, then it uses the same channel assigned already.

In order to distribute the traffic load in different channels, a traffic monitoring module is implemented as shown in Algorithm 4. When

TABLE III

AN ENTRY IN THE ACCESS POINT'S MODE TABLE.

node_id	bssid	mode	A-Mode channel	alive-timer	traffic
1	CSD3	A	1	15.0	local
⋮	⋮	⋮	⋮	⋮	⋮



(a) TCP flows.

(b) UDP flows.

Fig. 4. Throughput comparison in different networks.

the traffic load through the AP is too heavy, it searches in the mode table to find some local communications and tries to turn them into A-Mode such that the traffic crossing the AP can be reduced. When traffic is light, it accepts some return to I-Mode requests from nodes. How to measure traffic load precisely is worth studying. However, it is out of scope of this paper. Hence, we don't discuss it here in details. Currently, we use both the number of flows and channel utilization to measure traffic load. In our simulations, if the number of active flows is larger than a specific number and the AP channel is fully utilized, the traffic is treated as heavy.

Algorithm 1 Mode Switching Algorithm — nodes

```
if (Receive communication request (CR)) then
  if Communication request is local) then
    if (traffic_load > threshold) then
      Allocate a channel;
      Send MSN to switch the src and dst to A-Mode;
      Modify entry in mode table;
    else
      Let them run in I-Mode;
      Modify entry in mode table;
    end if
  else if (Outing communication) then {non-local traffic}
    Let it run in I-Mode;
  else if (The dsts are in A-Mode) then {Incoming traffic}
    Send MSN;
    Turn nodes back to I-Mode;
    Modify entry in mode table;
  end if
end if
if (Receive alive request (AR)) then
  if (Request to switch back) then
    if (Reason is out-of-range) then
      Send MSN;
      Turn the nodes back to I-Mode;
    else if (traffic_load > threshold) then
      if (Need to update alive-timer) then
        Send a new alive-timer;
      end if
    else
      Send MSN;
      Turn the nodes back to I-Mode;
    end if
  end if
  Update the corresponding entry in mode table;
end if
```

III. ILLUSTRATIVE NUMERICAL RESULTS

In this section, we present the performance evaluation of the framework using NS2 (ns-2.1b9a) with the CMU wireless extension.

Algorithm 2 Mode Switching Algorithm — nodes

```
Use I-Mode to associate to AP;
Get I-Mode channel;
Start communication requests (CR);
loop
  if (Receiving a MSN) then
    if (MSN to A-Mode) then
      if (Not in A-Mode) then
        Switch to A-Mode using the given channel;
        Set alive-timer;
      end if
      Update status table;
    else if (Not in I-Mode) then
      Switch back to I-Mode;
    else
      ERROR;
    end if
  end if
  if (This is a sender) then
    if (Cannot reach destination) then
      Set reason as "out-of-range";
      Send alive request (AR) to AP;
    end if
    if (Performance is not good) then
      Set reason as "low-performance";
    end if
  end if
  if (alive-timer = 0) then
    Send alive request (AR) to AP;
  end if
end loop
```

Algorithm 3 Channel Management Algorithm — AP

```
if (Channel request) then
  if (Free channel is available) then
    Assign a free channel;
  else
    Assign the best channel;
  end if
  Update the channel usage table;
end if
if (Channel reclaim) then
  if (No users in the channel) then
    Reclaim the channel;
  end if
  Update the channel usage table;
end if
```

Algorithm 4 Traffic Monitoring Algorithm — AP

```
Monitor traffic load in access point dynamically;
if (Current traffic_load > threshold) then
  if (Possible to turn some connections to A-Mode) then
    Find the best connections in mode table;
    Allocate a channel;
    Send MSN;
    Switch the corresponding nodes to A-Mode;
    Update mode table;
  end if
end if
```

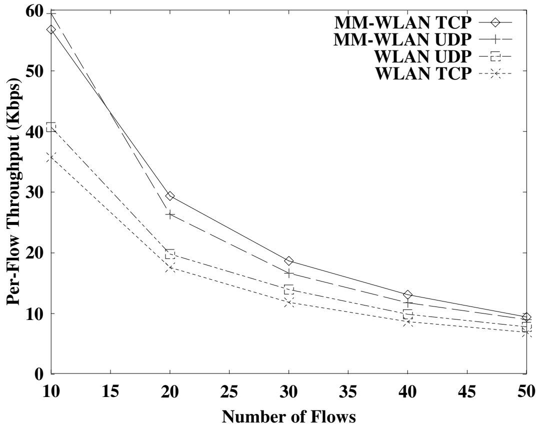
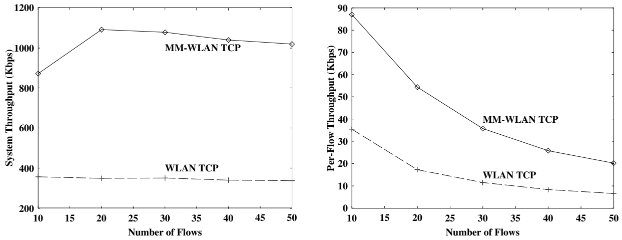


Fig. 5. Per-flow throughput comparison.



(a) System throughput. (b) Per-flow throughput.

Fig. 6. TCP throughput in different networks with two channels available.

We use 2 Mbps for channel and MAC bitrate to simulate IEEE 802.11 WLAN environment. Packet size is 512 bytes unless otherwise specified. For the radio propagation model, a two-ray path loss model is used. We do not consider fading and mobility in our simulations. We assume the transmission range of all nodes including the access point is 250 m. All simulation results are the average of 10 runs and each simulation runs for 20 seconds of simulation time. We use the pure WLAN performance as the baseline to compare with.

We consider a network with a single AP and a number of nodes (from 10 nodes to 50 nodes) uniformly distributed within its transmission range. A TCP or UDP flows is assigned to each node with a random destination. We use total system throughput and per-flow throughput as the performance metrics to study the network. We study the performance of M^2-WLAN with both single channel and multiple channels.

Note that the 802.11 protocol is known to exhibit some form of instability [2]. In particular, as the offered load increases, the throughput goes up to a maximum value. However, further increase of the offered load leads to an eventually significant decrease in the system throughput. Therefore, we have to specify a total offered load for UDP simulation runs. Based on the observation on TCP throughput, we use 600 Kbps as the offered load, i.e., when there are n UDP flows, the assigned bitrate of each flow is $600/n$ Kbps.

We show in Fig. 4 the total system throughput versus the number of flows in the networks with TCP and UDP traffic, respectively. From the figures, we see that for both TCP and UDP, M^2-WLAN achieves substantially higher system throughput than WLAN under all traffic loads (number of flows). For WLAN, its throughput is limited by the AP and channel capacity. As long as the offered load is larger than this limit, it can achieve a constant throughput. For M^2-WLAN, in contrast, its throughput decreases as the traffic load increases. This is because the performance of the ad hoc connections is significantly affected by the traffic condition. Specifically, as the traffic load increases, more collisions may lead to route error and route recovery process, resulting in throughput decrease. However, we still observe more than 10% throughput improvement even at very high traffic load. In other words, the use of ad hoc connections in such networks leads to significant performance gain.

Figure 5 plots the average per-flow throughput versus the number of flows in the networks. Again, the per-flow throughput in M^2-WLAN is always larger than that in WLAN. As the number of flows increases, the per-flow throughput in all networks decreases. This is because the network capacity remains approximately the same.

In Fig. 6, we show the results of TCP throughput in a network with two channels available. Clearly, the WLAN throughput remains the same since it can only make use of one channel. While in M^2-WLAN, the system throughput is improved dramatically, because the network capacity is increased by multiple channels. The ad hoc connections in M^2-WLAN make use of a different channel from the AP. Hence, both traffic load and collisions are reduced on both sides, resulting in significant throughput improvement. Moreover, since the traffic load in each channel is reduced, we do not observe the throughput decrease as in the single channel case. Therefore, if there are multiple channels, M^2-WLAN can lead to significant performance improvement.

IV. CONCLUSIONS

In this paper, we have studied the integration of infrastructure mode and ad hoc mode in a wireless network named mixed wireless LAN (M^2-WLAN). We have proposed a framework to switch some local connections to form small ad hoc networks using available channels. In this framework, a node in a WLAN can dynamically switch between infrastructure mode and ad hoc mode under the control of an access point. The M^2-WLAN is useful for companies looking for fully wireless solutions for office area, where many users need to exchange data among themselves as well as to get access to the Internet. We have implemented the framework in NS2. Via simulations, we show that M^2-WLAN has substantially better performance in terms of system throughput than traditional wireless LAN.

In our upcoming work, we will study power control to enable more spacial channel reuse to further improve the performance of the network. We will also study a network with multiple access points and develop a framework in which mobile nodes roaming around may connect to the nearby nodes by ad hoc connections or access points to get their demanding information.

REFERENCES

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

[2] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[3] J. Q. Bai, , and T. Lang, "A performance comparison between ad hoc and centrally controlled CDMA wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 829–841, Oct 2002.

[4] 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*, pp. 1568 – 1572, IEEE, 1999.

[5] 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, Oct 2001.

[6] 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*, pp. 586–590, IEEE, Fall 2002.

[7] 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*, pp. 3608–3612, IEEE, 2001.