WIRELESS INTERNET GATEWAYS (WINGS)

J.J. Garcia-Luna-Aceves,
Chane L. Fullmer, Ewerton Madruga
Computer Engineering Department
University of California
Santa Cruz, CA 95064
jj,chane,madruga@cse.ucsc.edu

David Beyer, Thane Frivold
Rooftop Communications, Corporation
Los Altos, CA 94024
dave,thane@rooftop.com

Abstract— **Today's internetwork technology has been extremely successful in linking huge numbers of computers and users. However, to date, this technology has been oriented to computer interconnection in relatively stable operational environments, and thus cannot adequately support many of the emerging civilian and military uses that require a more adaptive and more easily deployed technology. In particular, multihop packet radio networks are ideal for establishing "instant communication infrastructures" in disaster areas resulting from flood, earthquake, hurricane, or fires, supporting U.S. military doctrine for reliable, secure infrastructures for communication among all tiers down to the soldiers "on-the-move," and extending the global communication infrastructure to the wireless, mobile environment.**

The Defense Advanced Research Projects Agency (DARPA) is sponsoring the development of wireless internet gateways (WINGs) as part of the DARPA Global Mobile (GloMo) Information Systems program. WINGs are wireless IP routers that enable the seamless marriage of distributed, dynamic, self-organizing, multihop wireless networks with the emerging multimedia Internet. This paper describes the WING architecture and novel communication protocols for channel access and routing, as well as the hardware and software development environment used to prototype and demonstrate wireless mobile internetworking.

I. INTRODUCTION

Multihop packet-radio networks (or ad-hoc networks) are an ideal technology to establish an "instant communication infrastructure" for military and civilian applications (e.g., ad-hoc networks for disaster areas resulting from flood, earthquake, hurricane, or fire) in which both hosts and routers are mobile and can have multiple points of attachment to the global IP Internet. In multihop packet-radio networks, there are no dedicated base stations as in commercial cellular networks, and all nodes interact as peers for packet forwarding. This distributed nature eliminates single points of failure and makes packet-radio networks much more robust and survivable than commercial cellular networks. Furthermore, because packet-radio networks can be entirely deployed and operated by the end-users, there is no reliance on a wireless service-provider or a stable backbone infrastructure.

The DARPA packet radio and SURAN programs [2], [3] demonstrated the basic capabilities of ad-hoc networking. However, the ad-hoc networks proposed and implemented to date [8], [1] have been designed as opaque subnetworks using an intranet protocol for packet forwarding that enables packets to flow from one packet radio to the other and from one entry point of the ad-hoc network to an exit point. When the ad-hoc network is used as a subnet in an IP internet, one or more of the packet radios connect to the rest of the IP internet through IP routers in order to provide end-to-end connectivity. IP packets are encapsulated in intranet-level packets, and the routing functions within the ad-hoc network are carried out below the IP routing layer.

Over the past two years, the Wireless Internet Gateways project (WINGS) has introduced and demonstrated an architecture and protocols for *mobile wireless internetworking*, in which packet-radio nodes are wireless IP routers and the global IP Internet is extended to the mobile wireless environment in a seamless manner. Within the WINGS project, Wireless Internet Gateway (WING) prototypes were built to demonstrate the concept, architecture, and protocols for wireless mobile internetworking. A novel feature of the WINGs is that the same protocol code used to debug and analyze new protocols within a Unix simulation environment is also used to control the operation of the actual WING prototypes. The WINGS project is part of the DARPA Global Mobile (GloMo) Information Systems program [15].

This work was supported in part by the Defense Advanced Research Projects Agency under Grants DAAB07-95-C-D157 and DAAH04-96-1-0210.

WINGs are wireless IP routers designed to extend the global IP Internet to ad-hoc networking environments. Like an IP router, a WING accomplishes its routing functions at the IP layer; however, in contrast to wired IP routers, WINGs must also adapt to the dynamics of an ad-hoc network in which routers can move frequently, and must schedule their transmissions to maximize utilization of the available spectrum, while avoiding interference with other transmissions that they may not even be able to detect (the *hidden terminal* problem).

Section II describes our protocol architecture to support wireless mobile internetworking using WINGs. Section III describes the FAMA-NCS protocol (for floor acquisition multiple access with non-persistent carrier sensing), which eliminates the hidden-terminal problems of CSMA in single-channel networks [7]. Section IV describes the wireless internet routing protocol (WIRP), which supports internet routing in the wireless mobile environment. Section V presents the results of a number of simulation experiments designed to show the performance of the entire WING protocol stack. Section VI describes the software and hardware configuration used to build the WING-I prototype, as well as field demonstrations of ad-hoc networks based on WINGs.

II. WING PROTOCOL ARCHITECTURE

Figure 1 shows a high-level description of the WING protocol architecture that includes only the main protocols implemented for the WINGs when they operate over a single channel. The key differences between a WING and a traditional router are that: (a) we have improved upon traditional internet routing protocols like RIP and RIPv2 with WIRP, which can far more effectively handle the topological dynamics and broadcast radio channel of the wireless links; (b) the routing protocol interacts with the link-layer protocols in order to reduce control traffic needed to maintain routing tables; (c) we use a new set of protocols for link control and channel access designed for ad-hoc networks with hidden terminals.

An internal traffic generator (TG), which uses the User Datagram Protocol (UDP), is part of the basic architecture and is used

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
1997		**00-00-1997 to 00-00-1997**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Wireless Internet Gateways (WINGS)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**6**	

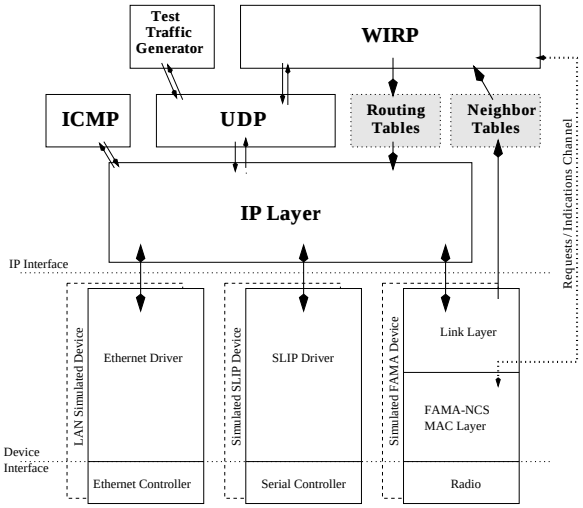

Fig. 1. WINGS I Protocol Architecture

extensively in our simulations and testing of WING prototypes.

The Internet Protocol (IP) uses a standard set of interface functions to access the routing table and to obtain routing instructions for packets being forwarded. The IP protocol's interface to the table is the same regardless of what network protocol is used to update the routing table. Similarly, all protocol modules that are connected to the bottom of the IP modules present the same standard IP interface (IpIf) to allow new protocol interface stacks to be easily added or swapped for existing ones.

The WING currently supports three interface protocol stacks for interfaces to an Ethernet LAN, a SLIP link, and a digital radio device. The FAMA-NCS protocol and a radio link-layer protocol are used to control the underlying radio device. An Ethernet protocol module which includes the Internet standard Address Resolution Protocol (ARP) is used to control the Ethernet device. A simple SLIP protocol module is used to control the underlying serial communications device. A common device applications programmer interface (API) provides a consistent interface structure between the control protocols and each of these interface devices. This API divides the protocol-to-device interface into three fundamental types of primitives: commands, variables, and signals. In addition, this Device API allows the developer to swap an actual interface device driver for one that simulates the communication channel with no changes required of the interface control protocols. For instance, unbeknownst to the MAC and logical link control protocols, the device driver for the radio used in the WING prototype (the Utilicom LongRanger radio) can be swapped for a module that simulates the radio channel in a simulation environment.

Because of its particular importance for developing open-architecture wireless internetwork systems, special attention was given to the definition of the interface between the protocol software and the digital radio modem. This has resulted in the emergence of a pair of standard interface specifications: the *Radio Device API* [12] and the *Physical Radio Interface* [4]. The Radio Device API defines the software interface between the MAC-layer protocols and the "transceiver frame controller" which converts a packet buffers to/from a synchronous bit stream. The Physical Radio Interface defines the lower-layer interface between this transceiver frame controller and the digital radio modem, and consists of a synchronous serial "Data Port" and an abstract "Command Port." The Command Port includes a set of variables, commands, and signals, most of which are also made available to the protocols over the Radio Device API, for controlling and accessing the frequency, RSSI, transmit power, receiver carrier state, and others. The intent of these Radio APIs is to facilitate both collaboration and independent development of the network protocols and digital radio modem hardware which can be easily mixed and matched into well-integrated systems.

III. FAMA-NCS

FAMA-NCS is similar to the protocol proposed for IEEE 802.11 [5]. However, this and prior protocols based on handshakes (also called collision avoidance) and carrier sensing or packet sensing do not provide floor acquisition in networks with hidden terminals [7].

A station that has just been initialized must wait the time it takes to transmit the maximum-size data packet in the channel plus one maximum round-trip time across the channel. This allows any neighboring station involved in the process of receiving a data packet to complete the reception un-obstructed. The initialization time also gives the station the ability to learn of any local traffic in progress. If no carrier is detected during the initialization period, the station transitions to the PASSIVE state. Otherwise, it transitions to the REMOTE state. A station can only be in the PASSIVE state if it is properly initialized (i.e., has no packet to send, and senses an idle channel). In all other states, the station must have listened to the channel for a time period that is sufficiently long for any neighbor involved in receiving data to have finished.

A station that is in the PASSIVE state and senses carrier transitions to the REMOTE state. On the other hand, a station that receives a packet to send in the PASSIVE state transmits a request-to-send (RTS) and transitions to the RTS state. The sending station waits long enough for the destination to send the clear-to-send (CTS) to the RTS. If the CTS is not received within the time allowed, the sender transitions to the BACKOFF state. If the sender hears noise on the channel after its RTS, it assumes a collision with a neighbor's dominating CTS and waits long enough for a maximum-length data packet to be received. Otherwise, upon receiving the CTS, the sender transmits its data packet. Because the CTS could be corrupted at the sender, once the destination station sends its CTS, it only needs to wait one maximum round-trip time to sense the beginning of the data packet from the source. If the data packet does not begin, the destination transitions either to the BACKOFF state (if it has traffic pending) or to the PASSIVE state.

In the BACKOFF state, if no carrier is detected during the entire backoff waiting period computed by the station, the station transmits an RTS and transitions to the RTS state as before. Otherwise, upon sensing carrier the station transitions to the REMOTE state. Any passive station that detects carrier transitions to the REMOTE state, and waiting periods are enforced after the channel clears based on what the station last heard (noise, a control packet, a data packet). Such waiting periods allow RTS/CTS exchanges and packet trains to terminate [7].

The channel becomes idle when all stations are in either the PASSIVE or BACKOFF state. The next access to the channel is driven by the arrival of new packets to the network and retransmission of packets that have been backed off.

The length of a CTS is larger than the aggregate of the length of

an RTS plus one maximum round trip time across the channel, the transmit to receive turn around time, and any processing time; the length of an RTS is larger than the maximum channel propagation delay plus the transmit-to-receive turn around time and any processing time. This is required to avoid one station hearing a complete RTS before another has started to receive it. The relationship of the size of the CTS to the RTS gives the CTS *dominance* over the RTS in the channel. Once a station has begun transmission of a CTS, any other station within range of it that transmits an RTS within one propagation delay of the beginning of the CTS hears at least a portion of the dominating CTS and backs off, thereby letting the data packet that will follow to arrive free from collision. The dominating CTS of FAMA-NCS plays the role of a busy tone sent in the same channel as data packets.

To increase the efficiency of the channel, a station that has successfully acquired the floor can dynamically send multiple packets together in a train, bounded by an upper limit. The signaling required to support packet trains with hidden terminals is described in [7]

IV. WIRP

The Wireless Internet Routing Protocol (WIRP) was designed for an IP internet in which topology changes are the rule, rather than the exception, and where control traffic must be limited. It runs on top of UDP and it can be functionally divided into three main components: Reliable exchange of updates, neighbor discovery mechanism, and its path-finding routing algorithm (PFA).

A. Reliable Transmission of Updates

Reliable transmission of update messages is implemented by means of multicasting of update messages that are acknowledged with update messages carrying both updates and acknowledgments to one or more other messages.

After receiving an update message free of errors, a node is required to acknowledge, which indicates that there is good connectivity with the neighbor and that the neighbor has processed the update message.

An update message is retransmitted if acknowledgments are missing after a finite time and specifies which neighbors should acknowledge. A WING keeps a Message Retransmission List (MRL) with the neighbors whose acknowledgments are still missing [9].

B. Neighbor Discovery Mechanism

Every WING checks the connectivity with its neighbors periodically. A WING transmits a HELLO packet if it does not have any data packet or routing-table update message to transmit during a HELLO interval. In the current implementation, the HELLO interval is set to 3 seconds.

To interoperate, WIRP and FAMA-NCS share a *Neighbor Information Table* (NIT) and a *Subnet Activity Table* (SAT). The NIT table contains an entry for each neighbor with a flag and a counter. FAMA-NCS sets the flag for a particular neighbor every time it hears a packet (control or data) with that neighbor as the source station. WIRP periodically scans the table to increment the counters and reset the neighbor flags to 0. The SAT table contains an entry for each subnet attached to the FAMA-NCS domain with a flag. FAMA-NCS sets the flag every time it sends a data packet to a particular subnet. WIRP also periodically scans this table and resets the flags.

In addition to these tables, a message channel is used to send requests and indications between WIRP and FAMA-NCS. When FAMA-NCS cannot successfully send a packet to any given destination (i.e., no CTS response is received after several RTS transmission to the destination) an indication is sent to WIRP informing it that a packet was dropped for the destination. WIRP can also send requests to FAMA-NCS. WIRP can tell FAMA-NCS which *proxy* address to use for broadcast packets at any given time, and to send *explicit* HELLOs when WIRP deems it necessary. FAMA-NCS sends explicit HELLOs by sending an RTS with a special destination address (different from the proxy for broadcast address) which no station will respond to directly, but will still send the source address up to its own WIRP layer as an implicit HELLO simply by having heard the control packet.

C. Wireless Internet Routing

The basic design concept in WIRP is simple. Each WING communicates to its neighbors a hierarchical routing tree in an incremental fashion. The hierarchical routing tree reported by a WING consists of all the WING's preferred shortest paths to each known IP network and IP host, where an IP host is typically a WING. An entire remote IP network is simply a node in the routing tree.

WINGs exchange their hierarchical routing trees incrementally by communicating only the distance and second-to-last hop (predecessor) to each destination. In the case of destinations within a WING's own IP network, the second-to-last hop consists of a WING (i.e., a host-level IP Address). In the case of a remote IP network known to the WING, the predecessor consists of another IP network. Hence, internet routing in WIRP does not require a WING to store more routing-table entries than an Internet routing protocol like RIPv2 would, for example.

In essence, WIRP implements Dijkstra's shortest-path algorithm distributed over a hierarchical graph representing the connectivity of IP networks as well as the connectivity of the WING's own IP network(s). The algorithm used for this purpose is a modification of the path-finding algorithm (PFA) [9].

The entry for destination j in WING i's routing table consists of the destination's IP address, the distance to the destination (D_j^i), the successor (s_j^i), and the predecessor (p_j^i) along the preferred path (shortest path) to the destination. Routing information is exchanged among neighboring WINGs by means of update messages. An update message from WING i consists of a vector of entries reporting incremental updates of its routing table; each entry specifies a destination j (i.e., an individual host or an IP network), the reported distance to that destination, and the reported predecessor (individual host address or an IP network) in the path to the destination.

Because every WING reports to its neighbors the second-to-last hop in the shortest path to the destination, the complete path to any destination (called the implicit path to the destination) is known by the WING's neighbors. This is done by a path traversal on the predecessor entries reported by the WING. This accounts for the elimination of counting to infinity problems in WIRP that plague RIP.

V. Simulation Experiments

The average throughput of FAMA-NCS and the effectiveness of WIRP in providing new paths after topology changes were analyzed by simulation using the C++ Protocol Toolkit (CPT) [13] on a Sun Ultra II Sparc workstation.

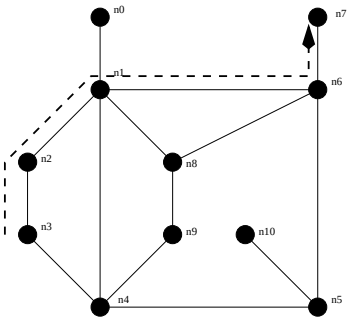

Fig. 2. WIRP routes established during simulation at startup

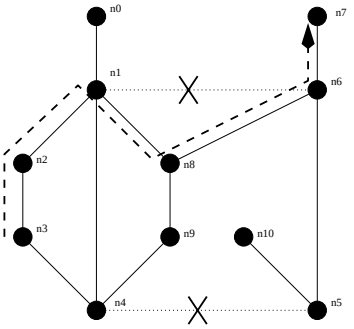

Fig. 3. WIRP routes established during simulation after links are lost

Figure 2 shows the "Los Nettos" network topology used in the simulations; the average degree of nodes in this topology is approximately three. We used two different types of channels for our simulations. In the first case, the nodes were capable of a maximum transmission rate of 1Mb/s and a zero transmit to receive turnaround time, with no preamble or processing time included. In the second case, we simulated the parameters of the Utilicom model 2020 radio device, which is the current platform of the WING I prototype. The Utilicom radio introduces a 5ms transmit ramp up time and a 5ms ramp down time; this includes a 745-bit preamble (for capture) and a 3ms capture release delay. Finally, the Utilicom radios do not provide true carrier sensing of the channel, and provide only a capture detection signal. As such, these devices cannot *hear* noise and signal the MAC layer of activity on the channel (i.e., a dominant CTS, or collisions by other nodes).

Nodes were separated by a distance of approximately one mile from each other, giving a propagation delay of about 6μs. In addition, each node had a single 20-packet output buffer at the MAC layer for all data packets, and a separate queue for control packets. FAMA-NCS attempted 10 transmissions to deliver a packet to the radio channel before giving up.

To test the convergence capabilities of WIRP, a single stream was initiated between two nodes on opposite sides of the network. The nodes were started with empty routing tables and allowed to find each other and stabilize for 50 seconds of simulation time. A UDP traffic stream was then started using the test traffic generator (TG) from Node 3 to Node 7 sending packets of 500 bytes at a rate of two packets per second. After the stream had been flowing for 100 seconds the links between Node 1 and Node 6 and between Node 4 and Node 5 were blocked (as in a long period of fading, or obstruction in the path). The simulation was allowed to run another 100 seconds.

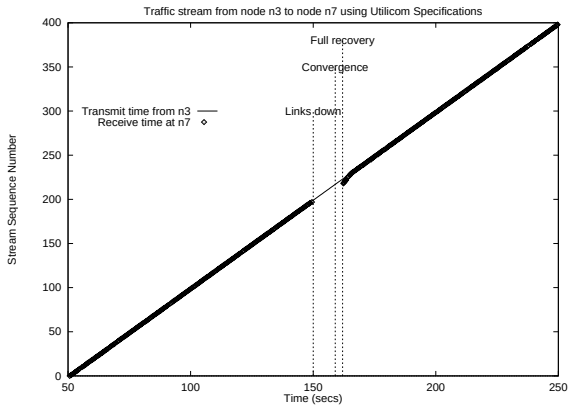

Fig. 4. Sequence ID versus time for stream from n3 to n7 for Utilicom radios

Figures 2 and 3 show the route established through WIRP before and after the links were blocked. Figure 4 shows the arrival of packets at the receiver by sequence number versus time for a network using the Utilicom parameters. Also shown are the point where the links were broken in the topology, the point where Node 1 converged to the new route , and the point of full recovery when the stream was again delivered to the destination. The time for Node 1 to converge with a new route for the stream to Node 7 was 9.1 seconds, full recovery of the stream at Node 7 took an additional 3.2 seconds for a total of 12.3 seconds, with a loss of 21 consecutive packets out of the stream.

To verify attainable throughput, we ran simulations using both the 1Mb/s and 298Kb/s channels with 500 byte data packets and 25 byte RTS. A set of traffic streams (from the TG) was started from edge nodes n0, n7, and n10, and inside nodes n2, n3 and n9. Each set consisted of a stream from the node to each of the other nodes in the network, and an echo back of the test packet. Nodes n1, n4, n5, n6 and n8 did not originate any streams (however, they did echo test packets as well as forward traffic to others).

Figures 5 and 6 shows the average throughput per node over one second intervals. Figure 7 shows the average delay for all packets received during a given interval (one second). The delay for the 298Kb/s channel is an order of magnitude greater than that for the 1Mb/s channel; as the throughput seen for the 1Mb/s channel is an order of magnitude greater than that of the 298Kb/s channel, this is an expected result. These simulation results agree with the performance predicted by the analytical model of FAMA-NCS with hidden terminals [7].

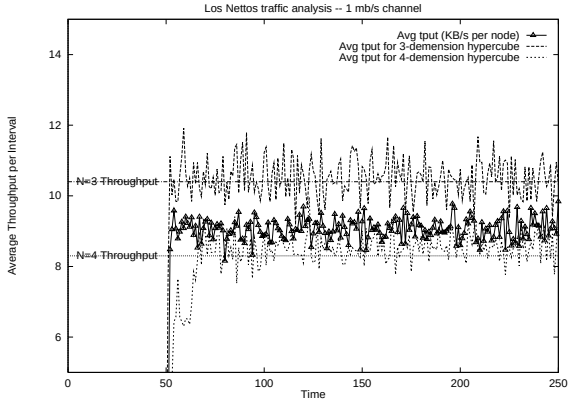

Fig. 5. Average throughput per node in the Los Nettos topology with ideal radios

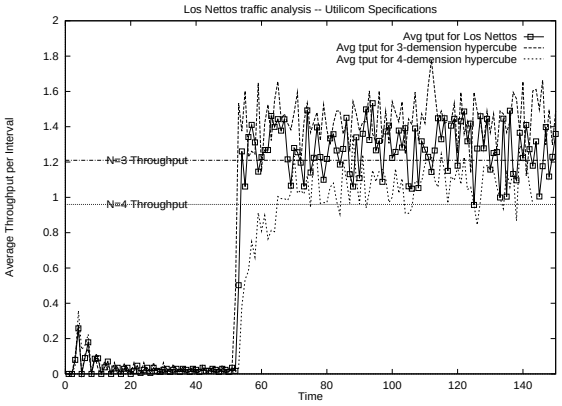

Fig. 6. Average throughput per node in the Los Nettos topology with Utilicom radios

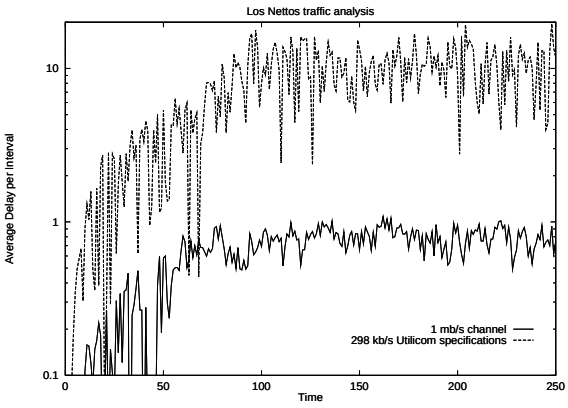

Fig. 7. Average delay per node in the Los Nettos throughput tests

VI. Implementation Experience

A. WING Software Approach

A well-defined, object-oriented framework for linking protocol modules was created within the project to allow the individual protocols to be coded and tested independently by multiple developers, and then easily integrated, swapped, or added into complete protocol stacks. This framework consisted of C++, Application Programming Interfaces (APIs) defined at key protocol boundaries and tables in the WINGS protocol stack. The design of this framework was facilitated by the use of the core protocol library objects available in the C++ Protocol Toolkit (CPT) [13] discussed below.

Development of the WINGs protocols has been facilitated through the use of CPT. CPT was created to support the efficient development, testing, and analysis of protocol software within a realistic simulation environment, and then allow the seamless transition of this same protocol software into an embedded hardware system. This support for seamless transition of the protocols into an embedded system is in stark contrast to the traditional two-phase approach where protocols are first developed and tested on simulation systems, and then re-implemented for a target embedded system. In particular, the development of the WINGs protocols benefited from the following key capabilities of the CPT:

- Rapid and Reliable Transition to Embedded Systems. Minimal, well-defined interfaces to device modules and the hardware system platform, allow protocol software to be transitioned from a simulation environment to an embedded system simply by recompiling and relinking with a new platform wrapper and device drivers libraries.
- The CPT Protocol Framework. The object-oriented, highly-instrumented, and robust CPT Protocol Framework library speeds development of network protocols by presenting the developer with standard, protocol-relevant objects such as packets, queues, timers, protocol modules, and state machines. Also, this framework provides a consistent protocol structure to permit the mixing-and-matching of protocol stacks.
- Realistic Simulations. A realistic simulation capability, particularly well-suited for wireless networks, allows the performance and reliability of the network protocols to be extensively tested in a highly-instrumented simulation environment prior to field testing.
- Public-Domain Graphical Analysis Tools. The performance and behavior of CPT simulations and operational networks can be analyzed using a suite of public-domain filtering and graphic visualization tools including the NetViz network animation tool [14].

B. WING Hardware Configuration

The hardware platform for the WING prototypes are based on a Motorola, 68360-based controllers for running the protocols and supplying the serial communication channels for communicating with digital radio modems (over the Physical Radio Interface [4]). The WING I prototype uses a 298-Kbps, direct-sequence spreading radio from Utilicom Inc. Table I provides the specifications for the WING I prototype.

However, it is important to note that, due to the flexibility of the 68360's communication capabilities and the growing acceptance of the Radio API specifications, the WING controller can be used in conjunction with a variety of other radios. For example, during the WINGS project and related efforts, the WING controller has been effectively integrated with two other commercial radios (one being a 1-Mbps, frequency-hopping radio by Netwave), and plans are currently being made to integrate the WING controller with radios being developed as part of the GloMo Program by UCLA, Virginia Tech, and ISI.

C. Wireless Internetworking Demonstrations

The CPT simulator was incorporated into the WINGs from its inception in November 1995. The baseline protocols were completed and installed on the first embedded system in May, 1996. In July, 1996 a WING ad-hoc network was demonstrated to the GloMo community at the CalNeva Lodge in Lake Tahoe, California. One WING was connected through a SLIP link to a local ISP, and three more were setup though the lodge to form a three-hop network connecting to a laptop running WWW sessions. In a second demonstration a satellite feed from Hughes Research Labs (HRL) was sent over a WaveLan link to a commercial router connected to a WING router and to the laptop via a single-hop WING network.

The WIRP and FAMA protocols were installed and operational on the WINGs in November 1996. In February 1997, these WINGs were demonstrated at the GloMo PI meeting. The network configuration consisted of a hub connected to the UCLA campus network. One WING was connected to the hub and served as the border router for the rest of the WING and their

Protocol Processor	25 MHz Motorola 68360
Memory	4 MByte RAM, 1 MByte Flash ROM
Wired Interfaces	Host Port: SLIP RS-232, 57.6 Kbps max rate LAN Port: 10 Mbps 10BaseT Ethernet Console: RS-232, 115 Kbps max bit rate
RF Frequency Range	905 to 925 MHz center frequency software selectable by 100 kHz increments.
RF Modulation Type	QPSK direct sequence
RF Output Power	800 mW (29 dBm) maximum software controlled for lower power settings.
RF Radiated Power	4 Watts (36 dBm) at maximum output power with a 7 dBi-gain antenna, neglecting cable loss.
Receiver Sensitivity	-92dBm at 10^{-6} bit error rate (BER) (at the code length and modulation used by the WING I)
Approximate Link Range	7 miles multipoint-to-multipoint max. 15 miles point-to-multipoint max. 30 miles point-to-point max.
PN Code Rate	4.6 Mchip/second
PN Code Length	31 chips/symbol 2 bits/symbol (15.5 chips/bit)
Channel Bit Rate	298 Kbps
Power Requirements	12 VDC at 1.1 Amps (11 Watts), receiving 1.25 Amps (15 Watts) transmitting
Dimensions	Controller: 7.25" W x 1.5" H x 6.5" D Radio: 4.125" W x 1.5" H x 6.5" D

TABLE I
WING I HARDWARE SPECIFICATIONS

respective clients. Two additional WINGs, each with a FreeBSD client attached to the Ethernet port, were operational in the network. Three internetworking demonstrations were accomplished. A video stream was sent between the two WING clients running FreeBSD and using the VIC Mbone tool over the WING link. Rates of eight to ten frames per second were shown. HRL again provided a satellite video feed as in the Tahoe demonstration, this time to the local subnet. A live video transmission was received and shown also at eight to ten frames per second. The WING router was instantiated in the UCLA routers to the DARTNET connection, and clients on the WING subnet were able to access and download files across DARTNET (i.e, clients were able to connect to SRI International's HTTP server to download files from it).

VII. CONCLUSIONS

We have presented the architecture, main protocols, and implementation of Wireless Internet Gateways (WINGS), wireless IP routers designed to provide wireless mobile internetworking over ad-hoc networks.

The WINGs and the basic concept of achieving mobile wireless internetworking have been demonstrated successfully in the DARPA GloMo program, and our work continues to analyze improvements on the initial protocols being used in the WINGs today. In particular, analyzing the performance of different types of routing and channel access protocols capable of using multiple channels as well as applying intelligent control of other link characteristics is an attractive area of research.

We have shown that using the FAMA-NCS protocol, a given station and it's neighbors are able to utilize at least one third of the channel capacity in the worst case (with all neighbors hidden from each other). This is in remarkable contrast with CSMA, whose behavior degrades to the basic ALOHA protocol under hidden terminals, which renders throughputs smaller than 18% because of the need to relay and acknowledge packets. The simulation results obtained using the parameters of the Utilicom radio also show the importance of carrier sensing; because the Utilicom radio does not provide true carrier sensing, the performance of FAMA-NCS degrades substantially, as predicted by the theory [6], [7].

We have also shown that WIRP provides internet routing in the ad-hoc network environment and converges efficiently, even when competing with heavy traffic for bandwidth to send it's routing-table update information. The simulation results presented assumed *single-path routing*, in which the protocol provided a single path to each destination. A new version of WIRP provides multiple paths, and we are developing new queueing schemes for the WINGs to establish a late binding of packets to their next hops, so that packets can be rerouted around failures more efficiently.

Implementing the WINGs has been simplified by our use of CPT, which allowed us to carry out simulations of large network topologies using the complete WING protocol suite, with each protocol being implemented exactly as it would be running in a WING, and then use the very same code written for our simulations in the actual prototype by simply recompiling. This eliminated the time needed to rewrite the protocols, as well as the associated recoding errors.

REFERENCES

[1] A. Alwan, R. Bagrodia, N. Bambos, M. Gerla, L. Kleinrock, J. Short, and J. Villasenor, "Adaptive Mobile Multimedia Networks," *IEEE Personal Communications*, pp. 34-51, April 1996.

[2] D. Beyer, "Accomplishments of the DARPA SURAN Program." *Proc. IEEE MILCOM '90 Conference*, Monterey, California, October 1990.

[3] D. Beyer, et al, "Packet Radio Network Research, Development, and Application." *SHAPE Conference on Packet Radio*, Amsterdam, 1989.

[4] D. Beyer, "Physical Radio Interface," Rooftop Communications Technical Document, May 1997, http://www.rooftop.com, "Radio Interface" Page.

[5] K. Biba, "A Hybrid Wireless MAC Protocol Supporting Asynchronous and Syncronous MSDU Delivery Services," Tech. Rep. Paper 802.11/91-92, IEEE 802.11 Working Group, 1992.

[6] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks", in *Proceedings of ACM SIGCOMM '95*, ACM, 1995.

[7] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," in *Proceedings of ACM SIGCOMM '97*, ACM, 1997.

[8] B.M. Leiner, D.L. Nielson, and F.A. Tobagi (Eds.), *Proc. IEEE*, Packet Radio Networks Special Issue, Jan. 1987.

[9] S. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal*, Special issue on Routing in Mobile Communication Networks, Vol. 1, No. 2, 1996.

[10] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk, Second Edition*. Addison-Wesley Publishing Company, Inc., 1990.

[11] D. Beyer, T. Frivold, "Wireless Internet Gateway (WING) I Advanced Operator's Manual," Technical Manual for DARPA-funded Wireless Internet Gateways (WINGS) Project, October 1996.

[12] D. Beyer, T. Frivold, D. Lancaster, M. Lewis, "Radio Device API," Rooftop Communications Technical Manual, December 1996, http://www.rooftop.com "Radio Interface" page.

[13] D. Beyer, B. Nguyen, "The C++ Protocol Toolkit: Overview," Rooftop communications Technical Manual, December 1995. Also see http://www.rooftop.com "Research and Development" page.

[14] D. Beyer, "Network Visualizer (NetViz)," Network animation tool. http://www.rooftop.com "Research and Development" page.

[15] DARPA, *DARPA Global Mobile GloMo Information Systems Program*. http://www.glomo.sri.com/

[16] D. Beyer, M. Vestrich, and J. J. Garcia-Luna-Aceves. "The Rooftop Community Network: Free, High-speed Network Access for Communities," to be published by Harvard University's Information Infrastructure Project, Spring 1997.