ARCHITECTURE AND PROTOTYPING OF AN 802.11-BASED SELF-ORGANIZING HIERARCHICAL AD-HOC WIRELESS NETWORK (SOHAN)*

S. Ganu, S. Zhao, L. Raju, B. Anepu, I. Seskar and D. Raychaudhuri

WINLAB, Rutgers University, 73 Brett Road, Piscataway, NJ 08854
{sachin, sulizhao, lalit, bhaskar, seskar, ray}@winlab.rutgers.edu

Abstract - This paper describes the design and implementation of a novel 802.11-based self-organizing hierarchical ad-hoc wireless network (SOHAN), and presents some initial experimental results obtained from a proof-of-concept prototype. The proposed network has a three-tier hierarchy consisting of low-power *mobile nodes* (MNs) at the lowest layer, *forwarding nodes* (FNs) with higher power and multi-hop routing capability at the middle layer, and wired *access points* (APs) without power constraints at the highest layer. Specifics of new protocols used for bootstrapping, node discovery and multi-hop routing are presented, and overall operation of the complete hierarchical ad-hoc network is explained. A prototype implementation of the SOHAN network is outlined in terms of major hardware and software components, and initial experimental results are given.

Keywords - Ad-hoc network, discovery and routing protocols, sensor networks, system prototyping.

I. INTRODUCTION

This paper describes a novel self-organizing hierarchical ad-hoc wireless network ("SOHAN") designed to provide significant improvements in system capacity and performance relative to conventional "flat" ad-hoc networking approaches. The proposed hierarchical ad-hoc network is motivated by the fact that flat ad-hoc architectures do not scale well as the number of radio nodes becomes large [1]. In addition, most realistic usage scenarios involve predominant mobile device traffic flows to and from the wired Internet, thus requiring effective integration of wired "access points" with the ad-hoc wireless network nodes.

The approach adopted here is based on a multi-tier hierarchy that scales well and integrates naturally with existing wireless access points or base stations, while retaining much of the robustness, coverage and power advantages of ad-hoc wireless networks. This architecture is applicable to a number of emerging ad-hoc networking scenarios including extended wireless local-area networks, home wireless networks and large-scale sensor networks. In each of these scenarios, the introduction of one or more tiers of ad-hoc forwarding nodes (FN) as intermediate radio relays between the MNs and APs helps to scale network throughput, reduce delay and lower power consumption at end-user devices. In this paper, we focus on practical design aspects and prototype implementation of protocols used in the SOHAN ad-hoc network, including those used for node bootstrapping, node discovery and multi-hop routing. More detailed consideration of system capacity scaling and network performance of the hierarchical ad-hoc network for alternative routing methods can be found in [2,3,4]. Design trade-offs for both discovery and routing are discussed, and a proof-of-concept prototype (implemented on Linux platforms with 802.11b radios) is described in terms of hardware and software components. Selected validation experiments and measurements are also given for the prototype hierarchical ad-hoc network.

II. SYSTEM ARCHITECTURE

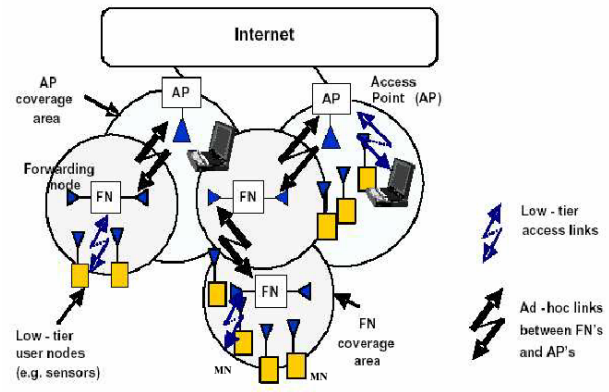

Figure 1. Three tier ad-hoc network architecture

The three-tier hierarchical ad-hoc network consists of the following components: Low-power end-user "mobile nodes" (MN) at the lowest tier, higher powered radio "forwarding nodes" (FN) that support multi-hop routing at the second level, and wired access points (AP) at the third and highest level. Each of the network entities in the proposed system is defined in further detail below:

Mobile Node, (MN), is a mobile end-user device (such as a sensor or a personal digital assistant) at the lowest tier (tier 1) of the network. The MN attaches itself to one or more nodes at the higher tiers of the network in order to obtain service using a discovery protocol. The MN uses a single

* Research supported in part by NJ Commission of Science and Technology Grant #03-2042-007-12, and in part by a grant from Cisco University Programs

802.11b radio operating in ad-hoc mode to communicate with the point(s) of attachment. As an end-user node, the MN is not required to route multi-hop traffic from other nodes. It is noted that as a battery-operated end-user device, the MN will typically have energy constraints

Forwarding Node (FN), is a fixed or mobile intermediate (tier 2) radio relay node capable of routing multi-hop traffic to and from all three tiers of the network's hierarchy. As an intermediate radio node without traffic of its own, the FN is only responsible for multi-hop routing of transit packets. A forwarding node with one 802.11 radio interface uses the same radio to connect in ad-hoc mode to MNs, other FNs and the higher-tier APs defined below. Optionally, an FN may have two radio cards*, one for traffic between FNs and MNs and another for inter FN and FN-AP traffic flows (typically carried on a different frequency). The FN is typically a compact radio device that can be plugged into an electrical outlet, but in certain scenarios, may also be also be a battery-powered mobile device. Thus, the FN is also energy constrained, but the cost is typically an order of magnitude lower than that of the MN defined above

Access Point (AP), is a fixed radio access node at the highest tier (tier 3) of the network, with both an 802.11 radio interface and a wired interface to the Internet. The AP is capable of connecting to any lower tier FN or AP within range but unlike typical 802.11 WLAN deployments, it operates in ad-hoc mode for each such radio link. The AP also participates in discovery and routing protocols used by the lower tier FNs and MNs., and is responsible for routing traffic within the ad-hoc network as well as to and from the Internet. Logically, the tier 3 APs are no different from tiers 1 and 2 when routing internal ad-hoc network traffic - the wired links between APs are reflected in (generally) lower path metrics. Since the AP is a wired node, it is usually associated with an electrical outlet and energy cost is thus considered negligible

III. SCALABILITY OF HIERARCHICAL NETWORKS

Scalability issues for flat ad-hoc networks as addressed in [1] motivate our proposed hierarchical architecture with more than one tier of ad-hoc radio nodes in which the lower tiers aggregate the traffic up to the intermediate relay nodes, while continuing to use robust ad-hoc self-organization and routing protocols. In order to study the performance of the hierarchical architecture under consideration, the performance of two routing protocols (DSR and AODV) when applied to a hierarchical and a conventional flat ad-hoc network was compared using ns-2 simulations. From Fig 2, it can be seen that the system capacity increases significantly when a hierarchical approach is adopted for the particular system example under consideration. Similar gains are observed for both DSR and AODV. Performance measures such as delay and packet delivery ratio are also improved in the hierarchical system. Further details can be found in [4], in which the authors also study the scalability of the three-tier hierarchical network's capacity as a function of the relative densities of FNs and APs.

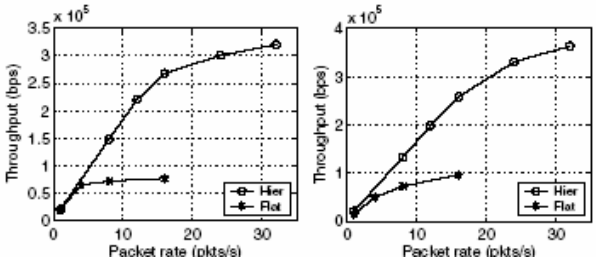

Figure 2. Performance of a) DSR and b) AODV applied to a flat and hierarchical network

Results indicate that it is possible to scale network capacity quite well with a mix of several (lower-cost) radio forwarding nodes and just a few wired access points.

IV. AD-HOC NETWORK PROTOCOLS FOR SOHAN

The above results motivated the design and development of a proof-of-concept prototype for the proposed self-organizing hierarchical ad-hoc network (SOHAN). The ad-hoc protocols used in the hierarchical network including those meant for 1) Bootstrapping, 2) Discovery, 3) Routing and Data Transmission are described below.

A. Bootstrapping

This phase involves the configuration of the different devices in terms of channel assignments and initial transmit power level settings. Note that the devices operate in the 802.11 ad-hoc [6] mode.

- Each AP is initialized on a pre-determined channel
- Each FN has two interfaces, one to communicate with other FNs and MNs (known as the *beaconing* interface) the other interface to communicate with APs (known as the *scanning* interface). These two interfaces are configured to operate on different channels that are specified at initialization so as to minimize interference.

In [7], a distributed bootstrapping mechanism that will automatically select appropriate channels for the particular interface based on the number of nodes already existing on that channel has been proposed. However, for the current implementation, the channel allocations are done manually using scripts.

B. Discovery

In traditional ad-hoc networks, there is no discovery phase and the routing protocol itself is responsible for building up topologies either using on-demand broadcast of route requests or by exchanging neighbor information proactively

* Note that the second intermediate-tier radio used on an FN may also use a wide-area cellular technology such as GPRS or CDMA2000 (3G) depending on coverage requirements.

with one hop neighbors. While this may be sufficient for smaller networks, as the number of nodes increases, it results in denser physical topologies, leading to extensive routing message exchanges. The problem is more severe in a multi-channel network where the multiple nodes that need to communicate could be on different radio channels. In this case, the routing messages need to be propagated across multiple channels in order to enable data transfer from one node to the other. In [8], using ns-2 simulations, it has been shown that by introducing discovery as a separate layer, the routing overhead is significantly reduced. These results, as shown in Fig. 3 demonstrate the improvement in routing overhead versus varying mobility and number of nodes with AODV as the routing protocol.

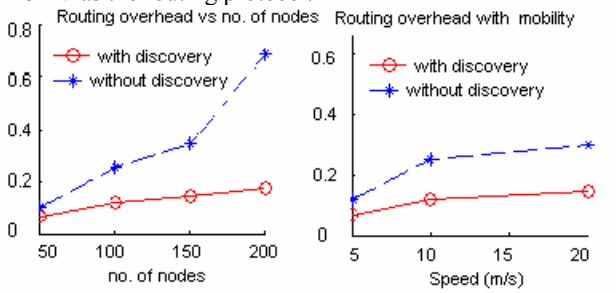

Figure 3. a) Routing overhead with increasing nodes b) Routing overhead with increasing mobility

Based on these results, we use augmented 802.11 MAC beacons and associations in SOHAN to support neighbor discovery and determination of the logical topology. Note also that for ease of implementation, the beacons used in the prototype are application-level packets, since actual 802.11 beacons are generated by the firmware in most of the existing 802.11b network adapters and are not customizable. The beacon format in SOHAN is shown in Fig 4.

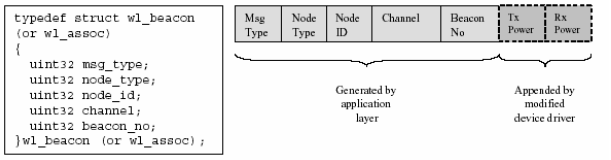

Figure 4. Beacon and Association Message Format

In SOHAN, FNs and APs periodically send beacons while the MNs scan different channels, listen to the beacons and send an association message to the best "cost" parent using the discovery metric described below.

1) Discovery metric

For our implementation, energy conservation at the MNs was chosen as the objective and the discovery metric was based on minimizing the transmit power consumption at the MNs. We modified the device drivers to append transmit power to each outgoing beacon at the APs/FNs and the received signal strength for each incoming beacon at the MNs. Using this information and assuming reciprocity of channel, the node with the minimum transmit power was chosen as the next hop neighbor. In case, there were two or more such nodes, the node whose beacon was received with the higher signal strength was chosen.

C. Routing

Motivated by the results in Fig. 3, we have implemented a distance-vector based routing protocol that uses the "logical" topology information presented by the discovery mechanism in order to create and maintain local neighbor tables at each of the FNs and APs. A combination of MAC addresses and node ID of the nodes is used for the routing protocol to handle the case of FNs that have two different MAC addresses for the two different interfaces but the same node ID. The routing protocol involves two phases: 1) Neighbor Table Formation and Updates 2) Table Update and Exchange. In phase 1, the FNs and APs build their local neighbor tables based on the beacons and the association messages exchanged during the discovery phase. The neighbor table format is shown in Table 1. Each entry is associated with a refresh timer that is reset or decremented respectively based on whether or not beacons are received from that neighbor every beacon interval.

Table 1. Local Neighbor Table Format

MAC Addr	Node Type	Refresh Timer	Channel to Next Hop	Cost to Dest	Interface to next Hop	Next Hop

During phase 2, FNs and APs exchange their local neighbor tables amongst themselves using sequence numbers to handle loops and update their neighbor tables based on this exchanged information. The MNs are not involved in the routing mechanism and simply forward their data to the best cost parent selected by the discovery procedure.

Note that any existing proactive (DSDV [5]) or reactive (DSR, AODV) routing mechanism can also be implemented on top of our discovery mechanism. For DSDV, the "forwarding table" at each node could be replaced by the neighbor table provided by discovery, while in AODV (or DSR), the route requests could be propagated only to a subset of nodes as selected by the discovery mechanism.

After the table exchanges, each FN computes a path that it can use to route data to the access point. The FN maintains a latest best cost path towards the AP at every instant and whenever it has data originated at the sensors, it consults the neighbor table to forward the data to the next hop on the channel and interface that it uses to reach the next hop neighbor. If a FN is disconnected from the network (there is no entry for an AP that exists in its neighbor table), it discards the packet and indicates a routing failure. Note that this routing implementation is based on the assumption that most of the traffic flow is from the MNs to the APs.

V. PROTOTYPE IMPLEMENTATION OF SOHAN

In this section, the practical design aspects and implementation of a proof-of-concept prototype for SOHAN architecture are described. The software architecture, protocol details, hardware architecture and initial experimental results are discussed in detail.

A. Software Architecture and Protocol Implementation

The implementation of the discovery and routing mechanism was done using C programming on embedded devices running Linux. We used the Libnet [9] open-source library to generate, send and receive custom packets. Fig. 5 shows the software architecture of the prototype. The modular software design as described below is consistent with the protocol stack and thus provides an easy way to modify functionality and add features at any layer.

- **Physical Layer**: The functionality of transmitting and receiving packets is handled using Libnet packet handling library that provides a portable and simplified interface for low-level network packet shaping, handling and injection.

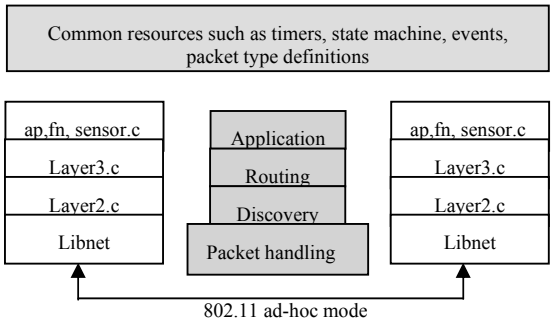

Figure 5. Software architecture of SOHAN

- **Layer2.c**: This handles the discovery and MAC layer functionality. Whenever a packet arrives from the lower layer, this layer handles the packet processing and passes the information to the higher layers.
- **Layer3.c**: This is responsible for handling the maintenance of the local neighbor tables and periodic exchange of neighbor tables amongst one-hop neighbors. The neighbor table is maintained and updated based on the beacons and the associations that are received from layer2.c. Upon the expiration of the route update timer, a periodic neighbor table exchange takes place. Entries are purged upon expiration of the refresh timer.
- **Application Layer** (Sensor.c, fwnode.c, ap.c): This layer handles the application specific functionality that depends on the type of the nodes.
- **Common functions**: The common functionality such as timer management, event management, finite state machine, packet type definitions and common wireless utilities is handled by programs common to all layers.

B. Hardware Platforms Used

The APs were based on a US Robotics 2450 Access Points running customized AP code for ad-hoc mode. The FNs were built on Compulab 586 CORE platform running a 133 Mhz processor with two PCMCIA slots for two wireless interfaces. The MNs were built on the embedded Cerfcube platform that was battery operated and ran customized sensor application. The selection of the hardware platforms was consistent with the system architecture and operated under the same set of constraints at each tier. Fig. 6 captures the different platforms used for the devices.

AP FN MN

Figure 6. Hardware platforms used for SOHAN

VI. EXPERIMENTAL RESULTS

The experimental nodes ran Linux (kernel 2.4.17) with device driver modifications for recording and appending transmit power and received power to every outgoing and incoming packet respectively. We ran simple tests to determine appropriate values for parameters such as beaconing interval, and channel dwell time prior to conducting our benchmark experiments.

A. Discovery delays versus beacon interval and dwell times

The MNs were configured to scan every alternate channel and varied the channel dwell times (from 100 ms to 1 sec) for different experimental runs. At the AP, the beacon interval was varied from 100 ms to 500 ms. As described in section III.B, the beacons were generated at the application layer and injected into the card using Libnet packet library. Also, scanning across channels at the MNs was performed at the application layer using *ioctl* calls to the device driver. We measured the discovery delays for a scenario consisting of a single AP and MN. This was repeated for different beacon intervals (100 ms, 250ms and 500 ms) at the AP with varying dwell times (from 100 ms to 1 sec) at the MNs.

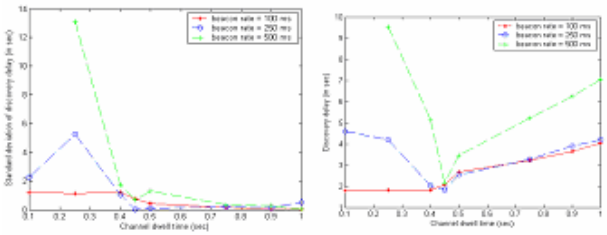

Figure 7. a) Discovery delays with different dwell times and beacon intervals b) Variance of discovery delay

Discovery delay is the time interval between beginning the experiment (both nodes starting at the same time) until the AP received the first 'association' message from the MN. Figure 7a show the results for the average discovery delays (in sec) for several sample runs for each setting along with the standard deviation of the delays. As shown in Figure 7b, for dwell times below 450ms, the discovery delay showed a high variation. This was because the application at the sensor missed a lot of beacons during its scan and hence the channel sweep iteration during which the first beacon was received was highly variable. When the dwell time per channel was higher than 450 ms, the variation of the discovery delay is significantly lesser than in the previous case. The performance with beacon intervals of 100ms and 250 ms was very similar. Hence, the beacon interval at the APs/FNs was chosen to be 250 ms with a channel dwell time of 450 ms at the MNs as a compromise between discovery delays and injecting more beacons in the network, which increased the discovery overhead.

B. Packet delivery ratio and average delays

In this experiment, the MNs transmitted at varying data rates to the AP over the hierarchical network. Two different packet sizes (1024 and 1472 bytes, UDP) were used. Fig. 8a shows that the packet delivery ratio for moderate loads is high. The small loss of packets may be attributed to the forwarding node rediscovery period, during which all packets received at the FN are dropped. For higher loads, the network degrades to deliver only 50 percent of the offered data. This is largely due to packets being dropped at the transmitter's interface. We also noticed that in such conditions, the application level beacons that we use were also dropped. This resulted in extremely large discovery times, which further amplified the problem of packet loss. However, using firmware-generated beacons should solve this problem. Fig. 8b shows that the end-to-end delay even for moderate loads is high (on the order of a second) which can be attributed to relatively high software latency with the embedded devices used. We note that this delay is the time elapsed between the MN application layer sending data and the AP application layer receiving data and includes system delays at the transmitter, receiver and switching between two interfaces at the intermediate FN. However, the system is tolerant to delays under increased traffic loads, which is due to the presence of two radio interfaces at each FN operating on different channels.

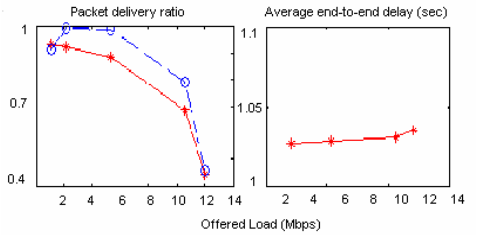

Figure 8.a) Packet delivery ratio and b) average delay of the network with increasing offered loads

A similar flat 802.11b ad-hoc network will tend to have a lower system capacity due to larger hop counts and a single frequency. These benchmark results indicate that the hierarchical network prototype we have developed provides promising results, which are fairly consistent with predictions from simulation. It is observed that implementing the discovery protocol in firmware will result in better network performance.

VII. CONCLUSIONS AND FUTURE WORK

We have presented the architecture and prototyping of SOHAN, a hierarchical self-organizing wireless ad-hoc network consisting of 802.11 based heterogeneous radio nodes at the three tiers. The architecture is motivated by potential improvements in scalability and system performance when compared with conventional flat ad-hoc networks. A proof-of-concept prototype was developed for evaluation of protocol design options and validation of system performance. Experimental results obtained so far are fairly consistent with predictions from simulation, and are indicative of the advantages of the proposed hierarchical structure with self-organizing discovery and routing protocols. Topics for future work include further optimization of discovery and routing algorithms, mobility support and joint MAC/routing methods for capacity and quality-of-service improvements.

VIII. REFERENCES

[1] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks", *IEEE Transactions on Information Theory*, vol. 46, March 2000, pp. 388-404.

[2] J.Broch, D.Maltz and D.Johnson, "Supporting Hierarchy and Heterogeneous Interfaces in Multihop Wireless Ad Hoc Networks", *Workshop on Mobile Computing*, June 1999

[3] E.M. Belding-Royer, "Hierarchical Routing in Ad hoc Mobile Networks"", *Wireless Communication and Mobile Computing*, pp.515-532, 2002.

[4] S. Zhao, I. Seskar and D. Raychaudhuri, "Performance and Scalability of Self-Organizing Hierarchical Ad Hoc Wireless Networks", *to appear in the Proceedings of the IEEE WCNC 2004,* March 21-24, 2004, Atlanta.

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

[6] IEEE 802 LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications", IEEE Standard 802.11, 1999.

[7] L. Raju, S. Ganu, B. Anepu, I. Seskar and D. Raychaudhuri, "BOOST: A Bootstrapping Protocol for Self-Organizing Hierarchical Ad-hoc Networks", *IEEE Sarnoff Symposium,* April 2004.

[8] L. Raju, S. Ganu, B. Anepu, I. Seskar and D. Raychaudhuri, "Beacon Assisted Discovery Protocol for Self-Organizing Hierarchical Ad-hoc Networks", submitted to *IEEE Globecom 04,* Nov. 2004.

[9] Libnet Packet Library, http://www.packetfactory.net/ libnet/