

# Multihop Wireless IEEE 802.11 LANs: A Prototype Implementation

Ying-Dar Lin, Yu-Ching Hsu, Kuan-Wen Oyang, Tzu-Chieh Tsai, and Dong-Su Yang

**Abstract:** In this paper, we present a prototype for a new architecture, MCN (Multihop Cellular Network) implemented over a wireless LAN platform. MCN preserves the virtue of traditional single-hop cellular networks where the service infrastructure is constructed by many BSs (Base Stations) but it also adds the flexibility of ad hoc networks where wireless transfer through mobile stations in multiple hops is allowed. MCN can reduce the number of required BSs or improve the throughput performance. In the prototype of MCN, a bridging protocol, BMBP (Base-driven Multihop Bridging Protocol) runs between mobile stations and access points to enable multihop routing and roaming. Demonstration show that MCN is a feasible architecture for wireless LANs.

**Index Terms:** Multihop, cellular, ad hoc networks, wireless LAN, BMBP, implementation, prototype.

## I. INTRODUCTION

Technologies and services of wireless communication have evolved rapidly during the past decade. Within this field, data-oriented services can further be divided into two categories [1]: 1) low-speed and wide-area systems such as ARDIS (Advanced Radio Data Information Service) [2] and CDPD (Cellular Digital Packet Data) [3] and 2) high-speed and local-area systems such as HIPERLAN (Hi Performance Radio Local Area Network) [4] and IEEE 802.11 [5].

Most of the services mentioned above are based on architectures with single-hop cellular networks (SCN). However, building a large number of BSs, for examples, in the metropolitan area with a dense population has a high cost. In fact, there is another kind of networks; namely, packet radio or ad hoc networks [6], [7], in which no BSs are needed. One of the advantages of these networks is low cost because no infrastructure is needed and the networks can be deployed instantly. However, these ad hoc networks may be limited to specialized applications, such as battlefields and traveling groups due to the vulnerability of paths with possibly many MSs (mobile stations). Nevertheless, this vulnerability can be much reduced if the number of wireless hops can be limited and the station mobility is not high.

In [8], we proposed a new architecture, Multihop Cellular Network (MCN), which is a hybrid architecture of SCNs and ad

hoc networks. The architecture of MCN is quite different from the hierarchical one in [9], where gateways serving each microcell or macrocell are linked to the fixed terrestrial network. In MCN, the transmission range of BSs and MSs are both reduced to  $1/k$  of that adopted in SCN, where  $k$  is defined as the transmission range reduction factor. Thus packets might be forwarded by MSs to arrive destinations in multiple hops. The analysis results in [8] showed that the throughput of MCN is superior than that of SCN and it increases as the transmission power decreases. This is because the mean hop count increases in the order of  $k$ , while the mean number of channels (i.e., simultaneous transmissions) increases in the order of  $k^2$ . In addition to the increased throughput, there are other advantages in MCN: 1) the number of BSs or the transmission range can be reduced, 2) connections are still allowed without BSs, and 3) paths are less vulnerable than the ones in ad hoc networks because the BSs can help to reduce hop count.

To demonstrate the feasibility of MCN architecture, we develop a prototype of MWLAN (Multihop Wireless Local Area Networks), which is an example of MCN and is based on the IEEE 802.11 compliant wireless LAN products developed by CCL, ITRI<sup>1</sup>. In the prototype, a bridging protocol, BMBP (Base-driven Multihop Bridging Protocol), is implemented in both MSs and APs (access points) to add the capabilities of multihop routing and roaming.

The rest of the paper is organized as follows. In section II, we describe the architecture of MWLAN. The design of our bridging protocol, BMBP, is presented in section III. Section IV describes the hardware and software architectures of our MWLAN prototype. Section V demonstrates multihop bridging among MSs. Finally, conclusion and future work are given in section VI.

## II. ARCHITECTURE

MWLAN, in which APs and MSs are not always mutually reachable in a single hop, is an example of MCN by turning the roles of APs in MWLAN to the roles of BSs in MCN. Therefore, besides translating packets between wired and wireless networks, an AP needs to know the MSs associated with it for mobility management. The key feature of MWLAN is that MSs can communicate directly with each others and help to relay packets, which leads to multihop routing.

We define the *reachable area* as the area that APs can reach

Manuscript received January 25, 2000; approved for publication by Dong Ho Cho, Division II Editor, August 17, 2000.

T.-C. Tsai is with Department of Computer Science, National Chengchi University, Taipei, Taiwan.

Other authors are with Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan, e-mail: ychsu@cis.nctu.edu.tw.

<sup>1</sup>Computer and Communication Laboratory, Industrial Technology Research Institute, in Hsinchu, Taiwan.

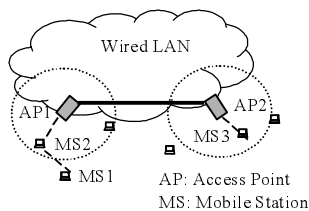


Fig. 1. A bridging example in MWLAN.

directly, as shown by the circled areas in Fig. 1. A bridging protocol, BMBP (Base-driven Multihop Bridging Protocol), which is presented in the next section, runs between MSs and APs to enable multihop routing and roaming. The AP that computes the bridging table for a MS is the one with which the MS is *associated*. Note that an MS is associated with the nearest AP, i.e., the AP to which the hop count is the smallest. When a new call arrives and the source has recorded the path to the destination, packets will be forwarded by the next hop. Otherwise, packets will be sent to the nearest associated AP, probably in multiple hops. Then, also through the BMBP, the associated AP can find the corresponding AP with which the destination is associated. From there, packets can be forwarded to the destination probably in multiple hops again.

Fig. 1 gives the bridging example that the source, MS1, does not record the path for the destination, MS3. Hence, MS1 sends packets to AP1 in multiple hops through MS2. Since AP1 knows, by BMBP, that MS3 is associated with AP2, it forwards packets to AP2. Thus MS3 receives packets from MS1 through AP2. In Fig. 1, the dashed lines indicate the wireless path and the solid line represents the backbone path which is wired in the prototype of MWLAN.

### III. DESIGNING BASE-DRIVEN MULTI-HOP BRIDGING PROTOCOL (BMBP)

The main object of BMBP is to build a bridging table at each node, which could be an AP or an MS. Fig. 2 shows the structure of bridging tables, where DSN (Destination Sequence Number) is used to prevent loops [10]. Also the Local System Clock is stamped by each MS when the MS receives the bridging table from the associated AP and is used to avoid out-of-date entries. The table also records the next hop and hop count to the destination.

Four message types are needed to implement BMBP: *Beacon*, *Hello*, *Bridge*, and *Care-of*. Both APs and MSs announce themselves periodically by Beacon and Hello messages, respectively. A Beacon message is flooded to help MSs identify their associated APs. A Hello message is relayed toward an AP through intermediate MSs which append their information to the message. Hence, when the Hello message arrives in the associated AP, it records the path information from the source to the associated AP. Therefore, the AP can compute and update the bridging tables for itself and for those MSs recorded in the received Hello message. Then the computed bridging tables, packed in Bridge messages, are sent to the MSs. Note that MSs do not have to compute their bridging tables. By exchanging the Care-of messages, APs know the associated AP of a remote

Destination MAC	Destination IP	Destination Sequence Number	Next-hop Mac	Hop count	Local System Clock
--------------------	-------------------	-----------------------------------	-----------------	--------------	--------------------------

Fig. 2. The structure of bridging tables.

Mac Header	Type (Beacon)	AP Information			MS Information			hop count
		IP	Mac	Seq.	IP	Mac	Seq.	

Fig. 3. The format of Beacon messages.

Mac Header	Type (Hello)	N: Entry Count (1~N hops)	Associated AP Information			MS1 Information			... MS N hops Information		
			IP	Mac	Seq.	IP	Mac	Seq.	IP	Mac	Seq.

Fig. 4. The format of Hello messages.

Mac Header	Type (Bridge)	AP Information			N: Entry Count	Destination's Information			Bridging Entry 1	...	Bridging Entry N
		IP	Mac	Seq.		IP	Mac	Seq.			

Bridging Entry i	→	Dest. i's Information			next hop's MAC	hop count
		IP	MAC	Seq.		

Fig. 5. The format of Bridge messages.

MS. Note that each node keeps a sequence number, which is increased by one when it is to be stamped onto a Beacon or a Hello message. The formats of the message types and the protocol procedures are presented below.

#### A. Protocol Messages

##### Beacon Message

Fig. 3 shows the format of Beacon messages which are periodically broadcast by the AP specified in the "AP Information" field. The MS who forwards the Beacon is indicated in the "MS Information" field. Furthermore, the hop count from the AP to the MS is presented in "hop count."

##### Hello Message

Fig. 4 shows the format of Hello messages, which are periodically broadcast by MSs and forwarded toward the "Associated AP." We define  $N$  hops as the maximum hop count that the Hello message can be propagated. The value of " $N$ : Entry Count" field, initialized to be one, indicates the number of MS entries contained and is limited to the value of  $N$  hops to reduce the vulnerability of wireless paths. Each MS information contains the IP address, MAC address and the Sequence number of the MS who relays this message. Note that MS1 is the MS who originates this Hello message.

##### Bridge Message

Fig. 5 shows the format of Bridge messages. This message contains the bridging table, computed by the AP specified in "AP Information" field, for the MS specified in "Destination's Information" field. The " $N$ : Entry Count" field indicates the number of bridging table entries contained in the message. Each entry comprises 5 fields, (destination's IP, destination's MAC, destination's sequence number, next hop's MAC and hop count) so that the MS knows the hop count and the next hop to forward packets toward the destination.

Mac Header	Type (Care-of)	AP Information			MS Information		
		IP	Mac	Seq.	IP	Mac	Seq.

Fig. 6. The format of Care-of messages.

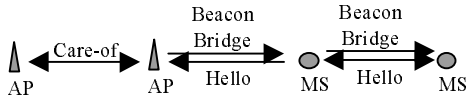


Fig. 7. Messages flow among APs and MSs.

```

Procedure MS_Beacon
Begin
  if(AP.seq >= data[AP.IP].seq)
  {
    if ( AP.seq > data[AP.IP].seq || (hopcount+1) < data[AP.IP].hop count)
    {
      update data[AP.IP];
      if(AP.IP== Associated AP.IP)
        flag=1;
      else if(data[AP.IP].hop count < Associated AP.hop count)
      {
        Associated AP.IP = AP.IP;
        Associated AP.hop count = hopcount+1;
        flag=1;
      }
      if (flag==1)
      {
        update "MS info" in the beacon;
        broadcast the updated Beacon message;
      }
    }
    else does nothing;
  }
End

```

Fig. 8. The activated procedure when an MS receives a Beacon message.

### Care-of Message

Fig. 6 shows the format of Care-of messages, which indicates that the specified MS is associated with the AP recorded in "AP Information." Fig. 7 illustrates the relationship between nodes and four types of messages.

Note that the MAC addresses contained in the MAC Header of Bridge message is set in unicast mode, while others contained in Beacon, Hello, and Care-of messages are set in broadcast mode.

### B. Protocol Procedures at MSs and APs

Beacon messages, originated from APs, are used to help MSs decide their associated APs. Fig. 8 gives the pseudo code to process the Beacon message. Note that an MS records the related information of APs in an array, *data*. If the hop count to the AP who originated the Beacon message is smaller than that to the associated AP, the MS will re-associated with the new AP. When the information of the *Associated AP* is updated, the received Beacon message will be forwarded to notify other MSs of the change.

The pseudo code to process the Hello message at MSs is given in Fig. 9. Note that an MS will only process the received Hello when *Associated AP* in the message is the same as the one it is associated with. Therefore, the Hello will be forwarded toward the *Associated AP*. Since the hop count of a path is limited to the value of *N* hops to reduce the vulnerability of a wireless path, the value in "*N*: Entry Count" field is first checked when an MS receives a Hello message. If the value equals the value of *N* hops, the message is ignored. Otherwise, the MS checks if it has been recorded in the message. If yes, i.e., the message is

```

Procedure MS_Hello
Begin
  if(Hello.Associated AP.IP == Associated AP.IP)
  {
    if ( Hello.N: Entry Count == N hops )
    then does nothing
    else if ( MyIP appears in the Hello message )
    then does nothing
    else
    {
      append MyInfo to the Hello message;
      Hello.N: Entry Count++;
      broadcast the updated Hello message;
    }
  }
End

```

Fig. 9. The activated procedure when an MS receives a Hello message.

```

Procedure AP_Hello
Begin
  if(Hello.Associated AP.IP==My IP)
  {
    for i = "Hello.N: Entry Count" downto 1
    {
      /* compute partial bridging table of the mobile station on Hello.MS i */
      k=1;
      for j=1 to "Hello.N: Entry Count" && j ≠ i )
      {
        flag=0;
        if(Hello.MS i.seq > BS_Table[MS i.IP][MS j.IP].seq)
          flag=1;
        else if(Hello.MS i.seq == BS_Table[MS i.IP][MS j.IP].seq)
          if(|i-j| < BS_Table[MS i.IP][MS j.IP].hop count)
            flag=1;
        if(flag == 1)
        {
          entry k = (Hello.MS j.IP, Hello.MS j.MAC, Hello.MS j.Seq, Hello.MS (i-1).MAC, |i-j|);
          k++;
          update BS_Table[MS i.IP][MS j.IP];
        }
      }
      pack the result in the Bridge message destined for MS i;
      transmit the Bridge message
    }
  }
End

```

Fig. 10. The activated procedure when an AP receives a Hello message.

looped back, the message is also ignored, or the MS will append its information to the message and increase the value in "*N*: Entry Count" by one. Then the MS will re-send this modified Hello message.

Fig. 10 gives the pseudo code for an AP to process a Hello message. The order of the MSs recorded in the Hello message reveals the upstream/downstream connectivity between these MSs. For example, the 3-rd MS can send packets to the first MS through the 2-nd MS in 2 hops or it can also send packets to the 6-th MS through the 4-th MS in 3 hops. Therefore, the AP can compute partial bridging tables of these MSs. Then the computed results, packed in the Bridge messages, are sent to the MSs, probably in multiple hops, beginning from the nearest MS, i.e., one hop away, up to the farthest MS. Further, the AP keeps the computed results in *BS\_Table* locally for some path optimization check. Note that each entry to be packed in a Bridge message should be better, i.e., fresher or shorter, than the old path recorded in *BS\_Table*. Furthermore, the first MS recorded in the message has the largest value of hop count to the AP. Thus, the "for" loop in the pseudo code starts from the *last* MS entry in the Hello message. This processing order is important because the Bridge messages for farther MSs need to be forwarded by the nearer MSs which then should have their bridging tables in place first.

When an MS receives a Bridge message, it checks whether the bridging table belongs to it. If yes, the MS replaces its bridging table with the new one, otherwise, if the MS is the next hop, it replaces the value in "next hop's MAC" by looking up its bridging table and then re-sends the message. Note that the MS will

```

Procedure MS_Bridge
Begin
  if ( Bridge.Destination's Mac == MyMac)
    replace the local bridging table with the new one
  else if ( next hop's Mac == MyMac)
    { update "next hop's Mac" field
      forward this message to next hop
    }
  else
    ignore the message
End
    
```

Fig. 11. The activated procedure when an MS receives a Bridge message.

```

Procedure AP_Care-of
Begin
  if the Care-of-MS is existed in care-of list
    update the care-of address of Care-of-MS
  else insert a new entry to the care-of list
End
    
```

Fig. 12. The activated procedure when an AP receives a Care-of message.

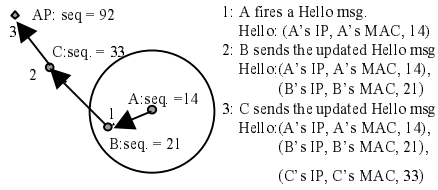


Fig. 13. Propagation of a Hello message.

know the next MS to forward the Bridge message because the AP sends the Bridge message destined to it earlier. If the message is not in above cases, it is ignored. The pseudo code is presented in Fig. 11.

When an AP receives a Care-of message, it records the care-of address of the remote MS recorded in the message. The pseudo code is shown in Fig. 12.

**Example 1:** Fig. 13 is an example illustrating the propagation process of the Hello message. The value of  $N$  hops equals three in this example. First, MS A fires a Hello message. When MS B receives the Hello message, it checks the value of " $N$ : Entry Count" field. Since the value, 1, is less than three, MS B appends its information to the message and then transmits the updated message. Similarly, MS C receives the Hello message from B and does the same thing. Finally, the Hello message is received by the AP. The order, A, B, C, of the MSs recorded in the Hello message reveals that A is the farthest MS to the AP while MS C is the nearest one. Furthermore, partial bridging tables of these MSs are also revealed. For example, B can communicate with A and C directly, and vice versa. Also, there exists a path between A and C through B. Fig. 14 shows the computed partial bridging tables of the AP and MSs, A, B, and C. Note that the computed bridging tables are sent to the MSs in the order of C, B, A.

#### IV. IMPLEMENTATION

Besides BMBP, one little function of sending a string to some specified MS is also implemented to demonstrate the usability of computed bridging tables. The implementation is divided into two parts: access point and mobile station. The OS (operating

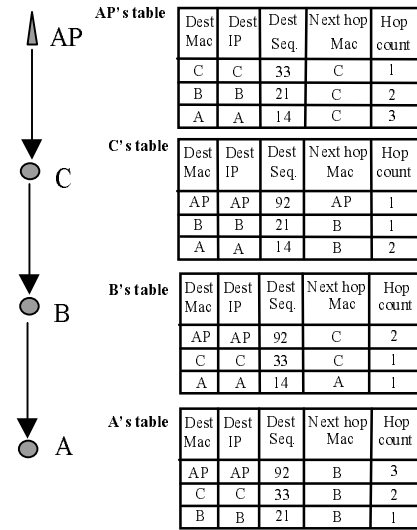


Fig. 14. Computed partial bridging tables from the received Hello message in Fig. 13.

system) for the notebook-based MS is Windows 95 while the MultiTask! real-time OS is used on the desktop-based AP. To lower the complexity of implementation, BMBP is implemented in the application layer. Note that BMBP maybe implemented in MAC layer or in network driver. In such case, it could be performed in view of adding new features, i.e., multihop, either to the current IEEE 802.11 or to the OS, respectively.

##### A. Access Point

The AP is an industry-standard personal computer with an 80486 microprocessor and two network adapters. One adapter is NE2000 compatible, connecting IEEE 802.3 wired LAN and the other is LAN-to-Go, a product of CCL, connecting IEEE 802.11 wireless LAN. Actually, the AP works as a bridge between the Ethernet and the IEEE 802.11 wireless LAN. The NE2000 compatible adapter continuously *intercepts* packets transmitted on the wired network and examines the MAC address of destination contained in the MAC header. If the MAC address belongs to the AP or one of the associated MSs, the packet is further processed. In the latter case the AP plays the role of a bridge where the packet is replicated by the driver of LAN-to-Go adapter and transmitted through the wireless medium. Packets intercepted in the wireless LAN are also processed by the AP in the same manner.

Fig. 15 shows the system architecture for an AP which can be divided into three parts. 1) In order to offer several services simultaneously, such as ftp, tftp and telnet, a Real-Time Operating System (RTOS) named MultiTask! is used to provide multi-task switching. 2) The USNET [11] is a library that supports the functionalities of protocols in network and transport layers, such as TCP, UDP, IP, ICMP, ARP, and RARP. 3) The implemented BMBP, which includes three user tasks, Timertsk, Rxtsk, and Txtsk. Timertsk is in charge of the periodically transmission of Beacon messages. Rxtsk processes the received Hello and Care-of messages and the demonstrating data packet. Txtsk is responsible for transmitting Bridge and Care-of messages and

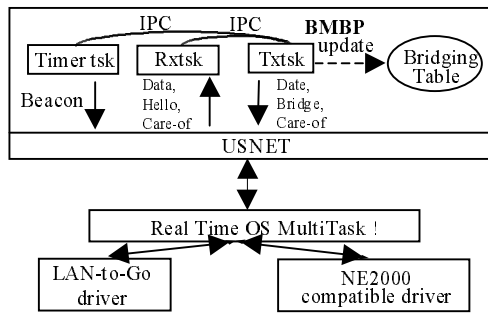


Fig. 15. System architecture for an AP.

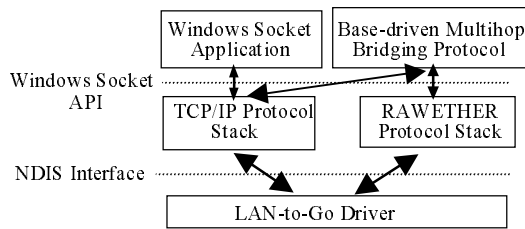


Fig. 16. System architecture for an MS.

data packet. Furthermore, the three tasks need to do IPC (Inter-Process Communication).

### B. Mobile Station

Fig. 16 shows the system architecture of an MS which is implemented on the platform of Windows 95. Besides TCP/IP protocol stack, the RAWETHER protocol stack also works. The LAN-to-Go driver operates in promiscuous mode and forwards all pass-by packets to RAWETHER. The main difference between RAWETHER and TCP/IP is that RAWETHER processes all pass-by packets and preserves the MAC headers of the packets, while TCP/IP only processes the packets destined to them. Therefore, those packets which need to be forwarded by an MS will not be filtered out and BMBP is thus workable.

As defined in the NDIS 3.0 driver specification [12], the drivers of LAN-to-Go and RAWETHER have to communicate with each other through the NDIS Interface Library Wrapper. Fig. 17 illustrates the detailed data paths. The BMBP program fills the next-hop MAC address in the Ethernet header of a packet, which could be a protocol message or the demonstrating data packet, by looking up the bridging table. Then RAWETHER sends out the packet through NDIS Interface Library Wrapper. No matter whether a coming packet is destined to the MS or not, the packet will be intercepted by the LAN-to-Go driver and sent to BMBP through NDIS and RAWETHER. Then BMBP judges whether the packet is destined to it, needed to be forwarded, or can be ignored. If the packet needs to be forwarded, BMBP looks up the MAC address of the next hop, updates the MAC header and then re-sends the packet.

## V. DEMONSTRATION

Two features of BMBP are important to demonstrate that MWLAN is feasible. They are the capabilities of multihop routing and roaming.

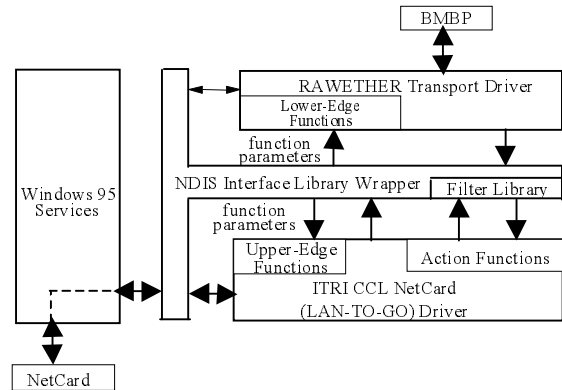


Fig. 17. Frame forwarding paths in an MS.

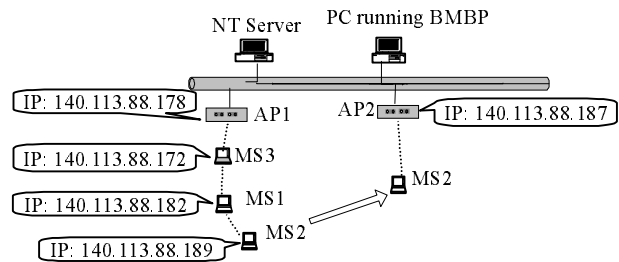


Fig. 18. Demonstration environment.

ing and roaming. Fig. 18 shows the demonstration environment where AP1 and AP2 are on different floors in the same building. Moreover, MS2, which is a notebook, moves up and down. To show the capability of multihop routing, we take the example that MS1 helps MS2 to relay packets to MS3. Regarding the example of roaming, we show that MS2 is still able to send packets to MS3 when it is re-associated with AP2. Note that the value of  $N$  hops is set to three in this demonstration. The IP addresses of AP1, AP2, MS1, MS2, and MS3 are configured as shown in Fig. 18.

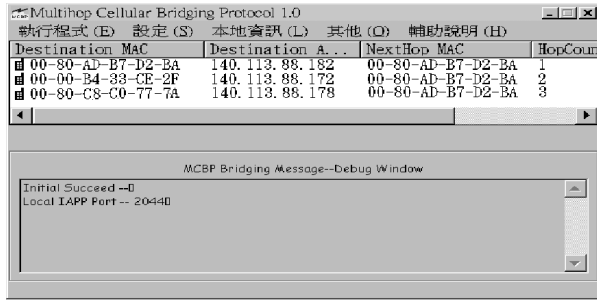
Fig. 19 dumps the bridging table and data frames at MS2. The bridging table of MS2 indicates that MS1 is the next hop to MS3, and the hop count to MS3 is 2. The frames show that MS2 sent four data packets to MS3 and received four replies from MS3.

Fig. 20 dumps the bridging table of MS2 and the data frames sent between MS2 and MS3, after MS2 is re-associated with AP2. The frames illustrate that MS2 sent two data packets to MS3 and received two acknowledgement packets from MS3.

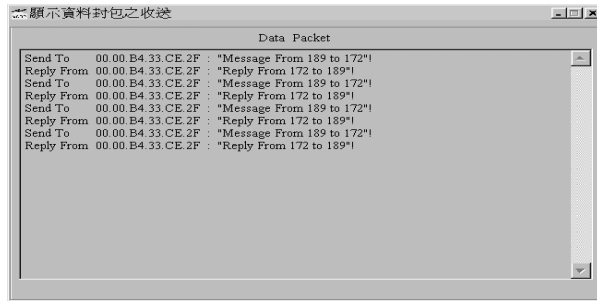
## VI. CONCLUSION AND FUTURE WORK

In this paper, we present the implemented prototype of a MWLAN (Multihop Wireless LAN) which is an example of a more general architecture, MCN (Multihop Cellular Network) [8]. The key component in our design is the bridging protocol, BMBP (Base-driven Multihop Bridging Protocol,) running on APs and MSs to enable multihop routing and roaming. The prototype has demonstrated the feasibility of the MWLAN architecture.

The BMBP is currently implemented in the application layer



Bridging table

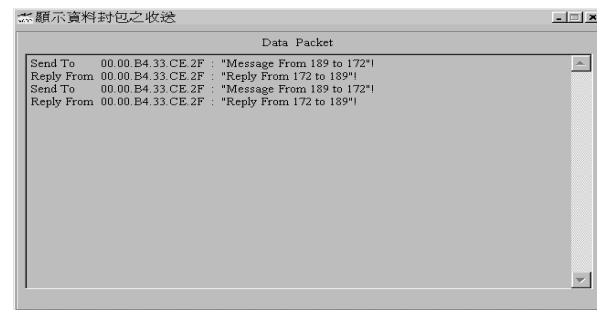


Frames

Fig. 19. Bridging table and frames at MS2 in the multihop routing example.



Bridging table



Frames

Fig. 20. Bridging table and frames at MS2 in the roaming example.

to lower the complexity of implementation by avoiding modifying the underlying drivers. Plans are undertaken to embed the protocol into the driver layer. Meanwhile, a distributed version of the bridging protocol named DMBP (Distributed Multihop Bridging Protocol) is being developed. A comparison between DMBP and BMBP is itemized in Table 1. The distributed bridging protocol has the advantages of better fault tolerance and less-concentrated control traffic, although the computational over-

Table 1. Comparison between DMBP and BMBP.

	DMBP	BMBP
Association Awareness	Access point can not know the associated MSs	Access point can know the associated MSs
Control Traffic	More control traffic	Less control traffic
Traffic Concentration	Control traffic is distributed	Control traffic is concentrated at APs
Mobility Support	For env. with higher mobility	For env. with lower mobility
Complexity	Computational overhead at MSs	Computational overhead at APs
Infrastructure	Workable without APs	Not workable without APs

head is shifted from APs to MSs.

REFERENCES

- [1] K. Pahlavan and A. H. Levesque, *Wireless Information Networks*, Wiley Interscience, 1995.
- [2] W. A. McGladdery and R. Clifford, "Survey of Current And Emerging Wireless Data Networks," *Canadian Conference on Electrical and Computer Engineering*, Vancouver, pp. 1000-1003, Sept. 1993.
- [3] K. C. Budka, "Cellular digital packet data-advanced mobile phone standard network bandwidth contention," in *Proc. 34th IEEE Conf. Decision and Control*, New Orleans, Dec. 1995, pp. 1941-1946.
- [4] T. Wilkinson, T. G. C. Phipps, and S. K. Barton, "A report on HIPERLAN standardization," *International J. Wireless Information Networks*, vol. 2, no. 2, pp. 99-120, 1995.
- [5] IEEE Standards Board, "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *The Institute of Electrical and Electronics Engineers, Inc.*, IEEE Std 802.11, 1997.
- [6] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," in *Proc. IEEE*, vol. 75, no. 1, Jan. 1987.
- [7] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, "Issues in packet radio network design," in *Proc. IEEE*, vol. 75, no. 1, Jan. 1987.
- [8] Y. Lin and Y. Hsu, "Multihop cellular: A new architecture for wireless communications," in *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.
- [9] L. Hu and S. S. Rappaport, "Personal communication systems using multiple hierarchical cellular overlays," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 406-415, Feb. 1995.
- [10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM'94*, 1994, pp. 234-244.
- [11] USNET, available at <http://www.ussw.com/products/>.
- [12] "Microsoft LAN manager, network driver interface specification (NDIS), version 3.0," Microsoft Corporation, 1 Microsoft Way, Redmond.



**Ying-Dar Lin** was born in Hsi-Lo, Taiwan in 1965. He received the Bachelor's degree in Computer Science and Information Engineering from National Taiwan University in 1988, and the M.S. and Ph.D. degrees in Computer Science from the University of California, Los Angeles in 1990 and 1993, respectively. At UCLA Computer Science Department, he worked as a Research Assistant from 1989 to 1993 and worked as a Teaching Assistant from 1991 to 1992. In the summers of 1987 and 1991, he was a technical staff member in IBM Taiwan and Bell Communications Research, respectively. He joined the faculty of the Department of Computer and Information Science at National Chiao Tung University in August 1993 and is Professor since 1999. His research interests include design, analysis, and implementation of network protocols and algorithms, wire-speed switching and routing, quality of services, and intranet servers. He has been a consultant for several high-tech companies and authored two books including a textbook on Computer Network Experiments. Dr. Lin is a member of ACM and IEEE. He can be reached at [ydlin@cis.nctu.edu.tw](mailto:ydlin@cis.nctu.edu.tw) and <http://www.cis.nctu.edu.tw/~ydlin>.



**Yu-Ching Hsu** was born in Tei-Tou, Taiwan in 1973. She received her B.A. and M.S. in Computer and Information Science from National Chiao Tung University in 1995 and 1997, respectively. She is currently a Ph.D. candidate in the same department. Her research interests include protocol design, analysis, and wireless communications. She can be reached at [ychs@cis.nctu.edu.tw](mailto:ychs@cis.nctu.edu.tw).



**Dong-Su Yang** received his B.S. and M.S. in Computer and Information Science from the National Chiao Tung University in 1996 and 1998, respectively. At the same department, he worked as a Teaching Assistant from 1996 to 1998. In the summers of 1996 and 1997, he was a technical staff member in Zcom, Inc. and CCL, ITRI, respectively. He currently works in Coventive Technologies. His interests include design, analysis, implementation of network protocols and algorithms, wireless communications, and embedded system. He can be reached at [gis85507@cis.nctu.edu.tw](mailto:gis85507@cis.nctu.edu.tw).



**Kuan-Wen Oyang** received his B.S. and M.S. in Computer and Information Science from the National Chiao Tung University in 1996 and 1998, respectively. In the summer of 1997, he was a technical staff member in CCL, ITRI. He is an associate researcher at the Telecommunication Laboratories of ChungHwa Telecom., Taiwan since 1998. His research interests are Next Generation Internet and wireless communications.



**Tzu-Chieh Tsai** was born in Tainan, Taiwan, R.O.C. He received his B.S. and M.S. degrees both in Electrical Engineering from National Taiwan University in 1988, and from University of Southern California in 1991, respectively. After that, he joined University of California, at Los Angeles in 1991. He got a Ph.D. degree in Computer Science at UCLA in 1996. Currently, he is an associate professor in Computer Science Department at National Cheng-Chi University, Taipei, Taiwan.

From 1989 to 1990, he was involved in a military joint project with Industrial Technology Research Institute. It included design of channel access protocols, and hierarchical routing for a radio network with tactical units. From 1993 to 1996, he was working on the ARPA WAMIS (Wireless Adaptive Mobile Information Systems) project, including network protocol design and implementation for a multimedia radio network. His research interests include network protocol design and implementation for wireless radio networks; performance evaluation and parallel simulation for distributed computer communication network systems.